URL 정규화의 적용 효과 및 평가
(Effects and Evaluations of URL Normalization)

정 효 숙 † 김 성 진 †† 이 상 호 †††
(Hyo Sook Jeong) (Sung Jin Kim) (Sang Ho Lee)

요 약 하나의 웹 문서는 문법적으로 서로 다른 두개 이상의 URL들로 표현 가능하다. URL 정규화는 URL을 정형화된 형태로 변환하는 과정이다. 정규화 과정을 통하여, 동일 웹 문서를 나타내는 URL들은 문법적으로 동일 URL들로 변환된다. 현재까지 정규화 방법의 개발 및 적용은 개발자의 경험적 지식에 기반을 두고 있으며, 체계적인 분석에 대한 연구가 부재하다. 본 논문에서는 웹 어플리케이션의 효율성과 효과성 측면에서 정규화 방법들을 평가하여 적절한 정규화 방법의 선택에 대한 지침 제공을 목적으로 한다. 또한, 웹 어플리케이션에서 정규화 적용으로 발생하는 효과를 분석하고, URL 정규화 평가를 위한 7가지 척도를 기술한다. 끝으로, 실제 웹 문서에서 추출된 약 2천 5백만개의 URL들을 대상으로 12개의 정규화 방법이 평가된다.

키워드 : URL, URL 정규화, URL 정규화 평가

Abstract A web page can be represented by syntactically different URLs. URL normalization is a process of transforming URL strings into canonical form. Through this process, duplicate URL representations for a web page can be reduced significantly. A number of normalization methods have been heuristically developed and used, and there has been no study on analyzing the normalization methods systematically. In this paper, we give a way to evaluate normalization methods in terms of efficiency and effectiveness of web applications, and give users guidelines for selecting appropriate methods. To this end, we examine all the effects that can take place when a normalization method is adopted to web applications, and describe seven metrics for evaluating normalization methods. Lastly, the evaluation results on 12 normalization methods with the 25 million actual URLs are reported.

Key words : URL, URL normalization, URL normalization evaluation

1. 서 론

URL(Uniform Resource Locator)은 웹에 존재하는 웹 자원(또는 문서)의 위치를 나타내는 문자열이다. 웹 로봇(web robot), 웹 브라우저(web browser), 프록시 서버(proxy server)등의 웹 어플리케이션들은 URL을 통하여 웹 문서를 요청하여 다운로드하거나 웹 문서를 저장하고 관리한다. 웹 문서의 위치는 다양한 URL로 표현이 가능하며, 동일 웹 문서를 가리키는 URL들을 동일(equivalent) URL이라고 한다. 동일 웹 문서를 가리키는 서로 다른 문자열의 URL들을 동일 URL로 인식하지 못할 경우, 웹 어플리케이션은 다수의 동일 문서를 반복 처리하게 된다. 예를 들어, 웹 로봇[1-4]은 동일 문서의 중복 저장에 따른 디스크 I/O와 사용 공간의 증가, 불필요한 문서 요청 회수 및 다운로드에 따른 네트워크 사용량 증가 등을 겪을 수 있다.

URL 정규화는 URL을 정형화된(canonical) 형태로 변환하는 과정이다. 정규화 과정을 통하여, 서로 다른 문자열로 표현된 동일 URL들이 동일한 문자열로 변환된다. "잘못된 긍정"은 비동일 URL을 동일 URL로 판단하는 것이다. URL 정규화에 "잘못된 긍정"이 발생할 경우, 동일하지 않은 두 URL을 동일한 문자열로 변환하게 된다. "잘못된 부정"은 동일한 URL들을 동일하지 않은 URL로 판단하는 것이다. URL 정규화에 "잘못된 부정"이 발생할 경우, 서로 다른 문자열의 동일 URL을 동일한 문자열의 URL로 변환하지 않는다.

표준화 그룹[5]에서는 문법 기반의 정규화(syntax-

· 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-005-J03803)
† 학생회원 : 숭실대학교 컴퓨터학과
hsjeong@comp.ssu.ac.kr
†† 학생회원 : 서울대학교 전기컴퓨터공학부
sjkim@oopsla.snu.ac.kr
††† 종신회원 : 숭실대학교 컴퓨터학부 교수
shlee@comp.ssu.ac.kr
논문접수 : 2005년 11월 18일
심사완료 : 2006년 5월 11일

based normalization), 스킴 기반의 정규화(scheme-based normalization), 프로토콜 기반의 정규화(protocol-based normalization)의 세 가지 정규화를 정의하였다. 표준 URL 정규화는 "잘못된 긍정"의 발생을 엄격히 피하고 "잘못된 부정"을 최소화하는 것을 목적으로 하고 있다. [6]은 표준 URL 정규화 방법의 성능 향상을 위한 확장 가능성과 필요성에 관하여 기술하였다. 또한, 표준 URL 정규화로부터 경로 요소의 대소문자 구분과 경로 요소의 마지막 슬래시("/") 기호의 제거와 기본문서 제거에 관한 정규화 사항을 논의하였다.

URL 정규화는 다양한 웹 어플리케이션에서 광범위하게 사용됨에도 불구하고, URL 정규화 적용이 웹 어플리케이션의 URL 처리에 미치는 효과를 이론적으로 분석하고 실험한 연구는 부재하다. 또한 현재까지 확장 URL 정규화는 적용 효과에 대한 정확한 이해 없이 사용자들의 경험과 직관에 의거하여 개발되어 적용되고 있으며, 다양한 정규화 방법들을 평가하여 개발자들의 정규화 선택에 지침을 주는 연구가 없다. 본 논문의 의의는 URL 정규화 적용이 웹 어플리케이션의 URL 처리에 미치는 효과를 상세히 분석하고 동시에 URL 정규화를 평가할 수 있는 평가 척도를 제안하는데 있다. 제안된 척도는 확 장 URL 정규화 뿐 아니라 표준 URL 정규화 모두에 적용될 수 있다.

사용자는 웹 어플리케이션의 효율성과 효과성을 고려하여 어플리케이션에 사용될 정규화 방법을 선택한다. 만약 어플리케이션의 효율성이 강조될 경우, 정규화를 통한 URL의 개수 감소가 정규화 선택의 주요한 기준이 된다. 사용자는 다수의 중복된 URL을 제거할 수 있는 정규화를 우선적으로 선택한다. 경우에 따라, 제한적 수준에서 "잘못된 긍정"을 부분적으로 허용하더라도 "잘못된 부정"을 현저히 감소시킬 수 있는 정규화 방법을 선택하기도 한다[6].

웹 어플리케이션의 효과성이 효율성 보다 강조될 경우, 적용되는 정규화가 웹 어플리케이션에 미치는 효과의 적절성이 정규화 선택의 주요한 기준이 된다. 사용자는 정확한 정규화(즉, "잘못된 긍정"을 발생시키지 않으며 동일한 URL들만을 동일 문자열로 변환시키는 정규화)만을 어플리케이션에서 사용할 수도 있다. 예를 들어, 페이지랭크 알고리즘과 같이 하이퍼링크의 URL 정보를 이용하는 순위 알고리즘에서는 정확한 정규화 방법을 선택하는 것이 중요하다. 이와 반대로, 어플리케이션의 목적에 따라 특정 "잘못된 긍정"이 선호되는 정규화 방법을 선택할 수도 있다. 예를 들어, 임의의 URL로 웹 문서의 다운로드를 실패하였으나, 이를 정규화하여 웹 문서를 성공적으로 다운로드하였다고 하자. 이는 동일하지 않은 두 URL(정규화 적용 전 URL은 존재하지 않는 웹 문서를 가리키고 있으며, 정규화 적용 후의 URL은 임의의 웹 문서를 가리키고 있다)을 동일하다고 판단하여 동일 문자열로 변환하였으므로 "잘못된 긍정"이 발생한 것이다. 그러나 웹 로봇은 이러한 "잘못된 긍정"이 발생하는 정규화를 오히려 선호할 수 있다.

본 논문에서는 정규화가 웹 어플리케이션에 미치는 효과를 분석하고 정규화 방법을 평가할 수 있는 평가 척도로서 URL 적용율, URL 감소율, URL 일관성, 문서 비손실율, 문서 손실율, 문서 획득율, 문서 변화율을 제안한다. 표준 문서[5]에서는 정규화 적용 시 바른/잘못된 긍정과 바른/잘못된 부정만을 고려하지만, 본 논문에서는 정규화 적용 시 웹 어플리케이션에서 발생하는 10가지 경우를 고려하여 정규화 효과를 크게 4가지로 분류하였다. 제안된 척도는 정규화 평가를 통하여 사용자들이 적합한 정규화 방법을 채택하는 지침이 될 수 있다. 본 논문에서는 2005년 7월에 20,000개의 한국 웹 사이트로부터 웹 로봇[2]을 이용하여 추출된 약 2천 5백만개의 URL들을 대상으로 12가지의 정규화 방법들을 평가한다.

본 논문은 다음과 같이 구성된다. 2장에서는 표준 URL 정규화와 확장 URL 정규화에 대해서 기술한다. 3장에서는 URL 정규화의 효과를 기술하고 평가 척도들을 제안한다. 4장에서는 실험 결과를 분석하고 마지막으로, 5장에서는 결론을 맺고 향후 계획을 기술한다.

2. URL 정규화

2.1 URL의 구성요소

URL은 스킴(scheme), 권한(authority), 경로(path), 질의(query), 단편(fragment)으로 구성된다[5]. 스킴은 웹 서버와의 통신에 사용될 프로토콜이 기술된다. 일반적으로 HTTP(Hyper Text Transfer Protocol) 프로토콜이 웹 브라우저와 웹 서버 사이의 통신에 사용된다.

권한은 사용자 정보, 호스트(host), 포트번호의 하위 구조로 구성된다. 사용자 정보에는 해당 웹 문서에 접근하기 위한 사용자 이름과 암호 등이 포함된다. 사용자 정보와 호스트는 엣("@") 기호로 구분된다. 호스트는 웹 서버의 위치를 나타내며, 웹 서버의 도메인 이름(domain name)이나 IP 주소(Internet Protocol address)로 기술된다. 포트 번호는 도메인 이름 또는 IP 주소와 콜론(":")으로 구분하여 사용한다. 예를 들어, 80번 포트를 사용하는 ACM 사이트"www.acm.org"의 웹 서버는 호스트에 "www.acm.org:80"으로 기술되거나 "199.222.69.250:80"로 기술될 수 있다.

경로는 웹 문서가 위치한 디렉토리와 파일명이 기술된다. 각 디렉토리와 파일은 슬래시("/") 문자로 구분된다. 디렉토리는 현재 디렉토리와 상위 디렉토리를 의미하는 문자열로서 "."과 ".."이 사용될 수 있다. 예를 들

어, 동일 사이트에서의 두 경로 문자열 "/membership/benefits.html"과 "/membership/../membership/benefits.html"는 동일 문서를 나타낸다.

질의는 웹 문서에 입력되는 파라미터(parameter)들의 이름과 값이 기술된다. 질의 문자열은 물음표("?")로 시작된다. 파라미터의 이름과 값은 등호("=")로 구분된다. 하나 이상의 파라미터가 존재할 때 각 파라미터는 앰퍼센트("&")로 구분된다. 예를 들어, 질의 "?name=smith &age=25"는 두 개의 파라미터 "name"과 "age"의 값이 각각 "smith"와 "25"임을 의미한다.

단편은 문서내의 특정 단락을 가리킨다. 단편의 시작은 샵("#") 문자로 시작한다. 예를 들어, URL "http://example.com/list.htm#chap1"은 파일 "list.htm"에서 "chap1"이라고 명명된 단락을 가리킨다.

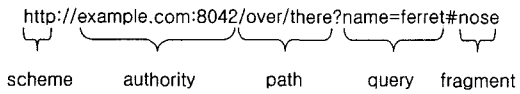


그림 1 URL의 구성

데이타 옥텟(data octet)은 하나의 데이타 문자를 표현하는 8자리 비트(bit) 문자열이다. 퍼센트 인코딩(percent encoding)은 URL 구성 요소 구분자나 데이타 옥텟으로 허용되지 않는 문자가 데이타 옥텟으로 표현되어야 할 때 사용되며, 퍼센트 인코딩 문자 트리플릿(triplet)으로 표현된다. 즉, 퍼센트 문자("%")와 옥텟의 숫자 값을 나타내는 두 자리 16진수로 표현된다. 예를 들어, "%20"은 USASCII 스페이스에 해당하는 2진법 옥텟 "00100000"의 퍼센트 인코딩 값이다.

2.2 표준 URL 정규화

URL 정규화는 다양하게 표현된 동일 URL들을 정형화된 형태로 변환하는 과정이다. 정규화 과정을 통하여, 문법적으로 서로 다른 동일 URL들은 문법적으로 동일한 URL들로 변환된다. 표준화 그룹[5]에서는 문법 기반의 정규화, 스킴 기반의 정규화, 프로토콜 기반의 정규화의 세 가지 유형의 정규화에 대해서 기술하였다.

문법 기반 정규화는 문자 정규화, 퍼센트 인코딩 정규화, 경로 세그먼트(segment) 정규화의 세 가지 정규화로 나뉜다. 문자 정규화는 트리플릿 안의 16진수로 표현된 문자열을 대문자로 변환하고(예를 들어, "%3a"를 "%3A"로 변환) 스킴과 호스트내의 문자열은 소문자로 변환한다. 예를 들어, "HTTP://EXAMPLE.com"는 "http://example.com/"으로 변환된다. 퍼센트 인코딩 정규화는 허용가능한 문자가 데이타 옥텟으로 퍼센트 인코딩되어 있을 때, 이를 디코딩하는 것을 말한다. 예를 들어, "http://example.com/%7Esmith"는 "http://example. com/~smith"으로 변환된다. 경로에서 "."과 ".."은 상대적인 참조를 하기 위해 설계되었다. 경로 세그먼트 정규화는 URL 문자열에 현재 디렉토리와 상위 디렉토리를 나타내는 "."와 ".."을 제거하는 정규화이다. 예를 들어, "http://example.com/a/b/./../c.htm"는 "http://example.com/a/c.htm"으로 정규화 된다.

스킴 기반 정규화는 스킴의 종류에 따라 정규화가 방법이 다를 수 있으며, "http" 스킴일 경우 다음과 같은 정규화를 정의한다. 첫째, 기본포트 번호(즉, 80번 포트)는 제거된다. 예를 들어, "http://example.com:80/"는 "http://example.com/"으로 정규화 된다. 둘째, 만약 경로가 널(NULL)일 경우 "/"로 대체된다. 경로가 주어지지 않은 URL은 웹 서버의 루트 디렉토리(root directory)를 요청하는 URL과 동일하다. 예를 들어, "http://example.com"는 "http://example.com/"로 정규화 된다. 셋째, 단편이 존재할 경우 이를 제거한다. 예를 들어, "http://example.com/list.htm#chap1"은 "http://example.com/list.htm"으로 정규화 된다.

프로토콜 기반 정규화는 관례적으로 다양한 형태로 표현 될 수 있는 동일 URL들에 대한 정규화이다. 예를 들어, "http://example.com/a/b/"와 "http://example.com/a/b"(경로 세그먼트 "b"는 디렉토리임)가 관례적으로 항상 동일하게 사용될 경우, 전자의 URL을 후자의 URL로 변환하는 정규화가 있을 수 있다.

2.3 확장 URL 정규화

표준 URL 정규화는 URL 변환 시 "잘못된 긍정"을 유발하지 않는다. 즉, 문법적으로 서로 다른 URL을 동일한 URL로 판단하여 동일한 문자열로 변환하지 않는다. 표준 URL 정규화는 "잘못된 긍정"이 전혀 발생하지 않는 반면 "잘못된 부정"(동일 URL을 비동일 URL로 판단하여 변환하지 않는 것)이 높은 수치를 나타내는 단점을 가진다. 확장 URL 정규화는 "잘못된 긍정"을 일부 허용할지라도 표준 URL 정규화 적용 시 발생할 수 있는 "잘못된 부정"을 줄이기 위해 개발된다. "잘못된 부정"이 줄어들면, 동일 웹 문서를 나타내는 중복된 URL들의 개수가 감소한다. 따라서 확장 URL 정규화의 효과는 다수의 URL을 처리하는 웹 어플리케이션에서 보다 크게 나타난다.

예를 들어, 두 개의 URL이 있다고 가정하자.

- u1: http://www.acm.org/
- u2: http://www.acm.org/index.html

URL u1은 루트 디렉토리의 기본 문서를 요청하고 URL u2는 URL 문자열에 기재된 기본 문서를 요청한다. 실제 웹에서 두 URL은 동일한 웹 문서를 가리키는 동일 URL이다. 표준 URL 정규화에 따르면 u1과 u2는 동일하게 변환되지 않으므로, 웹 어플리케이션은 동일한

웹 문서를 불필요하게 반복 처리 할 수 있다. 기본 문서를 나타내는 문자열이 "index.html"이라고 하고, 기본 문서 문자열을 제거하는 확장 URL 정규화를 사용하는 경우, u1과 u2는 동일하게 변환되므로 동일 웹 문서에 대한 불필요한 반복 처리를 피할 수 있다. 그러나 모든 웹 서버에서 "index.html"이 기본 문서를 나타내는 문자열로 사용되는 것은 아니며, 이때 기본 문서 문자열을 제거하는 확장 URL 정규화는 "잘못된 긍정"을 유발하게 된다.

3. 정규화 평가 척도

본 장에서는 URL 정규화가 실제 웹 어플리케이션에서 미치는 효과를 분석하고 정규화 평가를 위한 7가지 척도로서 URL 적용율, URL 감소율, URL 일관성, 문서 비손실율, 문서 손실율, 문서 획득율, 문서 변화율을 제안한다.

URL 정규화는 하나의 웹 자원에 대한 중복 표현 감소를 목적으로 하므로 정규화 적용에 따른 중복 표현의 감소를 평가하는 척도가 필요하다. URL 적용율은 보유한 URL중에서 해당 정규화가 적용되어 변경되어지는 URL들의 비율을 나타내며, "URL 감소율"은 해당 정규화가 적용되는 URL 중에서 중복 표현으로 제거되는 URL의 비율이다. 웹 어플리케이션에서 임의의 정규화에 적용 가능한 전체 URL 개수를 N이라고 가정하자. 임의의 정규화에 적용 가능한 URL 개수와 정규화 수행 후 감소되는 URL 개수에 대해서 URL 적용율과 URL 감소율은 다음과 같이 정의된다.

URL 적용율 = (N / 정규화 전 URL 개수)

URL 감소율 = (정규화 전 URL 개수 - 정규화 후 URL 개수) / N

예를 들어, 웹에서 200개 URL들을 수집하고 수집된 URL들 중 100개 URL들이 90개 URL들로 정규화 된다고 하면 URL 적용율은 (100 / 200) = 0.5이고 URL 감소율은 (200 - 190) / 100 = 0.1이다.

임의의 URL로 하나의 웹 문서를 다운로드한 이후, 동일한 URL이 웹 문서 요청에 사용되더라도, 동일한 문서가 지속적으로 다운로드 된다는 보장이 없다. URL 일관성은 임의의 URL로 동일한 문서를 지속적으로 다운로드하는 정도를 나타낸다. 일관적인 URL이 문서 요청에 사용될 경우, 일정 시간동안 동일한 문서가 지속적으로 다운로드된다. 주어진 시간 t안에 다운로드되는 문서의 개수를 R_t라고 하자. 웹 문서 요청이 실패할 경우(문서 다운로드를 실패한 경우)는 빈 문서가 다운로드된 것으로 간주한다. URL 일관성은 다음과 같이 정의된다.

URL 일관성 = 1 - ((유일한 문서 개수 - 1) / (R_t - 1))

예를 들어, 2초 동안 5번의 웹 문서를 요청하고, 다운로드된 결과가 ⓐ, ● ⓑ, ●, ⓒ라고 하자. ●은 다운로드가 실패한 문서를 나타내며, ⓐ, ⓑ, ⓒ은 다운로드 성공한 문서의 내용을 나타낸다. 이 경우 URL 일관성은 1 - ((4 - 1) / (5 - 1)) = 0.25 이다.

일관적이지 않은 URL들이 정규화 평가에 사용될 경우, 정규화 적용 전과 후의 URL들의 요청으로 다운로드된 웹 문서를 비교하여 해당 정규화의 옳고 그름을 판단할 수 없다. 예를 들어, 임의의 URL로 다운로드된 웹 문서와 정규화된 URL로 다운로드된 웹 문서가 상이하더라도 "잘못된 긍정"이 발생한 것으로 판단할 수 없다. 따라서 URL 정규화는 일정 수준 이상의 일관적인 URL들을 대상으로 평가되어야 한다.

임의의 정규화를 통해 많은 URL 중복 표현들이 제거된다고 하더라도 웹 어플리케이션은 잘못된 변환으로 인한 부작용을 겪을 수 있다. 즉 URL 정규화는 중복 표현을 줄이면서도 적용하기 전과 후에 보유한 문서 개수가 같아야 하나, 잘못된 변환으로 보유해야할 문서를 보유하지 못할 수 있다. URL이 정규화 되었을 때 웹 어플리케이션의 문서 획득에 미치는 효과를 알아보자. URL u1이 URL u2로 정규화 되고, u1과 u2는 각각 웹 문서 p1과 p2를 가리킨다고 하자. 문서 획득에 미치는 효과는 u2의 웹에 존재 유무(u2가 실제 웹에서 발견 가능 여부)와 p1과 p2의 문서 존재 유무 따라 10가지 경우로 요약된다(표 1 참고).

(1) 문서 p1이 웹에 존재

(A) 문서 p2는 웹에 부재(경우 4와 경우 9) : u1이 u2로 변환됨으로써 p2 문서를 획득할 수 없었으므로 "잘못된 긍정"이 발생한다. 결과적으로 p1 문서를 손실한다.

(B) 문서 p2는 웹에 존재하며, p1과 p2가 동일한 문서임(경우 1과 경우 6) : u1이 u2로 변환됨으로써 동일한 문서를 획득하였으므로 옳은 정규화 방법이다("잘못된 긍정" 발생하지 않음). 정규화를 수행하지 않을 경우 u1과 u2는 각각 p1과 p2를 요청하여야 하나, 정규화 수행 후 문서 요청 횟수를 줄일 수 있다.

(C) 문서 p2는 웹에 존재하며, p1과 p2가 상이한 문서임(경우 2와 경우 7) : 이 경우는 정규화 수행 후 서로 다른 문서를 다운로드 하였으므로 "잘못된 긍정"이 발생한다. 잘못된 정규화 방법으로 인해 p1 문서를 잃게 된다. 만약 u2가 실세계 웹에 존재하지 않는 URL이였다면(예를 들어, u2는 새로 발견된 URL이며 경우 7을 말함) u2 문서 요청으로 인해서 u1과 다른 새로운 문서를 획득하게 될 것이다. 이와 반대로, u2가 실제 웹에 존재하는 URL이였다면(예를 들어, u1은 이미 존재

표 1 정규화 방법의 효과성

u2 \ u1		문서 존재		문서 부재
u2 데이타베이스내 존재	문서 존재	동일	비손실(1)	비손실(3)
		상이	손실(2)	
	문서 부재	손실(4)		비손실(5)
u2 데이타베이스내 부재	문서 존재	동일	비손실(6)	이득(8)
		상이	변화(7) (= 손실+이득)	
	문서 부재	손실(9)		비손실(10)

하는 URL로 u2로 변환되며 경우 2를 말함) 문서 p2를 잃게 된다.

(2) 문서 p1이 웹에 존재 하지 않음

(A) u2는 실세계 웹에 존재(예를 들어, u2는 데이타베이스 내에 존재하는 URL임) : 문서 p1이 존재하지 않으므로 손실할 웹 문서도 없다. 또한 u2는 정규화 수행 후 데이타베이스 내에 존재하는 URL로 변환하였으므로 한 번의 문서요청 횟수를 줄일 수 있다. 경우 3과 경우 5를 나타낸다.

(B) u2는 실세계 웹에 부재(예를 들어, u2는 데이타베이스 내에 부재하는 URL임) : 이 경우 두 가지 경우로 나눠서 생각해 볼 수 있다. 만약 p2가 웹에 존재한다면(경우 8), 하나의 새로운 웹 문서를 얻는다. 이와 반대로 p2가 웹에 존재하지 않는 문서였다면(경우 10), u1과 u2의 문서가 모두 부재하였으므로 어떤 웹 문서도 잃지 않는다. 두 경우 모두 문서요청 횟수에는 변함이 없다.

임의의 URL 정규화 방법이 적용될 경우, 웹 어플리케이션은 얻어야 할 웹 문서를 손실(경우 2, 경우 4, 경우 9)하거나, 정규화가 수행되지 않을 경우 얻지 못할 새로운 웹 문서를 획득(경우 8) 하거나, 문서 p1을 손실하는 대신에 다른 문서 p2를 획득하거나(경우 7), 웹 문서 획득(경우 1, 경우 3, 경우 5, 경우 6, 경우 10)에 아무런 영향을 받지 않을 수 있다. 정규화 방법의 효과성을 보이는 4가지 척도는 다음과 같다. 문서 비손실율, 문서 손실율, 문서 획득율, 문서 변화율의 합은 1이다.

문서 비손실율 = 전체 비손실한 문서들의 개수 /*N*

문서 손실율 = 전체 손실한 문서들의 개수 /*N*

문서 획득율 = 전체 획득한 문서들의 개수 /*N*

문서 변화율 = 전체 변화한 문서들의 개수 /*N*

임의의 정규화가 웹 어플리케이션에 비손실의 효과를 나타내는 것은 "잘못된 부정"이 없었음을 의미한다. 즉, 적용된 정규화가 올바르게 URL을 변환하였음을 의미한다. 반대로 임의의 정규화가 손실, 획득, 변화를 나타내는 경우 u1과 u2는 서로 다른 URL이고 정규화가 올바르게 적용되지 않았음을 의미한다.

4. 실험적 평가

4.1 실험 환경

본 실험은 다음과 같은 단계로 수행된다. 첫째, 웹 로봇은 웹 문서들을 수집하기 위해 사용된다. 둘째, 수집된 웹 문서로부터 원시 URL들을 추출한다. 셋째, 정규화가 적용될 URL 집합을 얻기 위해 간단한 문자열 비교 연산을 통해 중복된 URL을 제거한다. 이 단계는 URL 정규화를 배제하고 문법적으로 서로 다른 URL들의 집합을 얻는다. 넷째, 상대 URL들을 절대 URL들로 변환한다. 다섯째, 절대 URL들에 표준 URL 정규화(문법 기반 정규화, 스킴 기반 정규화)를 적용한다. 이 단계에서 생성된 URL들을 "표준화된 URL"이라고 명명한다. 여섯째, "표준화된 URL"들을 대상으로 6가지 확장 URL 정규화(자세한 설명은 4.2장에서 기술됨)를 적용한다. 일곱째, 정규화가 적용되기 전과 후의 URL들로 웹 문서를 요청하여 요청 결과를 비교한다.

웹 로봇[2]을 이용하여 2005년 7월에 한국 웹 사이트로부터 랜덤(random)하게 추출된 20,000개의 사이트들을 대상으로 655,645개의 웹 문서를 수집하였다. 로봇은 각 사이트별로 수집할 수 있는 최대 문서 개수를 3,000개로 제한하였고 사이트 루트 문서로부터 9홉(hop) 이내의 문서를 요청하였다. 타임아웃(timeout)은 3초로 설정하였으며, 3초 이내로 웹 서버로부터 응답이 없을 경우 해당 웹 문서에 대한 요청을 포기하였다.

수집된 웹 문서들로부터 약 2천 5백만개(25,838,285개)의 원시 URL들을 추출하였다. 추출된 원시 URL들에는 다수의 동일 URL들이 포함되어 있다. 예를 들어, "http://www.example.com/"가 두개의 웹 문서에서 발견되었다면, 동일한 문자열의 두 URL이 모두 원시 URL 집합에 포함되었다.

원시 URL 집합에서 서로 다른 URL들의 집합을 얻기 위해 URL 정규화를 배제하고 중복된 URL을 제거하였다. 다음과 같은 세 가지 간단한 문자열 비교를 통하여 중복 제거를 수행하였다. 첫째, 동일 문서 내에서 발견된 URL들 중 동일 문자열의 URL 중복을 제거한다. 둘째, 동일 사이트 내에서 절대 경로 URL(즉, 슬래

표 2 중복된 URL들의 제거 효과

	URL 개수	비율
원시 URL	25,838,285	100%
문서 내, 동일 문자열의 URL 중복 제거	22,757,954	88.1%
동일 사이트 내, 동일한 문자열의 절대 경로 URL 중복 제거	19,647,693	76.0%
동일한 문자열의 절대 URL("http:"로 시작하는 URL) 중복 제거	11,046,159	42.8%
상대 URL들을 절대 URL로 변환 후 동일 문자열 절대 URL 중복 제거	2,329,770	9.0%

시로 시작하는 URL)의 동일 문자열을 제거한다. 셋째, 원시 URL 집합에서 절대 URL(즉, "http:"로 시작하는 URL)의 동일 문자열 URL 중복을 제거한다. 표 2에서는 각 문자열 비교연산 적용 후 남은 URL들의 개수와 중복된 URL의 제거 결과를 나타내고 있다. 세 가지 문자열 비교 연산 후, 획득한 11,046,159개의 URL들을 대상으로 상대 URL들을 절대 URL로 변환하여 2,329,770개의 절대 URL을 얻었다. 최종적으로 2,329,770개의 URL이 정규화 평가를 위해 사용된다.

4.2 정규화 평가 결과

본 절에서는 표준 URL 정규화와 확장 URL 정규화의 평가 결과를 기술한다. 우선, 2,329,770개의 절대 URL에 대한 표준 URL 정규화 적용 후의 평가 결과를 보이고, 표준 URL 정규화가 적용된 "표준화된 URL"들에 대한 URL 일관성을 측정한다. 또한 6가지의 확장 URL 정규화를 소개하고 일관성 있는 표준 URL을 대상으로 확장 URL 정규화를 적용하여 평가한다.

표준 URL 정규화는 "잘못된 긍정"이 전혀 발생하지 않는다. 따라서 문서 손실/획득/변화율은 0이고 문서 비손실율은 1이다. 표준 URL 정규화에 대한 URL 적용율과 감소율은 그림 2에 나타나 있다. 표준 URL 정규화 방법 6가지는 SN1, SN2, ..., SN6으로 표기한다.

SN1 : 스킴 요소 소문자 변환 정규화
SN2 : 호스트 소문자 변환 정규화
SN3 : 기본 포트(즉, ":80") 제거
SN4 : 경로가 없을 경우 슬래시 기호로 대체

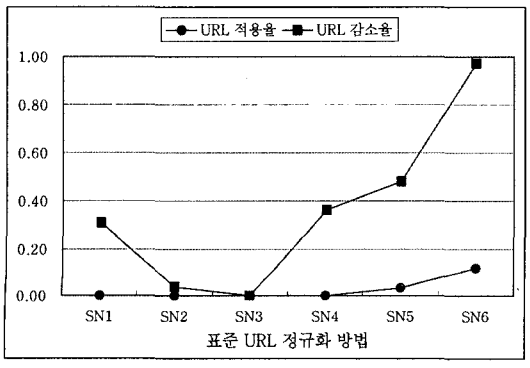


그림 2 표준 URL 정규화의 URL 적용율과 URL 감소율

SN5 : 예약되지 않은 문자 디코드
SN6 : 단편 요소 제거

SN1, SN2, SN3, SN4의 URL 적용율은 0.01 미만의 값으로 나타났으며, SN5와 SN6의 URL 적용율은 각각 0.03과 0.12으로 나타났다. URL 감소율을 살펴보면, SN2과 SN3이 0.05 미만의 값을 나타냈다. 즉, SN2나 SN3을 적용할 경우 변환된 URL의 5% 미만의 URL이 중복으로 제거되었다. 또한, SN1과 SN4가 적용된 URL의 3분의 1이 중복으로 제거되었으며, SN5가 적용된 URL들은 약 절반(48.5%)이 중복으로 제거되었다. SN6의 URL 적용율과 URL 감소율은 각각 11.7%와 97.3%로 나타났다. 즉, 절대 URL의 11.7%는 SN6에 의해서 정규화 되었으며, 정규화된 URL의 97.3%가 중복으로 제거되었다. SN6의 URL 적용율과 URL 감소율은 다른 표준 URL 정규화와 비교하여 상대적으로 매우 높게 나타났으며, SN6의 적용이 어플리케이션에서 처리할 URL의 개수 감소에 상대적으로 큰 영향을 나타내었다. 표준 URL 정규화 6가지를 모두 적용하여 2,027,512개의 "표준화된 URL"을 획득하였다.

URL 일관성은 "표준화된 URL"을 대상으로 3초 동안 웹 문서를 3번 요청하여 측정되었다. 문서 요청을 3번 할 경우, 다운로드된 문서의 개수는 1개거나 2개 또는 3개일 수 있다. 다운로드된 문서 개수가 1개(즉, 모두 동일 문서)라면, URL 일관성은 1이다. 다운로드된 문서의 개수가 3개(즉, 모두 다른 문서)라면, URL 일관성은 0이다. URL 일관성이 0인 URL은 문서 요청 시마다 매번 상이한 문서가 다운로드 되었음을 의미한다. 다운로드된 문서의 개수가 2개라면, URL 일관성은 1 - ((2 - 1) / (3 - 1)) = 0.5이다. 그림 3은 2,027,512개의 "표준화된 URL"들에 대한 URL 일관성 값의 분포를 나타낸다. x축은 URL 일관성 값을 나타내며 y축은 URL 개수를 나타낸다. 절대 URL의 33%(663,081개)가 URL 일관성이 0으로 나타났으며 65%(1,290,450개)가 URL 일관성이 1로 나타났다. 일관성이 1로 나타난 1,290,450개의 URL들이 확장 URL 정규화 방법의 평가에 사용되었다.

이후, 6가지 확장 URL 정규화 방법들을 소개하고 평가 결과를 나타낸다. URL의 경로 요소에서는 원칙적으로 대소문자를 구분한다. 그러나 윈도우(Windows) 운

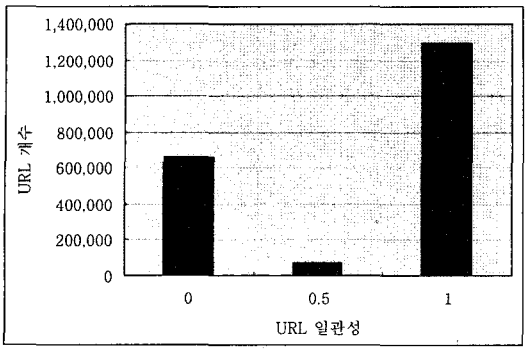


그림 3 URL 일관성

영체제가 기본적으로 사용되는 파일 시스템인 FAT과 NTFS는 디렉토리와 파일이름에 대한 대소문자를 구분하지 않는다. 윈도우 운영체제 기반의 웹 서버에서, 한 웹 문서는 경로의 대소문자 표현을 달리하여 다양한 URL로 표현될 수 있다. 예를 들어, "http://www.nasdaq.com/asp/ownership.asp"과 "http://www.nasdaq.com/ASP/ownership.asp"은 동일 문서를 나타내는 URL이다. 한편, 유닉스(Unix)나 리눅스(Linux) 운영체제는 디렉토리와 파일이름에 대한 대소문자를 구분하여, 대소문자 구분이 다른 두 URL 문자열은 서로 다른 문서를 나타낸다. 예를 들어, "http://www.acm.org/pubs/journals.html"과 "http://www.acm.org/PUBS/journals.html"은 다른 문서를 나타낸다. 윈도우 운영체제에서 경로 요소의 대소문자의 표현을 달리하는 URL들로 인한 "잘못된 부정"을 줄이기 위해 다음의 확장 URL 정규화를 고려해 볼 수 있다.

EN1 : 경로 요소 소문자 변환 정규화

질의 요소의 대소문자 구분은 질의 파라미터를 입력으로 받는 프로그램의 대소문자 처리에 따라 결정된다. 예를 들어, "http://nasdaq.com/asp/MasterDataEntry.asp?page=Charts"과 "http://nasdaq.com/asp/MasterDataEntry.asp?page=charts"은 동일 문서를 가리킨다. 질의 요소의 대소문자 표현을 달리하는 URL들로 인한 "잘못된 부정"을 줄이기 위해 다음의 확장 URL 정규화를 고려해 볼 수 있다.

EN2 : 질의 요소 소문자 변환 정규화

경로 요소의 마지막 문자가 슬래시 기호인 URL은 디렉토리를 가리키는 URL이다. 웹 클라이언트가 웹 서버에게 디렉토리를 요청할 경우에, 웹 서버는 요청된 디렉토리의 기본문서를 보여주거나 디렉토리내에 존재하는 모든 파일을 목록화하여 보여주는 웹 문서를 생성하여 반환한다. 디렉토리를 요청하는 마지막 슬래시 기호는 종종 생략한다. 이때 웹 서버는 마지막 슬래시 기호가 생략된 URL을 마지막 슬래시 문자를 포함한 URL로 리다이렉션한다. 예를 들어, "http://acm.org/pubs"는 "http://acm.org/pubs/"로 리다이렉션된다. 경로부의 마지막 슬래시 문자에 의한 "잘못된 부정"을 줄이기 위하여 다음의 확장 URL 정규화를 고려해 볼 수 있다.

EN3 : 마지막 슬래시 문자 제거 정규화

기본문서는 웹 클라이언트가 디렉토리를 요청할 때 웹 서버의 응답에 사용되는 문서이다. 예를 들어, "index.html"을 기본 문서로 하는 사이트(acm.org)에서 "http://www.acm.org/"과 "http://www.acm.org/index.html"은 동일 문서를 가리킨다. 웹 서버 관리자는 기본 문서로 사용될 문서의 파일명을 설정할 수 있다. [7]의 보고에 따르면, 실제 웹에서 사용되는 웹 서버의 85%를 아파치(Apache)와 IIS(Internet Information Server)가 차지하고 있다. 이 두 서버에서 "index.htm", "index.html", "default.htm"가 기본 문서로 설정되어 있다. 기본 문서로 인해 발생하는 "잘못된 부정"을 줄이기 위해 다음과 같은 확장 URL 정규화를 고려해 볼 수 있다.

EN4 : 기본 문서("index.htm") 제거 정규화

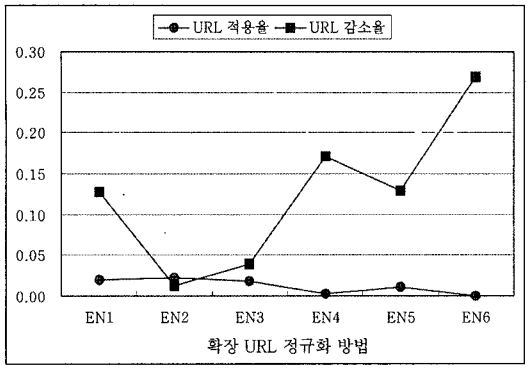
EN5 : 기본 문서("index.html") 제거 정규화

EN6 : 기본 문서("default.htm") 제거 정규화

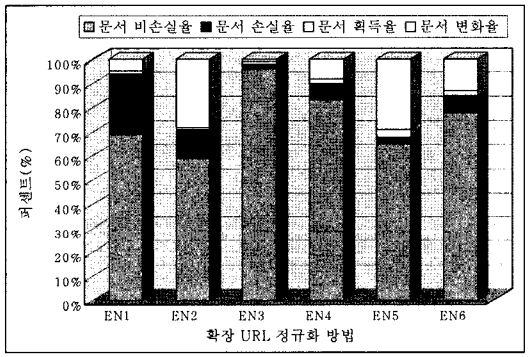
그림 4(가)는 6가지 확장 URL 정규화 방법들의 URL 적용율과 URL 감소율을 나타낸다. 26,169(20.3%), 290,104(22.5%), 22,938(1.8%), 3,783(0.3%), 13,678(1.1%), 171(0.0%)개의 URL들이 6가지의 확장 URL 정규화에 각각 적용되었으며, 적용된 URL의 1.3%, 0.1%, 4.0%, 17.1%, 12.9%, 27.0%가 중복으로 제거되었다. 확장 URL 정규화 방법에 대한 URL 감소율은 표준 URL 정규화 적용 시에는 감소시킬 수 없었던 URL의 중복("잘못된 긍정"을 포함하고 있음)을 확장 URL 정규화의 적용으로 감소시키는 정도를 나타낸다.

각 확장 URL 정규화에서 발생하는 "잘못된 긍정"은 그림 4(나)에 나타났다. 그림 4(나)는 확장 URL 정규화의 문서 비손실율, 손실율, 획득율, 변화율을 나타낸다. 문서 비손실율은 "잘못된 긍정"이 발생하지 않음을 의미한다. 문서 손실율, 획득율, 변화율은 "잘못된 긍정"이 발생함을 의미한다. 6가지 확장 URL 정규화에서 문서 비손실율이 가장 높은 비중을 차지하였으나 "잘못된 긍정" 또한 상당히 발생하였다(31%, 41%, 4%, 17%, 35%, 22%). "잘못된 긍정" 발생의 대부분은 문서 손실과 문서 변화가 그 이유로 나타났다. 문서 획득율은 6가지 확장 URL 정규화에서 2% 이하로 나타났다.

그림 4는 처리할 URL의 개수를 줄이면서 웹 어플리케이션의 문서 수집에 미치는 효과를 보여준다. 사용자들은 확장 URL 정규화 평가 결과를 통해 사용할 정규화를 선택할 수 있다. 예를 들어, 웹 로봇에서 사용할 확장 URL 정규화의 선택 기준이 수집되는 문서 개수라



(a) URL 적용율과 URL 감소율



(b) 문서 비손실/손실/획득/변화 비율

그림 4 확장 URL 정규화 평가

고 하면, 사용자는 획득율이 높고 손실율이 낮은 정규화를 선택하게 된다. 이때 문서 변화율은 다운로드된 문서 개수에 변함이 없으므로 정규화 선택에 영향을 미치지 않는다. "실손실율"을 손실율과 획득율의 차이로 정의하고 "실손실율"이 0.05 미만인 확장 URL 정규화만을 웹 로봇에 적용한다고 하면, EN3(0.02)와 EN5(0.03)가 웹 로봇 정규화에 적용된다.

5. 결론 및 향후 계획

본 논문에서는 URL 정규화에 관한 체계적인 분석과 더불어 정규화 선택의 지침이 될 수 있는 평가 척도를 제안하였다. 정규화의 효과는 웹 어플리케이션에서 발생하는 10가지 경우를 고려하여 정규화 효과를 4가지로 분류되고, URL 적용율, URL 감소율, 문서 비손실율, 문서 손실율, 문서 획득율, 문서 변화율이 정규화의 평가를 위한 척도로 제안되었다. 또한 빈번하게 변경되는 웹 문서로 인한 평가 오류를 줄이기 위해, URL 일관성 척도가 제안되었다. 제안된 평가 척도는 표 3에 요약되었다.

URL 정규화는 대부분의 웹 어플리케이션에서 광범위하게 사용되고 있음에도 불구하고, 현재까지도 웹 어플리케이션에 적용될 정규화들이 개발자들의 경험적 지식에 기반하여 선택되고 개발되고 있다. 본 논문에서는 표준 URL 정규화와 확장 URL 정규화를 소개하고 평가하였다. 표준 URL 정규화 방법은 서로 다른 문서를 가리키는 두 URL을 동일하게 변환하지 않고(즉, "잘못된 긍정"이 발생하지 않음) 처리할 URL의 개수를 줄일 수 있었다. 특히 단편 요소를 제거하는 정규화는 272,478개의 URL에 적용되었고 그 중 97%가 중복으로 제거되었다. 확장 URL 정규화 방법은 처리할 URL의 개수가 감소하면서 "잘못된 긍정"이 부분적으로 발생하였다. 실험 결과에 따르면, 경로 요소의 마지막 슬래시 문자를 제거하는 정규화 방법에서 상대적으로 가장 많은 수의 URL 중복이 제거되면서 최소의 "잘못된 긍정"이 발생하였다.

향후 계획은 다음과 같다. 첫째, 효과적인 확장 URL 정규화 방법의 추가 개발에 관한 연구가 필요하다. 둘째, 각 정규화 방법들의 적용 순서와 적용 조합의 효과에 대한 연구가 필요하다. 셋째, 문자열 변환 규칙에 기반하여 URL 정규화하는 방법을 확장하는 연구가 필요하다. 현재, 문서 내용을 이용하여 동일 URL들의 목록을 포함한 정규화 테이블을 만든 후 URL 정규화시에 정규화 테이블을 참조하여 정규화하는 방법의 연구가 진행 중에 있다.

참 고 문 헌

[1] Burner, M., "Crawling Towards Eternity: Building an Archive of the World Wide Web," Web Techniques Magazine, Vol.2, No.5, pp. 37-40, 1997.

표 3 정규화 평가 척도 요약

척도	설명
URL 적용율	임의의 확장 URL 정규화에 의해 변환되는 URL들의 비율
URL 감소율	임의의 확장 URL 정규화 적용 후 감소하는 URL들의 비율
URL 일관성	한 URL로부터 임의의 시간 동안 동일한 문서가 지속적으로 다운로드되는 정도
문서 비손실율	확장 URL 정규화가 적용된 URL 중 비손실이 발생한 URL들의 비율
문서 손실율	확장 URL 정규화가 적용된 URL 중 손실이 발생한 URL들의 비율
문서 획득율	확장 URL 정규화가 적용된 URL 중 획득이 발생한 URL들의 비율
문서 변화율	확장 URL 정규화가 적용된 URL 중 변화가 발생한 URL들의 비율

[2] Kim, S.J. and Lee, S.H., "Implementation of a Web Robot and Statistics on the Korean Web," *Springer-Verlag Lecture Notes in Computer Science*, Vol.2713, pp. 341-350, 2003.
[3] Heydon, A. and Najork, M., "Mercator: A Scalable, Extensible Web Crawler," International Journal of WWW, Vol.2, No.4, pp. 219-229, 1999.
[4] Shkapenyuk, V. and Suel, T., "Design and Implementation of a High-performance Distributed Web Crawler," In Proceedings of 18th Data Engineering Conference, pp. 357-368, 2002.
[5] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax," http://gbiv.com/protocols/uri/rfc/rfc2396.html, 2005.
[6] Lee, S.H., Kim, S.J., and Hong, S.H., "On URL Normalization," Springer-Verlag Lecture Notes in Computer Science, Vol.3481, Part II, pp. 1076-1085, 2005.
[7] Netcraft., "Web Server Survey," http://news.netcraft.com/archives/web_server_survey.html, 2004.

정 효 숙

2004년 건양대학교 컴퓨터학과(학사) 2006년 숭실대학교 대학원 컴퓨터학과(석사). 관심분야는 인터넷 데이타베이스

김 성 진

1998년 숭실대학교 소프트웨어 공학과(학사). 2000년 숭실대학교 대학원 컴퓨터학과(석사). 2004년 숭실대학교 대학원 컴퓨터학과(박사). 2004년~현재 서울대학교 전기컴퓨터공학부, 박사후과정연구원. 관심분야는 인터넷 데이타베이스, 데이타베이스 시스템 성능평가

이 상 호

1984년 서울대학교 전산공학과(학사). 1986년 미국 노스웨스턴대 전산학과(석사) 1989년 미국 노스웨스턴대 전산학과(박사). 1990년~1992년 한국전자통신 연구원, 선임연구원. 1999년~2000년 미국 조지 메이슨대 소프트웨어 정보 공학과 교환 교수. 1992년~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 인터넷 데이타베이스, 데이타베이스 시스템 성능 평가 및 튜닝