URL 정규화 향상을 위한 URL 서명
(URL Signatures for Improving URL Normalization)

순 레 이 키 † 이 상 호 ††
(Lay-Ki Soon) (Sang Ho Lee)

요 약 URL은 표준 URL 정규화에서 정의한 단계에 의하여 구문적으로 정규화된다. 본 논문에서는 웹 페이지의 메타데이타를 이용하여 표준 URL 정규화를 보완하는 기법을 제안한다. 메타데이타는 HTML 분석 도중 추출될 수 있는 웹 페이지 본문과 페이지 크기이다. 첫 번째 실험에서는 웹 페이지 본문이 동등한 URL 식별에 효과적이라는 것을 보인다. 두 번째 실험에서는 웹 페이지 본문을 Message-Digest 5 알고리즘으로 해싱하여 URL 서명을 만들며, 동일한 서명을 가지는 URL은 동일하게 취급한다. 두 번째 실험 결과에서, 우리가 제시한 URL 서명이 표준 URL 정규화와 비교하여 32.94%의 중복 URL을 더 감소시킬 수 있음을 알 수 있었다.

키워드 : URL 정규화, URL 서명, 웹 페이지 수집

Abstract In the standard URL normalization mechanism, URLs are normalized syntactically by a set of predefined steps. In this paper, we propose to complement the standard URL normalization by incorporating the semantically meaningful metadata of the web pages. The metadata taken into consideration are the body texts and the page size of the web pages, which can be extracted during HTML parsing. The results from our first exploratory experiment indicate that the body texts are effective in identifying equivalent URLs. Hence, given a URL which has undergone the standard normalization, we construct its URL signature by hashing the body text of the associated web page using Message-Digest algorithm 5 in the second experiment. URLs which share identical signatures are considered to be equivalent in our scheme. The results in the second experiment show that our proposed URL signatures were able to further reduce redundant URLs by 32.94% in comparison with the standard URL normalization.

Key words : URL normalization, URL signatures, web pages crawling

1. Introduction

World Wide Web (WWW) is inarguably one of the main sources for obtaining information nowadays. Relevant web pages are periodically crawled and indexed for the required information. However, the huge amount of information available on WWW demands tremendous effort in identifying and locating relevant information. Often, the information obtained from the web may be irrelevant or redundant.

The existing architecture of WWW uses Uniform Resource Locator (URL) or Uniform Resource Identifier (URI) to identify web pages [1]. Nevertheless, different URLs representing the same pages are commonly found on the web. Hence, in the process of crawling, URL normalization is performed by crawlers to determine if two syntactically different URLs are equivalent [1,2]. URLs are deemed equivalent if they point to the same resource or web page.

Owing to the ever-growing size of the web, URL normalization or canonicalization has become more

· This work was supported by the Korea Research Foundation funded by the Korean Government (MOEHRD) (KRF-2006-005-J03803).

† 학생회원 : 숭실대학교 컴퓨터학부
laykisoon@gmail.com
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
shlee199@gmail.com

논문접수 : 2008년 10월 15일
심사완료 : 2009년 1월 30일

정보과학회논문지 : 데이타베이스 제36권 제2호(2009.4)

and more crucial in helping crawlers to refrain from crawling and fetching the same web pages [1,2]. The ultimate aim of the URL normalization is to reduce redundant web crawling by having a set of URLs which point to a unique set of web pages. Besides, URL normalization is also deployed by search engines to determine the importance of web pages as well as to avoid indexing same web pages.

In contrast to the conventional standard URL normalization which compares two URLs without referring to the resources or web pages of the URLs, we are interested to explore the possibility of incorporating metadata of the associated web pages to enhance URL normalization process. In other words, instead of merely standardizing the URLs syntactically, we utilize some metadata of the web pages which are semantically meaningful to enhance the process of identifying equivalent URLs. The metadata considered in our experiment are the body texts and page size of the corresponding web pages, which can be obtained from HTML parsing without incurring unnecessary additional cost.

There are two main experiments featured in this paper. Both experiments start by applying the standard URL normalization on our dataset. The first experiment is conducted to explore the effectiveness of using the extracted body texts and the page size of the web pages associated to the URLs to identify syntactically different, but equivalent URLs. Our exploratory experiment shows that body texts are sufficiently effective in identifying equivalent URLs without considering the page size. In the second experiment, given the encouraging results from the first experiment, we proceed to construct URL signatures by hashing the extracted body texts using Message Digest Algorithm 5 (MD5). URLs which have identical URL signatures are considered as equivalent. Our results demonstrate that URL signatures are able to further reduce equivalent URLs by 32.94% in addition to the standard URL normalization.

Note that our proposed method is different from identifying redundant web pages where complete web contents should be taken into account for thorough comparison. Instead, we aim to enhance the standard URL normalization by identifying more equivalent URLs using the metadata of the associated Web pages.

The rest of this paper is organized as follows. Our preliminary study on the standard URL normalization is presented in Section 2. Section 3 discusses the related works. The metadata considered in our method, the construction of URL signatures and the flow of our proposed method are explained in Section 4. Section 5 presents the dataset as well as the evaluation metrics used to evaluate the experimental output. The experimental results are discussed in Section 6. Lastly, we conclude this paper in Section 7.

2. Standard URL Normalization

Figure 1 illustrates the typical flow of crawling process [3]. Given a set of seed URLs, the frontier which stores unvisited URLs is initialized at step 1. Each crawling loop then continue with four main steps, which include picking the next URL to crawl from the frontier, fetching the corresponding page of the URL, parsing the retrieved page to extract URLs or any other intended information, and finally adding the extracted and unvisited URLs into the frontier. In order to ensure storage efficiency of the frontier and to avoid fetching the same web pages

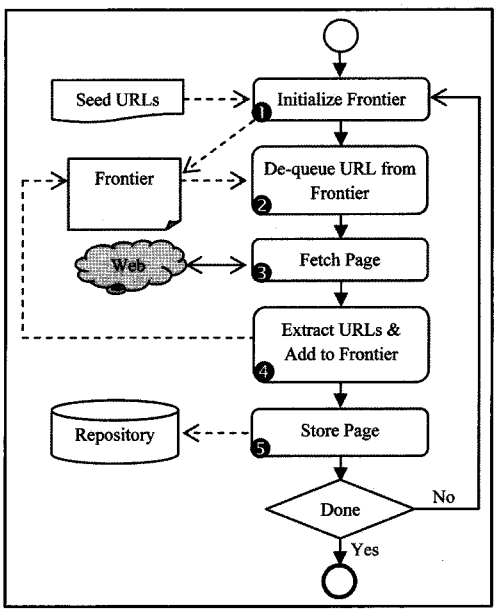


Figure 1 Typical flow of crawling process

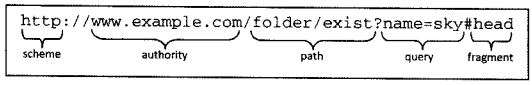


Figure 2 Components of URL

more than once, the URL normalization is performed on the URLs extracted at step 4 for discarding equivalent URLs.

Standard URL normalization consists of syntax-based, scheme-based and protocol-based methods [1,2]. Each step in these methods may focus on specific components of URL. Figure 2 shows the standard components of URL while Table 1 lists the steps in the standard URL normalization [1]. Other normalization means may be applied based on the application and prior knowledge about the sites. Prior knowledge may be obtained from previously crawled web pages.

After all the URLs have undergone the steps specified above, simple string comparison will be used to discard syntactically identical URLs. The fragment component of a URL denotes indirect identification of a secondary resource by referencing to a primary resource. For example, there may be links in a web page which link to different fragments within the same web page. Since the fragment component identifies the different fragments or resources within the same web page or the same URL, it is not considered in the standard URL normalization mechanism [1].

3. Related Works

URL normalization has been studied extensively in [2] and [4]. Additional steps were proposed to extend the standard URL normalization [2], which include changing the path component into lower case letters, eliminating the default pages of web servers and also eliminating the trailing slash symbol. Default pages considered in [2] are index.html, index.htm, default.html and default.htm. The goal of the extended URL normalization is to reduce false negatives while allowing false positives at a limited level. In our proposed method, we are able to identify equivalent URLs which contain default pages even without applying the extended URL normalization. Kim et al. presented a set of evaluation metrics to assess the performance of the steps in the standard URL normalization [4]. One of the steps proposed in [2], namely eliminating the trailing slash symbol was also evaluated in [4]. The proposed evaluation metrics are URL consistency, URL applying rate, URL reduction rate as well as true positive rate. Only URL reduction rate, as further explained in Section 5.2 is applied to evaluate the performance of our experiment. Different from [2] and [4], we have used contingency tables with the related evaluation metrics to compare the results of our proposed method with the standard URL normalization.

Bar-Yossef et al. proposed DustBuster to mine DUST (different URLs with similar text) rules from URL lists [5]. The URL lists were obtained from either previous crawls or web Server logs within a web site. DUST rules transform a given URL of that particular web site to others that are likely to have similar contents. One example of the DUST rules is *"http://site-name/story?=id="* → *"http://site-name/story_"*. DUST rules are generated by decomposing the components the URLs, followed by identifying the patterns of how URLs are formed within

Table 1 Standard URL Normalization

Methods	Steps
Syntax-based	i. Case normalization - convert all letters at scheme and authority components to lower case. ii. Percent-encoded normalization - decode any percent-encoded octet that corresponds to an unreserved character, such as %2D for hyphen and %5F for underscore. iii. Path segment normalization - remove dot-segments from the path component, such as '.' and '..'.
Scheme-based	i. Add trailinig '/' after the authority component of URL. ii. Remove default port number, such as 80 for http scheme. iii. Truncate the fragment of URL, e.g. *http://www.example.com/name.html#ali* is truncated to *http://www.example.com/name.html*.
Protocol-based	i. Only appropriate when the results of accessing the resources are equivalent. ii. For example, *http://example.com/data* is directed to *http://example.com/data/* by http origin server.

that web site using some heuristics. Although their DUST rules manage to produce as high as 26% of crawling reduction in one of the web sites, they need to set several thresholds in the process of generating DUST rules. Moreover, DUST rules are site-specific where each web site observes its own set of DUST rules. As of June 2008, Netcraft charted 174 millions of web sites available on the WWW [6]. As such, it is practically infeasible if DUST rules are to be mined from each web site individually. Different from Bar-Yossef et al.'s approach, our proposed method is not site-dependent and can be applied to all URLs without any additional costs.

4. Our Proposed Method

4.1 Metadata Considered

By convention, after undergoing the standard URL normalization process, URLs which are syntactically identical are deemed equivalent and thus get eliminated [3]. However, there are many syntactically different and yet equivalent URLs, which point to the similar web pages. For examples, two pairs of equivalent URLs which have been identified by our proposed URL signatures are *http://www.cnn.com/TECH/* equivalent to *http://www.cnn.com/technology/*, and *http://www.weather.com/jobs/* equivalent to *http://www.weather.com/careers/*. These two pairs would not be identified by the standard URL normalization mechanism since they are syntactically different.

These syntactically different but equivalent URLs have motivated us to explore the possibility of eliminating these syntactically different and yet equivalent URLs using the metadata of the corresponding web pages. In this paper, the metadata considered are the page size and the body text of the web pages.

We define the body text as textual data which is not embraced by any HTML tags within an HTML document. The body text in our proposal by no means represents the complete content of the web pages as we do not take into account the scripts, images, hyperlinks, style settings or any other types of data carried in the actual content of the web pages. Note that the same body text does not imply the same page size, which represents the size of the complete web page. Therefore, similar to the page size, we consider the extracted body text as metadata describing the web pages. Section 4.3 explains how the effectiveness of these metadata are investigated in the first experiment.

4.2 URL Signatures

As aforementioned, in the first experiment, we explore the effectiveness of using the body texts as well as the page size of the corresponding web pages in identifying equivalent URLs. The results of the first experiment, as detailed in Section 6 have convinced us that the body texts extracted from the web pages are sufficiently indicative in the effort of identifying equivalent URLs. In fact, the results have motivated us to further propose URL signatures, which are constructed by fingerprinting the body texts of the associated web pages using Message-Digest algorithm 5 (MD5).

The rationale behind fingerprinting the body texts instead of comparing them in their raw format is to reduce the comparison dimension. In other words, comparison on URL signatures which consists of only 32 hexadecimal characters is definitely faster than comparing hundreds or even thousands of words in the raw body texts. Besides, by having URL signatures, we may represent all URLs in fixed-size format.

Once the body text b_i of URL u_i is extracted, b_i will then be hashed into 32 hexadecimal characters $hash(b_i)$ by using MD5, forming the URL signature $sig(u_i)$ for u_i. In other words, instead of representing each URL with $hash(u_i')$, where u_i' denotes u_i after the standard URL normalization [7,8], we propose to use $hash(b_i)$ as the $sig(u_i)$ to represent u_i.

MD5 has been widely used to encrypt messages for secured transmission [9]. Besides, digest or fingerprints of files are also generated by using MD5, which is mainly used for checking the integrity of files. We use MD5 as the hashing function to generate our URL signatures because it is sensitive to even a small change. Figure 3 shows the basic steps in MD5 algorithm, more detailed algorithm can be referred at [9].

In short, given a set of URLs U, we extract the body texts B of the associated web pages, followed by generating URL signature $sig(u_i)$ for each URL u_i, where $sig(u_i) = hash(b_i)$, $u_i \in U$ and $b_i \in B$.

Step 1: Append padding bits.
Pad the original message so that its length is divisible by 512. Add 1 to the end of the message and followed by 0s to make the length up to 64 bits fewer than a multiple of 512.

Step 2: Append length.
64 bits are appended to the end of the padded message to denote the length of the original message in bytes. Break the 64 bit length into 2 words (32 bits each). Subsequently, append the low-order word and followed by high-order word.

Step 3: Initialize MD buffer.
The 128-bit buffer of MD5 is initialized as:
- The buffer is divided into 4 words (32 bits each), named as A, B, C, and D.
- Word A is initialized to: 0x67452301.
- Word B is initialized to: 0xEFCDAB89.
- Word C is initialized to: 0x98BADCFE.
- Word D is initialized to: 0x10325476.

Step 4: Process message in 512-bit blocks.
For each input block, 4 rounds of operation, each with 16 operations are performed. The details of this main step can be referred at [9].

Step 5: Produce output.
The contents in buffer words A, B, C and D are returned following the sequence of low-order byte first.

Figure 3 Basic steps of MD5 algorithm

URLs with the same signatures are considered equivalent and will be eliminated for avoiding redundant crawling and fetching of the same web pages.

4.3 The Flow of Our Proposed Method

In this paper, we apply our proposed method on regular URLs U_{reg}, where U_{reg} is defined as a set of URLs which appear in all the crawling sessions. The rationale behind our decision to use only U_{reg} for our experiment comes in line with the objective of our proposed method, which aims to reduce redundant web crawling. Much benefit can be gained if redundant crawling of these regular URLs can be reduced.

Figure 4 illustrates the flow diagram of our proposed method. The input data is the set of regular URLs U_{reg}, while the expected output data is a final set of unique URLs U_{fin}, which has undergone standard URL normalization and our proposed method. Note that there are two branches of processes after the third process, where the box on the left illustrate the fourth process in our first exploratory experiment, whereas the box on the right illustrates the subsequent processes (highlighted) in our second experiment, which constructs the URL signatures.

In the first process, given $U_{reg} = \{u_1, u_2, \cdots, u_m\}$, where m is the number of URLs in U_{reg}, we apply the steps of standard URL normalization, as listed in Table 1. After performing string comparison among the standard-normalized URLs, we eliminate identical URLs and obtain a list of syntactically unique URLs $U_{std} = \{u_1, u_2, \cdots, u_n\}$, where n is the number of URLs left after the standard URL normalization process. The second process is commenced by fetching the corresponding web pages of the URLs in U_{std}. In order to obtain the body texts, these web pages are de-tagged and the page sizes are recorded in third process. We de-tag the Web pages by using *ParserDelegator* from *javax.swing.html*.

In the first experiment, as included in the left box, all URLs in U_{std} are compared in terms of the associative body text and page size in the fourth process. To evaluate how well the adopted metadata perform in identifying equivalent URLs either individually or by combination, we consider the following options in the exploratory experiment:

i. Body text only

ii. Page size and body text

In the first option where only body text is considered, URLs with share the same body texts are predicted as equivalent. Eventually, only URLs

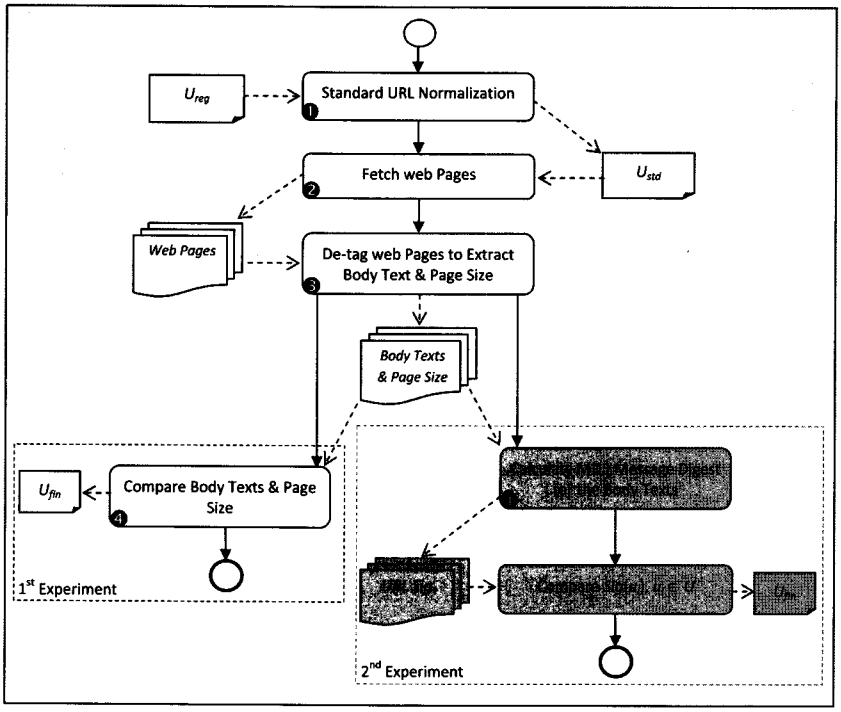


Figure 4 Flow diagram of our proposed method

which have unique body texts are included in U_{fin}. Intuitively, the page size is not sufficiently indicative to be considered by itself because it is recorded in continuous numerical format. Hence, in the second option, we attempt to identify equivalent URLs by comparing both their page size as well as the extracted body text. Likewise, only URLs which share the same page size and body texts are considered as equivalent.

Based on the results from the first experiment, we subsequently carry out the second experiment, where the first three processes remain the same. In the fourth process as highlighted in the right box, the extracted body texts are MD5-hashed using the *MessageDigest* class provided by *java.math*. The generated 32 hexadecimal characters message digests or fingerprints form the URL signatures for all the URLs in U_{std}. Finally, URL signatures among the URLs are compared and URLs which share same signatures are eliminated to form U_{fin} in the second experiment. Owing to the relatively small dataset, we verify the identified equivalent URLs manually in this paper.

5. Dataset and Evaluation Metrics

5.1 Dataset

For the purpose of this experiment, we have used Web Data Extractor 7.0 [10] to obtain URLs from five web sites, as listed in Table 2. These web sites are selected considering the diverse nature of the web sites, as well as the geographical location. These web sites were crawled every two days, starting from 9 April to 27 May 2008, amounting to 25 crawling sessions. The results produced by Web Data Extractor 7.0 include a list of URLs within the web site, and metadata of each URL, such as keywords, title, description, page size and the date of last modification.

As mentioned above, to obtain a steady set of URLs for our experiment, we have selected only URLs which appear in all the crawling sessions to be included in U_{reg}. Table 2 shows the average number of URLs per crawling session and the percentage of U_{reg} for each web site. There are a total of 5257 URLs in the U_{reg} for this experiment. The page sizes of the web pages in this experiment are

Table 2 Web sites crawled for the experimental dataset

Web site	Nature of Company / Country	Average Number of URLs Per Crawling	Number of Regular URLs	Percentage of Regular URLs
Arirang Korea <HTUhttp://www.arirang.co.krUTH>	Broadcasting / Korea	4362	1017	23.31%
British Telecom <HTUhttp://www.bt.comUTH>	Telecommunication / United Kingdom	4776	849	17.78%
Cable News NetworkTF[1)]	News cable television network / United States	3968	594	14.97%
Multimedia University <HTUhttp://www.mmu.edu.myUTH>	Education / Malaysia	1306	891	68.22%
The Weather Channel <HTUhttp://www.weather.comUTH>	Weather forecasts / United States	3967	1906	48.05%
Total		18379	5257	28.60%

obtained from Web Data Extractor 7.0 while the body texts of web pages are de-tagged or extracted by using *ParserDelegator* from *javax.swing.html*. The average size of the body text of our dataset is around 4KB.

5.2 Evaluation Metrics

To evaluate the effectiveness of our URL signatures, we apply one of the metrics proposed by Lee et al. [4], which is URL reduction rate:

$$\text{URL Reduction Rate} = \frac{|U_{std}| - |U_{fin}|}{|U_{std}|} \quad (1)$$

Originally, the first evaluation metric is used to measure the probability that a normalized URL equals to the non-normalized URLs such that

$$\text{URL Reduction Rate} = (M_b - M_a) / N$$

where M_b is the number of URLs to be handled before the normalization, M_a is the number of URLs to be handled after the normalization and N is the number of URLs to which a normalization method is applied [6]. In our experiment, M_b and M_a represent the number of URLs in U_{std} since our proposed method considers the metadata extracted from all the URLs in U_{std}. As such, URL redundancy rate denotes the percentage of URLs that can be reduced by our proposed URL normalization method.

In addition, we also tabulate our results in a contingency table as shown in Table 3. Having the contingency table, we further analyze the results by using the following metrics [11]:

$$\text{Sensitivity} = \text{TP} / \text{Positive} \quad (2)$$

$$\text{Specificity} = \text{TN} / \text{Negative} \quad (3)$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (4)$$

$$\text{Accuracy} = \text{Sensitivity} * (\text{Positive}/(\text{Positive}+\text{Negative})) + \text{Specificity} * (\text{Negative}/(\text{Positive}+\text{Negative})) \quad (5)$$

Table 3 Description of contingency table

		Prediction		
		Positive	Negative	Total
Actual	Positive	True Positive (TP)	False Negative (FN)	Positive
	Negative	False Positive (FP)	True Negative (TN)	Negative

Positive indicates the number of URLs which are equivalent whereas negative indicates the number of non-equivalent URLs in our dataset. True positive denotes the number of equivalent URLs which are identified or predicted as equivalent by the metadata or our URL signatures, and false negative otherwise. Likewise, true negative carries the number of non-equivalent URLs which are predicted as non-equivalent.

As such, sensitivity and specificity evaluate the performance of our proposed method to incorporate the metadata of the web pages (in the first experiment) and URL signatures (in the second experiment) in terms of identifying actual equivalent and non-equivalent URLs respectively. On the other hand, precision shows the percentage of correct prediction in identifying equivalent URLs while accuracy presents the performance of as a whole.

6. Results and Discussion

1) T The file of the crawling session for CNN on 7 May 2008 is corrupted. Hence the selected regular URLs of CNN are from 24 crawling sessions.

6.1 Exploratory Experiment on Body Texts and Page Size

Having selecting only the regular URLs from all the crawling sessions, we have obtained U_{reg} = {u_1, u_2, ···, u_m} where m = 5257. We have then applied the steps of the standard URL normalization on U_{reg}, as listed in Table 1. After performing string comparison among the standard-normalized URLs, we have gained a list of syntactically unique URLs U_{std}, where U_{std} = {u_1, u_2, ···, u_n} and n = 5089. In other words, a total of 168 syntactically equivalent URLs have been identified by the standard URL normalization. MMU observes the highest reduction of equivalent URLs, which amounts to 18.29%. Most of the syntactically equivalent URLs in MMU's U_{reg} contain percent-encoded octet of unreserved characters. For an instance, '~' is stated as '%7E' in some of its URLs.

Given the standard-normalized URLs U_{std}, we have then proceeded to fetch the web pages and de-tagged the web pages for extracting the body texts. To obtain the U_{fin} in the first exploratory experiment, we have eliminated equivalent URLs in U_{std} by comparing the URLs with regard to the adopted metadata of the corresponding web pages. Table 4 and 5 show the contingency tables of using the metadata in identifying equivalent URLs in addition to the standard URL normalization, as applied in the two options in our experiment respectively. The evaluation metrics for both options are computed in Table 6.

Table 4 The contingency table of the first option (body text only)

		Prediction		
		Positive	Negative	Total
Actual	Positive	1679	0	1679
	Negative	235	3343	3578
	Total	1914	3343	5257

Table 5 The contingency table of the second option (page size and body text)

		Prediction		
		Positive	Negative	Total
Actual	Positive	1206	473	1679
	Negative	235	3343	3578
	Total	1441	3816	5257

Referring to Table 4 and 5, the first option has produced better normalization results where 1679 (1511 on top of the 168 equivalent URLs identified by teh standard URL normalization technique) of actual equivalent URLs are identified, compared to only 1206 identified by the second option. Besides, the second option has recorded as high as 473 of false negatives. Most of the similar web pages linked by equivalent URLs observe the difference of merely few kilobytes in terms of the page size, such as 1 or 2KB. These can be due to insignificant differences, such as an extra space appearing in the web pages. On the other hand, the main reason of having nonequivalent URLs being identified as equivalent (false positive) in the first option is due to our proposed method which considers only textual data (body text) within the web pages. In other words, these web pages may observe differences in other types of their web contents, such as images or hyperlinks.

Table 6 Evaluation metrics for the 2 options in our experiment

Metrics / Metadata	Option 1: Body Text	Option 2: Page Size & Body Text
URL Reduction Rate	0.36	0.27
Sensitivity	100%	71.83%
Specificity	93.43%	93.43%
Precision	87.72%	83.69%
Accuracy	95.53%	86.53%

As listed in Table 6, the first option outperforms the second option significantly in terms of identifying actual equivalent URLs by having 100% of sensitivity, compared to only 71.83% by the second option. However, both options perform similar in terms of identifying nonequivalent URLs, having same specificity of 93.43%. Besides sensitivity, the first option also observes higer precision and accuracy. The overall accuracy achieved by the first option is 9% higher than the second option, amounting to 95.53%.

In terms of reducing redudant or equivalent URLs, the first option has 3343 URLs predicted as nonequivalent in its U_{fin} whereas the second option has 3816 (refer to Table 4 and 5). Therefore, the first option records better URL reduction rate of 0.36, outperforms the second option by 0.09. Based on

the results from this first experiment, we may conclude that body text by itself is sufficiently effective to be used as the metadata to represent the web pages for the purpose of identifying equivalent URLs.

6.2 Second Experiment with URL Signatures

Given the convincing results from the first experiment, we have proceeded MD5-hashed the body texts extracted from the web pages in the second experiment. In other words, URL signatures are generated for all URLs in U_{std} by MD5-hashing the extracted body texts.

Table 7 lists some of the equivalent URLs identified by URL signatures within these five web sites. Rows numbered 3, 5, 6, 7 and 9 show that URL signatures are able to identify equivalent URLs which contain the default pages without omitting the default pages as proposed by [2]. Besides, the 4th and 8th row indicates that these are the default web pages in the particular directories although they do not contain the typical names for default pages, such as *index.html* in the URLs. Likewise, the first and second pairs of equivalent URLs demonstrate that "*&id=&page=1*" and "*&page=1*" are the corresponding default web pages. Obviously, the standard URL normalization mechanism would not be able to detect the equivalency listed in Table 7 since all of them are syntactically different.

As expected, we have obtained the same results in identifying equivalent URLs by using both the raw body texts and URL signatures. Table 8 shows the overall contingency table of the standard URL normalization technique, while Table 9 compares the effectiveness of our URL signatures with the standard URL normalization using the evaluation metrics.

As we can see from Table 8, out of 5257 URLs in U_{reg}, 5089 of them are syntactically different. Out of 1679 equivalent URLs in U_{reg}, only 168 equivalent URLs are reduced from U_{reg} in order to form U_{std}. The remaining 1511 URLs which are syntactically different will form U_{std} with other 3578 non-equivalent URLs in Uneg. Nevertheless, note that the standard URL normalization records 0 for its false positive because only syntactically identical URLs are considered to be equivalent.

Likewise, refer back to in Table 4, 3578 out of

Table 8 Contingency table of the standard URL normalization

		Prediction		
		Positive	Negative	Total
Actual	Positive	168	1511	1679
	Negative	0	3578	3578
	Total	168	5089	5257

Table 7 Examples of equivalent URLs

No.	Equivalent URLs
1	• http://www.arirang.co.kr/Blog/Arirang_Town.asp?code=Bl1 • http://www.arirang.co.kr/Blog/Arirang_Town.asp?code=Bl1&id=&page=1
2	• http://www.arirang.co.kr/News/News_List.asp?code=Ne5 • http://www.arirang.co.kr/News/News_List.asp?code=Ne5&page=1
3	• http://www.bt.com/ • http://www.bt.com/index.jsp
4	• http://www.bt.com/business/ • http://www.bt.com/business/home/
5	• http://www.cnn.com/CNN/Programs/american.morning • http://www.cnn.com/CNN/Programs/american.morning/index.html
6	• http://www.cnn.com/TECH/science/archive/ • http://www.cnn.com/TECH/science/archive/index.html
7	• http://www.mmu.edu.my/~fom/ • http://www.mmu.edu.my/~fom/index.html
8	• http://www.mmu.edu.my/~dev/ • http://www.mmu.edu.my/~dev/aboutus.htm
9	• http://www.weather.com/activities/homeandgarden/home/mosquito/ • http://www.weather.com/activities/homeandgarden/home/mosquito/index.html
10	• http://www.weather.com/achesandpains/ • http://www.weather.com/activities/health/achesandpains/

5257 URLs in U_{reg} have unique URL signatures and grouped in U_{neg}. However, 1511 (false negative in Table 8) equivalent URLs which are not identified by the standard URL normalization technique have identical URL signatures. Hence, these 1511 URLs are predicted to be equivalent by our proposed method. However, there are 235 non-equivalent URLs mistakenly predicted as equivalent by our method.

The URL reduction rate in Table 9 shows that there is a possibility of 0.33 (0.36 - 0.03) for each URL in U_{std} to have same URL signature with its peer URLs in U_{std}. The 100% of sensitivity of our proposed URL signatures shows that all equivalent URLs in U_{pos} are identified by the URL signatures. However, only 87.72% (1679) of the 1914 URLs predicted to be equivalent by URL signatures are indeed equivalent.

Table 9 The evaluation metrics

Metrics	Standard URL Normalization	URL Signatures
URL Reduction Rate	0.03	0.36
Sensitivity	10.01%	100%
Specificity	100%	93.43%
Precision	100%	87.72%
Accuracy	71.26%	95.53%

On the other hand, there is 6.57% (100% - 93.43%) of non-equivalent URLs in U_{neg} share the same URL signatures. This is mainly due to the fact that our signatures are constructed using only the body texts of the Web pages, without considering other types of Web contents. The standard URL normalization technique records higher specificity and precision compared to our proposed method as it records 0 false positive. The main reason is that URLs are only considered as equivalent if are they syntactically identical in the standard URL normalization technique.

Since the main objective of using URL signatures is to identify equivalent URLs, our proposed method is more preferable as it records 100% of sensitivity compared to only 10.01% by the standard URL normalization. In fact, our proposed method shows promising accuracy of 95.53% compared to only 71.26% of accuracy by the standard URL normalization technique.

Table 10 further compares the results of using our proposed URL signatures in addition to the standard URL normalization mechanism. As we can see, when URL signatures are constructed and referred on top of the standard URL normalization, another 1732 (1900 - 168) equivalent URLs have been identified, contributing to 32.94% (36.14% - 3.20%) of additional reduction in redundant crawling and retrieval of the same web pages.

7. Conclusion and Future Works

In contrast to the conventional way of representing URLs using the hashed value of URLs after the standard URL normalization, we proposed to represent URLs using URL signatures. The URL signature for each URL is the MD5-hashed body text of the corresponding web page, where body text is the textual data which is not embraced by any HTML tags within that particular web page. In other words, we enhance the standard URL normalization mechanism by generating and utilizing URL signatures to further identify syntactically different and yet equivalent URLs. Comparing to the standard URL normalization mechanism, our experimental results demonstrate that the proposed URL signatures managed to identify equivalent URLs as accurate as 95.53%, which further reduce equivalent URLs by additional 32.94%. Similar results were

Table 10 Results comparison between the standard URL normalization mechanism and URL signatures

URL Normalization	Arirang	BT	CNN	MMU	Weather	Total of URLs	Reduced URLs	
							No.	%
Regular URLs	1017	849	594	891	1906	5257		
Reduction by the standard URL normalization	1016	848	593	728	1904	5089	168	**3.20**
Reduction by using URL signatures in addition to the standard URL normalization	816	134	531	598	1278	3357	1900	**36.14**

obtained when applying URL signatures on the the same web sites which were crawled in a longer period [12].

For the future works, we plan to study the performance of our proposed method in a larger scale, which includes crawling more web sites, we will apply our method in our web crawler. Besides body texts, we also plan to explore the possiblity of incorporating other metadata of web pages, such as page size to dynamically construct the URL signatures. Since the web changes rapidly, we are also interested to derive the optimum frequency of re-contructing URL signatures for the URLs.

References

[1] Berners-Lee, T., Fielding, R., Masinter, L., *"Uniform Resource Identifier (URI): General Syntax,"* available at Hhttp://gbiv.com/protocols/uri/rfc/rfc3986.htmlH.

[2] Lee, S. H., Kim, S. J., Hong, S. H., *"On URL Normalization,"* in Proceedings of the 2005 International Conference on Computational Science and its Applications (ICCSA), Singapore, pp. 1076-1085, May 2005.

[3] Pant, G., Srinivasan, P., Menczer, F., *"Crawling the Web,"* Web Dynamics 2004, pp. 153-178.

[4] Kim, S. J., Jeong, H. S., and Lee, S. H., *"Reliable Evaluations of URL Normalization,"* in Proceedings of the 2006 International Conference on Computational Science and its Applications (ICCSA), Glasgow, pp. 609-617, May 2006.

[5] Bar-Yossef, Z., Keidar, I., Schonfeld, U., *"Do Not Crawl in the DUST: Different URLs with Similar Text,"* in the Proceedings of the International World Wide web Conference (WWW 2007), pp. 111 - 120, May 2007.

[6] Netcraft June 2008 Web Server Survey, available at: http://news.netcraft.com/archives/web_server_survey.html

[7] Burner M., *"Crawling Towards Eternity: Building an archive of the World Wide Web,"* Web Techniques Magazine, 2(5), May 1997.

[8] Chakrabarti, S., *Mining the web, Discovering Knowledge from Hypertext Data*, Morgan Kaufmann Publishers, Elservier, San Francisco, CA, 2003.

[9] The MD5 Message-Digest Algorithm, available at: http://tools.ietf.org/html/rfc1321

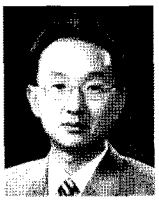
[10] Web Data Extractor, available at: http://www.webextractor.com/

[11] Han, J., Kamber, M., *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, Elsevier, San Francisco, CA, 2006.

[12] Soon, L. K. and Lee, S. H., *"Identifying Equivalent URLs using URL Signatures,"* to appear in the Proceedings of the 4th IEEE International Conference on Signal-Image Technology & Internet-Based Systems (SITIS 2008), Bali, Indonesia, December 2008.

순레이키

1999년 Bachelor of Computer Science, University of Putra Malaysia(학사) 2002년 Master of Science, University of Putra Malaysia(석사). 2003년~2006년 Multimedia University Malaysia IT 학과 강사. 2006년~현재 숭실대학교 대학원 컴퓨터학과 박사과정 관심분야는 인터넷 데이타베이스, 데이타베이스



이 상 호

1984년 서울대학교 컴퓨터공학과(학사) 1986년 미국 노스웨스턴 대학교 전산학과(석사). 1989년 미국 노스웨스턴 대학교 전산학과(박사). 1990년~1992년 한국전자통신연구원 선임연구원. 1999년~2000년 미국 조지메이슨 대학교 방문교수. 2006년~2007년 미국 타우슨 대학교 방문교수. 1992년~현재 숭실대학교 컴퓨터학과 교수. 관심분야는 인터넷 데이타베이스, 데이타베이스 튜닝 및 성능 평가