사이트 기반의 URL 정규화 평가□

정효숙○ 김성진 이상호
숭실대학교 대학원 컴퓨터학과
서울대학교 전기컴퓨터공학부
숭실대학교 컴퓨터학부
hsjeong@comp.ssu.ac.kr○, sjkim@oopsla.snu.ac.kr, shlee@comp.ssu.ac.kr

Evaluating Site-based URL Normalization

Hyo Sook Jeong○, Sung Jin Kim, Sang Ho Lee

Department of Computing Graduate School, Soongsil University
School of Computer Science and Engineering, Seoul National University
School of Computing, Soongsil University

요 약

URL 정규화는 다양하게 표현된 동일 URL들을 하나의 통일된(cannonical) 형태의 URL로 변환하는 과정이다. 동일 문서에 대한 중복된 URL 표현은 URL 정규화를 통하여 제거된다. 표준 정규화는 잘못된 긍정(동일하지 않는 URL들을 동일 문자열로 변환)이 없도록 개발되었다. 그러나 표준 정규화는 많은 잘못된 부정이 발생하게 되므로, 잘못된 긍정을 일부 허용하면서 잘못된 부정을 현격히 줄일 수 있는 확장 정규화가 제기되고 연구되어 왔다. 본 논문에서는 동일 사이트 내의 URL들에 대한 확장 정규화의 적용 결과가 유사한 정도를 보임으로써, 한 사이트 내의 URL에 대한 임의의 확장 정규화 결과 정보가 동일 사이트 내의 다른 URL들의 정규화에 효과적으로 사용될 수 있음을 보인다. 이를 위하여, 한 사이트의 확장 정규화 결과 동일성 척도와 사이트 기반의 확장 정규화 평가 척도를 제안한다. 20,000만개의 실제 국내 웹 사이트에서 추출된 25만개의 URL에 대해 6가지 확장 정규화가 평가된다.

1. 서 론

URL은 하나의 웹 자원을 가리키는 문자열이다. 하나의 웹 자원(이하 웹 문서)은 다향한 문자열의 URL로 표현이 될 수 있다. 웹에서는 두개의 서로 다른 문자열의 URL이 동일 웹 문서를 가리킬 수도 있으며, 이 경우 두 URL을 동일(equivalent) URL이라고 한다.

URL 정규화는 다양하게 표현된 동일 URL들을 하나의 통일된(cannonical) 형태의 URL로 변환하는 과정이다. URL 정규화는 웹 데이터를 관리하는 다양한 분야에서 사용되고 있다. 예를 들어, 웹 로봇[1,2,3]은 동일 문서의 반복 요청을 피하기 위하여 URL 정규화가 사용된다. 웹 캐시 시스템에서는 동일 문서의 중복 저장을 막기 위하여 사용된다. 링크 정보[4]를 이용한 웹 문서 랭킹에서는 올바른 링크 정보 파악을 위하여 사용된다.

현재 URL 정규화의 표준안을 정의하기 위한 연구가 진행 중이다[5]. "잘못된 부정"은 동일한 두 URL을 동일하지 않다고 판단하는 것이다. "잘못된 긍정"은 비동일 URL을 동일 URL로 판단하는 것이다. 표준 URL 정규화는 "잘못된 긍정"의 발생을 허용하지 않으면서, "잘못된 부정"을 최소화하는 것을 목적으로 한다. [5]은 표준 정규화 방법의 확장 필요성을 제기하고 확장 정규화 방법을 제시하였으며, 4가지 확장 정규화 방안을 제시하였다. [6]은 확장 정규화를 평가할 수 있는 방법을 제시하였다.

본 논문에서는 한 사이트 내의 URL에 대한 임의의 확장 정규화 결과가 동일 사이트내의 다른 URL들의 정규화에 효과적으로 사용될 수 있는 정도를 보이는 것을 목적으로 한다. 이를 위하여, 사이트 기반의 확장 정규화 평가 척도 E를 제안한다. E은 동일 사이트 내의 URL들의 확장 정규화 결과가 서로 동일할수록 높은 값을 갖는다. 마지막으로, 제안된 척도를 통하여 [5,6]에 제안된 확장 정규화 방안을 20,000만개의 실제 웹 사이트에서 추출된 25만개의 URL에 대해 평가한다.

본 논문은 다음과 같이 구성된다. 2장에서는 표준 정규화와 확장 정규화에 관하여 설명한다. 3장에서는 사이트 기반의 확장 정규화를 평가하는 평가 방법을 기술한다. 4장에서는 확장 정규화를 평가한 실험 결과를 기술하고, 5장에서 결론을 맺는다.

2. 관련 연구

URL은 스킴(scheme), 권한(authority), 경로(path), 질의(query), 단편(fragment)의 5개의 구성요소로 구분된다. 스킴에는 웹 서버와 클라이언트 사이에서 통신에 사용될 프로토콜이 기술된다. 권한에는 호스트(host), 사용자 정보, 포트번호의 하위 구조로 구성된다. 경로에는 웹 문서가 위치한 디렉토리와 파일명이 기술된다. 질의에는 웹 문서에 입력되는 파라미터(parameter)들의 이름과 값이 기술된다. 단편은 문서내의 특정 단락을 가리킨다. 그림1은 URL의 구성 예를 보인다.

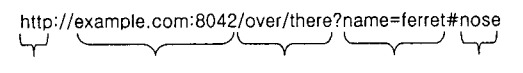


그림 1 URL의 구성 예

[7]에서는 문법 기반 정규화, 스키마 기반 정규화, 프로토콜 기반 정규화의 3가지 유형의 정규화가 기술되어 있다. 첫째, 문법 기반 정규화는 스킴의 종류에 상관없이 URL 자체의 문법에 따라 행해지는 정규화이다. 스킴부와 호스트부는 소문자로 변환되고, 디렉토리를 나타내는 "."과 ".."은 적절한 디렉토리명으로 대체된다. 둘째, 스키마 기반 정규화는 특정 스킴(본 논문에서는 "http")을 고려한 정규화이다. 첫째, 기본 포트(즉 80)가 제거된다. 둘째, 경로가 널(NULL)일 경우 '/'로 대체된다. 셋째, 단편이 존재할 경우 단편이 제거된다. 셋째, 프로토

□ 본 연구는 한국학술진흥재단의 지원에 의하여 수행되었음. (KRF-2004-005-D00172)

콜 기반 정규화는 관례적으로 다양한 형태로 표현될 수 있는 동일 URL들에 대한 정규화이다. 예를 들어, 경로부가 널이 아닐 경우 마지막 슬래시 문자를 제거하는 것이 프로토콜 기반 정규화가 될 수 있다.

표준 정규화는 잘못된 긍정을 허용하지 않으나, [5,6]에서는 잘못된 긍정을 제한된 범위에서 허용하여 다수의 잘못된 부정을 줄이는 것을 목적으로 하는 확장 정규화가 제안하였다. [5, 6]에서 소개된 확장 정규화는 다음과 같다.

URL의 경로부는 대/소문자가 구분된다. 윈도우즈(Windows) 운영체제는 디렉토리와 파일에 대한 대/소문자 구분을 하지 않으며, 유닉스(Unix)나 리눅스(Linux) 운영체제는 디렉토리와 파일 이름에 대한 대소문자를 구분한다. 따라서 경로부 문자를 소문자로 변환하는 정규화가 제안되었다.

마지막이 슬래시로 끝나는 URL은 디렉토리를 나타내는 URL이다. 웹 서버는 디렉토리를 나타내는 URL을 요청 받은 경우, 요청된 디렉토리 내의 기본 문서로 응답하거나 디렉토리가 포함하는 모든 파일을 보여주는 문서를 생성하여 응답한다. 따라서 마지막 슬래시 문자를 제거하는 정규화가 제안되었다.

기본 문서는 웹 클라이언트가 디렉토리를 요청할 때 응답하는 문서이다. 기본 문서로 설정된 파일을 요청하는 URL과 기본 문서를 생략하고 디렉토리를 요청하는 두 URL은 동일하다. 따라서 기본 문서(index.htm, index.html, default.htm)을 제거하는 정규화가 제안되었다.

3. URL 정규화 방법 평가

본 장에서는 확장 정규화 방법의 효과성을 측정하는 방법에 관하여 논한다. 문서 획득에 있어서의 정규화의 효과는 10가지로 구분될 수 있으며 표 1에 나타나 있다. 본래 주어진 URL u1을 URL u2로 정규화 하였고, u1과 u2는 각각 웹 문서 p1과 p2를 가리킨다고 하자.

표 1 정규화 방안의 효과

u2 \ u1		문서 존재		문서 부재
u2가 데이타베이스내에 존재	문서 존재	동일	비손실(1)	비손실(3)
		상이	손실(2)	
	문서 부재	손실(4)		비손실(5)
u2가 데이타베이스내에 부재	문서 존재	동일	비손실(6)	이득(8)
		상이	변화(7) (=손실+이득)	
	문서 부재	손실(9)		비손실(10)

(1) 문서 p1이 웹에 존재

(A) p2가 웹에 부재(경우4, 경우9) : 문서 p1이 손실됨

(B) p2가 웹에 존재, p1과 p2가 동일 문서(경우1, 경우6) : 손실되는 문서가 없고, u1에서 u2중 하나의 URL로만 문서 요청을 수행하므로 문서 요청 횟수의 감소가 가능하다.

(C) p2가 웹에 존재, p1과 p2가 다른 문서 : 두 가지 경우가 있다. 첫째, u2를 웹에서 얻을 수 있는 경우 우리가 이미 알고 있었다면 문서 p1을 잃게 된다. 둘째, u2를 웹에서 얻지 못하는 경우에 정규화를 통해 p2를 새로 획득하며 대신 p1을 잃는다.

(2) 문서 p1이 웹에 부재

(A) u2가 웹 데이터베이스 내 존재(경우3, 경우5) : 손실되는 문서가 없고, u1에서 u2중 하나의 URL로만 문서 요청을 수행하므로 문서 요청 횟수의 감소가 가능하다.

(B) u2가 데이터베이스 내에 부재 : 두 가지 경우가 있다. 첫째, p2가 웹에 존재하면, 새로운 웹 문서 p2를 획득할 수 있다(경우8). 둘째, p2가 웹에 존재하지 않을 경우 p1도 p2도 잃지 않는 것이 된다(경우10).

임의의 확장 정규화에 의해 변경되는 URL 집합을 U라 하고, U에 속한 URL들의 개수를 n(U)라고 하자. 집합 U에 존재하는 사이트들의 개수를 ns(U)라고 하자. U에서 임의의 사이트 i에 속하는 URL들의 집합을 S_i ($1 <= i <= ns(S)$)라고 하자. S_i에 속하는 URL들의 개수는 $n(S_i)$이다. 확장 정규화에 의한 사이트 i의 손실율(Loss Rate, LR_i), 획득율(Gain Rate, GR_i), 변화율(Change Rate, CR_i), 비손실율(Non-loss Rate, NR_i)은 다음과 같이 정의된다. 네 값의 합은 항상 1이며, 평균은 0.25가된다.

- LR_i = i에서 손실한 문서개수 / $n(S_i)$
- GR_i = i에서 획득한 문서개수 / $n(S_i)$
- CR_i = i에서 변화한 문서개수 / $n(S_i)$
- NR_i = i에서 비손실한 문서개수 / $n(S_i)$

임의의 사이트 내에 존재하는 하나의 URL에 대한 정규화 결과가 동일 사이트내의 다른 URL들의 정규화 결과와의 연관성을 측정하기 위하여, 웹 사이트 i의 정규화 결과 동일성(Analogy)이 다음과 같이 정의된다.

$$A_i = 2 \times \sqrt{\frac{(.25 - LR_i)^2 + (.25 - GR_i)^2 + (.25 - C}{3}} \quad (1)$$

A_i는 0에서 1의 값을 가진다. A_i가 0이면, 손실율, 획득율, 변화율, 비손실율이 모두 0.25임을 의미한다. A_i가 1이면, 손실율, 획득율, 변화율, 비손실율 중 하나의 값이 1이고 나머지는 0임을 의미한다. A_i의 값이 높을수록 사이트 i에는 소수의 정규화 결과가 다수 나타나는 것을 의미한다.

확장 정규화의 효과성은 각 사이트 i에 대한 A_i의 평균으로 정의되며 수식 (2)와 같다.

$$E = \sum_{i=1}^{ns(U)} \frac{Ai}{ns(U)} \quad (2)$$

효과성 E의 값이 클수록 사이트 기반의 확장 정규화의 다수의 사이트에서 효과가 있음을 의미한다. 예를 들어, E의 값이 1이면, 모든 사이트의 정규화 결과 동일성 A_i가 1임을 의미한다.

4. 실험 및 결과

본 실험은 6단계로 이루어진다. 1단계는 웹 로봇[2]이 웹 문서를 수집하고 2단계는 수집된 웹 문서에서 원시 URL을 추출한다. 3단계는 간단한 문자열 비교를 통해 중복된 URL을 제거한 후 4단계에서는 이전 단계에서 얻어진 URL들의 집합에서 표준 정규화를 적용한다. 5단계는 확장 정규화를 적용한다. 6단계는 정규화가 적용된 URL들로 웹 문서를 요청한다.

2005년 7월에 20,000개의 한국 웹 사이트로부터 추출된 655,645개의 웹 문서들을 수집하였고, 수집된 웹 문서로부터 실험 대상이 될 25,838,285개의 URL들을 추출하였다. 정규화 없이 문자열 비교만으로 동일성 판별이 가능한 URL들을 제거하여 11,046,159개의 URL을 얻었다. 이후 URL들을 절대 경로로 변환하고 표준 정규화를 적용하여 2,329,770개의 표준 URL들을 얻었다. 웹 문서를 요청할 때마다 다른 내용이 반환되는 URL은 확장 정규화 적용결과의 정확성 유무를 알 수 없으므로 실험대상에서 제외되었고, 2,027,512개의 표준 URL들이 최종 실험 대상 URL로 사용되었다.

질의부의 대소문자 구분은 질의부의 파라미터를 입력으로 받는 웹 프로그램의 대소문자 처리에 따라 결정된다. 본 실험에서는[6]에서 소개된 5개의 확장 정규화와 더불어 질의부를 소문자로 변환하는 확장 정규화를 추가적으로 실험한다. 총 6개의 확장 정규화가 평가되며, EN1을 경로부 소문자 변환 정규화, EN2를 질의부 소문자 변환 정규화, EN3을 마지막 슬래시 문자 제거 정규화, EN4, EN5, EN6을 기본문서 "index.htm", "index.html", "default.htm" 제거 정규화로 표기한다.

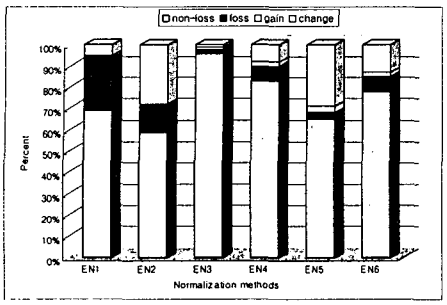
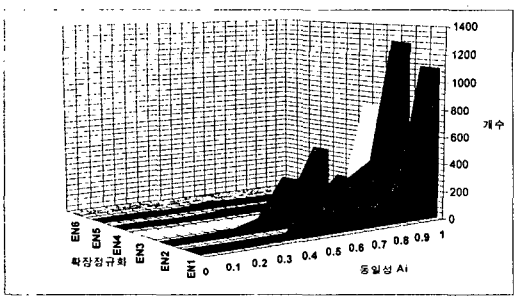
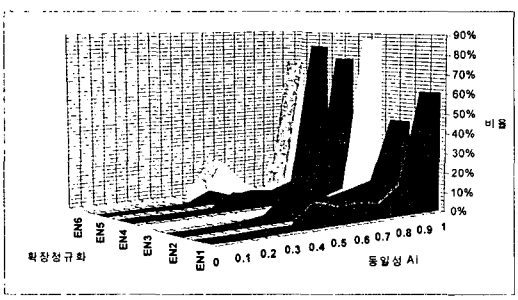


그림 2 확장 정규화의 문서 손실/획득/변화/비손실율

그림 2 은 각 확장 정규화가 적용되는 URL들에 대하여 문서 손실율, 획득율, 변화율, 비손실율의 비율을 나타낸다. 문서 비손실율은 확장 정규화에서 잘못된 긍정이 없음을 의미한다. 문서 손실/획득/변화는 잘못된 긍정이 일어났음을 의미한다. EN3 정규화 사용 시에 적용되는 URL의 95%가 잘못된 부정 없이 변환되었다. EN2 정규화 사용 시에는 41%의 잘못된 긍정이 발생하였다.



(가) 개수



(나) 비율

그림 3 확장 정규화별 사이트의 A_i의 분포

그림 3은 6가지 확장 정규화를 적용하였을 때, 각 사이트들의 A_i값을 11개의 계급 구간(0이상 0.1미만, 0.1이상 0.2미만, ... 0.9이상 1미만, 1)으로 나누어, 각 계급구간에 속한 사이트들의 개수와 비율 분포를 나타낸다. 모든 정규화에서 많은 사이트들의 A_i가 1로 나타났다 (각 정규화 별로 61%, 45%, 86%, 76%, 82%, 73%).

동일성의 높고 낮음은 0.88을 기준으로 하였다. 임의의 사이트의 LR_i, GR_i, CR_i, NR_i 중 하나의 값이 0.91보다 클 때, A_i가 0.88이상의 값을 가진다. 즉, 사이트내의 91%의 URL이 동일한 확장 정규화 적용 결과를 보일 때, A_i의 값이 0.88보다 크다. 각 확장 정규화에 대해 74%, 60%, 94%, 82%, 88%, 73% 사이트가 0.88 이상의 값을 가졌다. 세 개의 확장 정규화 EN3, EN4, EN5에서 상대적으로 높은 A_i의 값을 가지는 사이트들이 많이 나타났다. 즉, 세 가지 확장 정규화는 동일 사이트 내의 URL들에 대한 정규화 결과가 상대적으로 비슷하였다. EN1, EN2, EN6에서는 낮은 A_i값을 가지는 사이트들이 다소 발견되었다. 이는 EN1, EN2, EN6 정규화 적용 시 동일 사이트 내에서도 정규화 적용 결과가 다를 수 있는 경우가 다소 나타날 수 있음을 나타낸다.

표 2 사이트 기반의 확장 정규화 평가

확장 정규화	E	최소 A_i	사이트 수	사이트 당 평균 URL 수
EN1	0.91	0.38	1823	143
EN2	0.86	0.19	2832	102
EN3	0.98	0.41	908	24
EN4	0.93	0.41	234	13
EN5	0.96	0.45	456	18
EN6	0.91	0.60	15	7

표 2는 6개의 확장 정규화의 평가 결과가 나타낸다. EN3의 효과성 E가 0.98로 가장 높게 나타났다. EN3는 908개의 사이트에서 각 사이트 당 평균 24개의 URL들이 적용되었다. EN3 적용 시 최소 결과 동일성 A_i로 0.41을 가진 사이트가 존재하였으나, 각 사이트들은 평균적으로 0.98의 A_i 값을 나타내었다.

5. 결론

본 논문은 사이트에 기반 하여 문서 손실율, 문서 획득율, 문서 변화율, 문서 비손실율, 문서 감소율을 측정하고 각 정규화 결과의 동일성을 평가하는 척도를 제안하였다. 또한 웹에서 추출된 실제 URL들을 통하여 확장 정규화들의 평가되었으며, 어플리케이션에서 적용하기에 충분한 결과를 나타냈다. 다양한 확장 정규화가 개발될 수 있다. 확장 정규화는 개발자들의 경험적인 요인에 의하여 개발되는 경우가 다수이며, 개발된 확장 정규화를 분석적으로 평가하는 방법이 부재하였다. 특히 본 논문은 한 사이트 내의 URL에 대한 정규화 결과를 통하여 동일 사이트내의 다른 URL들을 정규화 할 때 잘못된 긍정을 줄일 수 있음을 실증하였다.

참고문헌

[1] A. Heydon and M. Najork, Mercator: A Scalable, Extensible Web Crawler, International Journal of WWW, Vol. 2, No. 4, pages 219-229, 1999.

[2] S.J. Kim and S.H Lee, Implementation of a Web Robot and Statistics on the Korean Web, Springer-Verlag Lecture Notes in Computer Science, Vol. 2713, pages 341-350, 2003.

[3] V. Shkapenyuk and T. Suel, Design and Implementation of a High-performance Distributed Web Crawler, In Proceedings of 18th Data Engineering Conference, pages 357-368, 2002.

[4] S. Brin and L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, In Proceedings of 7th International World Wide Web Conference (WWW7), Vol. 30, No. 1-7, pages 107-117, 1998.

[5] S.H. Lee, S.J. Kim, and S.H. Hong, On URL Normalization, Springer-Verlag Lecture Notes in Computer Science, Vol. 3481, Part II, pages 1076-1085, 2005.

[6] S.H. Lee, S.J. Kim, and H.S. Jeong, How to Evaluate the Effectiveness of URL Normalizations, Spring-Verlag Lecture Notes in Computer Science, Vol. 3597, pages 228-237, 2005.

[7] T. Berners-Lee, R. Fielding and L. Masinter, Uniform Resource Identifiers (URI): Generic Syntax, http://gbiv.com/protocols/uri/rev-2002/rfc2396bis.html, 2004.