효과적인 웹 문서 변경도 측정 방법
(An Effective Metric for Measuring the Degree of Web Page Changes)

권 신 영 † 김 성 진 †† 이 상 호 †††
(Shin Young Kwon) (Sung Jin Kim) (Sang Ho Lee)

요 약 다양한 유사도 측정 방법들이 웹 문서의 변경도 측정에 사용되어 왔다. 본 논문은 여섯 가지 웹 문서 변경 종류에 근거하여 변경도 측정 방법의 효과성 평가 척도를 정의하고, 새로운 유사도 측정 방법을 제안한다. 실제 웹 문서들과 인위적 문서들을 사용하여, 기존의 다섯 가지 측정 방법들(바이트 비교, TF·IDF 코사인 거리, 단어 거리, 편집 거리, 쉥글링)과 제안된 측정 방법을 비교 평가한다. 실험 결과 분석을 통해 제안된 측정 방법이 웹 문서의 변경 측정에 효과적임을 보인다. 본 연구는 웹 문서의 변화 정보를 필요로 하는 웹 응용 분야에서 웹 문서 변경도 측정 방법의 적합한 선택을 위한 지침이 될 수 있다.

키워드 : 웹 데이타베이스, 웹 데이타베이스 관리, 웹 문서 변경

Abstract A variety of similarity metrics have been used to measure the degree of web page changes. In this paper, we first define criteria for web page changes to evaluate the effectiveness of the similarity metrics in terms of six important types of web page changes. Second, we propose a new similarity metric appropriate for measuring the degree of web page changes. Using real web pages and synthesized pages, we analyze the five existing metrics (i.e., the byte-wise comparison, the TF·IDF cosine distance, the word distance, the edit distance, and the shingling) and ours under the proposed criteria. The analysis result shows that our metric represents the changes more effectively than other metrics. We expect that our study can help users select an appropriate metric for particular web applications.

Key words : web databases, web database management, web page changes

1. 서 론

웹 데이타베이스(웹 문서들의 집합)는 많은 웹 응용 분야에서 구축되어 관리된다. 예를 들어, 웹 검색 서비스(Google[1], Yahoo[2] 등)는 사용자들의 정보 검색을 위하여 웹 데이타베이스를 구축하여 관리한다. 프락시 서버와 웹 브라우저는 웹 문서 캐시의 목적으로 웹 데이타베이스를 유지한다. 웹 문서들은 지속적으로 변화하며, 웹 데이타베이스 내의 웹 문서들 또한 원본 웹 문서들의 변화를 반영하도록 갱신될 필요가 있다[3]. 웹 데이타베이스 내의 모든 문서를 갱신하는 것은 변경되지 않은 문서 수집에 따른 불필요한 문서 갱신, 네트워크 부하 증가와 같은 자원 낭비를 야기할 수 있다. 웹 데이타베이스의 효과적 갱신을 위하여, 실제 웹 문서들의 변화에 관한 연구가 다수 진행되고 있다[4-8].

[6,7]은 웹 문서의 변경 전후를 바이트 단위로 비교하여 변경 주기 및 분포를 조사하였다. [8]은 편집 거리(edit distance) 기반의 문서 거리 식을 정의하여 웹의 변경도 분포를 조사하는데 사용하였다. [4]은 쉥글링(shingling) 방법을 사용하여 대량의 웹 문서 집합을 대상으로 웹의 변화를 조사하였다. [5]는 정보 검색 분야에서 널리 사용되는 TF·IDF 코사인 거리(term frequency·inverse document frequency cosine distance)와 단어 거리(word distance) 방법을 사용하여 검색 엔진 관점에서 웹의 변화를 연구하였다. 기존 웹 문서 변경도 측정 방법들은 동일한 변경을 서로 다르게 표현할 수 있기 때문에, 변경도를 정확하게 표현할 수 있는 측정 방법의 선택이 중요하다. 현재까지 웹 문서 변경도 측정 방법들의 비교 평가에 관한 연구는 부재하다.

본 논문에서는 웹 문서 변경도 측정 방법의 효과성 평가를 위한 척도를 정의한다. 정의된 척도는 웹 문서의 변경을 추가(add), 복사(copy), 제거(drop), 감소(shrink),

† 정 회 원 : 숭실대학교 대학원 컴퓨터학과
shin025@naver.com
†† 학생회원 : 서울대학교 컴퓨터학과 박사후과정 연구원
sjkim@oopsla.snu.ac.kr
††† 종신회원 : 숭실대학교 컴퓨터학부 교수
shlee@comp.ssu.ac.kr
논문접수 : 2005년 12월 27일
심사완료 : 2007년 5월 29일

대체(replace), 이동(move)의 여섯 가지 종류로 분류하여, 각 변경 종류에 대해 평가 기준이 되는 변경도 값을 제시한다. 임의의 측정 방법이 주어진 변경에서 척도보다 높은 변경도를 나타내면 민감도가 높다고 하며, 반대로 낮은 변경도를 나타내면 민감도가 낮다고 한다. 본 논문은 평가 척도 정의에 이어서, 새로운 웹 문서 변경도 측정 방법을 제안한다. 제안된 측정 방법은 편집 거리 알고리즘을 기반으로 하여, 웹 문서의 여섯 가지 변경 종류를 여섯 가지의 편집 연산(edit operation)으로 반영하도록 고안되었다.

본 논문에서는 두 가지 실험이 수행되었다. 첫 번째 실험은 실제 웹 문서들의 변경에 대해 기존의 웹 문서 변경도 측정 방법(바이트 단위 비교, TF · IDF 코사인 거리, 단어 거리, 편집 거리, 슁글링)과 제안한 측정 방법의 변경도 표현 차이를 보인다. 두 번째 실험에서는 각 측정 방법의 효과성을 제안된 척도로 평가한다. 평가 결과를 통해 측정 방법들의 특징을 분석하고, 제안된 측정 방법의 효과성을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 사용된 웹 문서 변경도 측정 방법들을 간략히 소개한다. 3장에서는 웹 문서 변경을 여섯 가지 종류로 분류하고, 각 변경 종류에 대해 척도를 정의한다. 4장에서는 3장에서 정의된 척도를 잘 반영하도록 고안된 새로운 웹 문서 변경도 측정 방법을 제안한다. 5장에서는 실험을 통해 각 측정 방법의 다양성을 보이고, 정의된 척도에 따라 이들의 효과성을 평가한다. 끝으로 6장에서는 결론과 함께 향후 계획을 기술한다.

2. 관련 연구

본 장에서는 웹 문서 변경도 측정에 사용될 수 있는 다섯 가지 문서 비교 방법을 소개한다. 본 논문에서는 하나의 웹 문서를 p로, p의 변경 후 문서를 p'으로 표기한다. 소개된 측정 방법들은 웹 문서의 변경도를 0에서 1 사이의 값으로 반환한다. [6,7]에서 사용된 바이트 단위 비교 방법은 변경 전후의 웹 문서를 바이트 단위로 비교하여 1 바이트(byte)라도 다르면 1을, 모두 같으면 0을 반환하는 가장 간단한 방법이다. 대부분의 웹 문서가 부분적으로 변화함에도 불구하고 작은 변경에도 1을 반환하므로, 이 방법은 웹 문서의 변경도를 표현하지 못한다.

TF · IDF 코사인 거리 방법은 정보 검색 분야에서 내용 기반으로 질의에 가장 적합한 문서를 결정하는데 주로 사용된다. 이 방법은 두 웹 문서 p와 p'을 TF · IDF 가중치 벡터 v_p, $v_{p'}$으로 변환하여[9], 두 벡터간 코사인 거리를 식 (1)과 같이 계산한다. $v_p \cdot v_{p'}$은 두 벡터 v_p, $v_{p'}$의 내적을, $\|v_i\|_2$는 벡터 v_i의 크기(second norm)를 의미한다.

$$D_{COS}(p,p') = 1 - \frac{v_p \cdot v_{p'}}{\|v_p\|_2 \|v_{p'}\|_2} \quad (1)$$

단어 거리 방법은 한 문서에서 얼마나 많은 단어가 변경되었는지 그 수를 계산하는 방식이다. 두 웹 문서 p와 p' 사이의 단어 거리는 식 (2)와 같이 정의된다. m, n은 각각 p와 p' 내에 존재하는 단어 개수를 의미한다. 단어 거리 방법은 TF · IDF 코사인 거리 방법과 함께 [5]에서 사용되었다.

$$D_{WD}(p,p') = 1 - \frac{2 \cdot |common\ words|}{m+n} \quad (2)$$

TF · IDF 코사인 거리 방법과 단어 거리 방법은 각각의 웹 문서를 단어들의 중복 허용 집합(bag of words)으로 간주하기 때문에, 단어순서의 변경을 고려하지 못한다. 실제 웹에서 단어순서의 변경은 매우 중요할 수 있다. 예를 들어, 쇼핑몰 사이트에서 판매하는 상품의 인기도가 변화하면, 해당 상품을 나타내는 단어순서 역시 변경된다. 이러한 변경은 쇼핑몰 회사의 판매 전략에 큰 영향을 끼칠 수 있기 때문에 중요한 변경으로 고려되어야 할 것이다.

편집 거리란 하나의 시퀀스를 다른 시퀀스로 변환하는데 요구되는 편집 연산 시퀀스의 최소 비용을 말한다[10]. 편집 연산 시퀀스란 삽입(insertion), 삭제(deletion), 치환(substitution)과 같은 편집 연산으로 구성된 시퀀스이며, 편집 연산 시퀀스의 비용이란 시퀀스 내 각 편집 연산 비용의 총 합을 의미한다. 예를 들어, 편집 연산의 비용이 모두 1이고 시퀀스 $<A,G,B,A,A>$가 $<A,B,A,T,A>$로 변경될 때, 최소 비용 편집 연산 시퀀스는 <G의 삭제, T의 삽입>이 되며 편집 거리는 2가 된다. [8]은 웹 문서의 변경도를 측정하기 위해 웹 문서를 단어 시퀀스로 간주하였으며 편집 거리 기반의 측정 방법을 식 (3)과 같이 정의하였다. m과 n은 각각 p와 p' 내에 존재하는 단어의 개수를, δ는 p와 p' 사이의 편집 거리를 의미한다. 편집 거리 δ를 구할 때 두 개의 편집 연산(삽입, 삭제)을 사용하였으며, 두 연산의 비용은 각각 1로 설정하였다. 만약 두 문서가 동일하다면 δ는 0이 되어 문서 간 최종 거리 역시 0이 된다. 반면에 두 문서가 완전히 다르다면 m개의 단어가 삭제된 후 n개의 단어가 삽입되므로 δ는 $m+n$이 되어 최종 거리는 1이 된다.

$$D_{ED}(p,p') = \frac{\delta}{m+n} \quad (3)$$

단어 거리 방법과 편집 거리 방법은 유일한 단어의 삽입(또는 삭제)과 중복 단어의 삽입(또는 삭제)을 구별

하지 못한다. 예를 들어, 웹 문서 $<w_1, w_2, w_3, w_4>$가 $<w_1, w_2, w_3, w_4, w_2, w_3>$으로 변경될 때 두 측정 방법은 모두 0.2를 반환하며, $<w_1, w_2, w_3, w_4>$가 $<w_1, w_2, w_3, w_4, w_5, w_6>$으로 변경되어도 역시 동일하게 0.2를 반환한다. 일반적으로 사람들은 중복되는 정보의 변경보다는 유일한 정보의 변경에 더 관심을 갖기 때문에 이러한 변경은 구별될 필요가 있다.

슁글링 방법에서 각 문서는 k개의 연속된 단어 시퀀스 - "슁글(shingle)" - 의 집합으로 표현되며, 이러한 집합을 "k-슁글링"이라 한다. 하나의 문서로부터 슁글 집합을 생성할 때 문서의 끝을 포장(wrapping)하여 문서 내 모든 단어들이 각 슁글의 첫 번째 단어로 나타나도록 한다. 예를 들어, 문서 $<w_1, w_2, w_3, w_4>$의 3-슁글링은 $\{<w_1, w_2, w_3>, <w_2, w_3, w_4>, <w_3, w_4, w_1>, <w_4, w_1, w_2>\}$이다. 슁글링 방법을 사용한 p와 p' 사이의 거리는 식 (4)와 같다. $S(p)$는 문서 p의 k-슁글링이며, $|S(p)|$는 집합 $S(p)$ 내에 존재하는 원소, 즉 슁글의 개수이다.

$$D_{k-SH}(p, p') = 1 - \frac{|S(p) \cap S(p')|}{|S(p) \cup S(p')|} \quad (4)$$

슁글링 방법은 작은 크기의 웹 문서 변경에 민감한 단점을 가지고 있다. 예를 들어, 5개의 단어로 구성된 웹 문서 $<w_1, w_2, w_3, w_4, w_5>$가 $<w_1, w_2, w_3, w_6, w_5>$로 변경되었고, 이러한 변경을 k값이 3인 슁글링 방법으로 측정한다고 가정하자. 하나의 단어(w_4)가 변경되었음에도 불구하고, 결과는 0.75로 반환된다. 이는 사용자로 하여금 주어진 웹 문서가 75%나 변경되었다는 오해를 갖게 한다. 다른 예로, $<w_1, w_2, w_3, w_4, w_5>$가 $<w_2, w_1, w_3, w_4, w_5>$로 변경된다면, 단지 하나의 단어 w_1이 w_2 다음 위치로 이동한 것임에도 불구하고 변경도 1을 반환하게 된다.

3. 웹 문서 변경의 평가 척도

본 장에서는 웹 문서 변경도 측정 방법의 효과성을 평가하기 위한 척도를 제안한다. 평가 척도는 웹 문서의 변경을 추가, 복사, 제거, 축소, 대체, 이동의 여섯 가지 종류로 분류 한다. 각 변경 종류에 대한 예는 그림 1에 나타나 있다.

정의 1. 문서 p에 존재하지 않던 새로운 단어가 삽입될 때, 이를 "추가" 변경이라 한다. 문서 p에 존재하던 기존 단어가 삽입될 때, 이를 "복사" 변경이라 한다.

정의 2. 문서 p에 유일하게 존재하는 단어가 삭제될 때, 이를 "제거" 변경이라 한다. "제거"된 단어는 변경된 문서 p'에 더 이상 출현하지 않게 된다. 문서 p에 중복하여 존재하는 단어가 삭제될 때, 이를 "축소" 변경이라 한다. "축소"된 단어는 변경된 문서 p'에 여전히 출현하게 된다.

정의 3. 문서 p에 존재하는 단어가 다른 단어로 교체될 때, 이를 "대체" 변경이라 한다.

정의 4. 문서 p에 존재하는 단어의 위치가 변경될 때, 이를 "이동" 변경이라 한다.

정의된 여섯 가지 변경 종류가 실제 웹에서 어떤 형태로 존재하는지 알아본다. 도서를 판매하는 전자상거래 사이트에서, 도서 상품의 제목, 요약, 가격, 인기도, 고객 의견의 정보를 인기도 순서로 정렬하여 보여주는 웹 문서를 가정하자. 고객은 각 도서에 대한 의견을 웹 문서에 기록할 수 있다. 기록된 의견이 기존의 내용들과 다를 경우 삽입된 단어의 대부분은 문서 내에 존재하지 않는 새로운 단어일 것이며(추가 변경), 유사한 의견일 경우 삽입된 단어들의 대부분은 이미 존재하는 단어일 것이다(복사 변경). 고객이 기록된 의견을 삭제한다고 하자. 삭제된 의견이 삭제되지 않은 다른 의견들과 상이한 내용이면, 삭제된 단어의 대부분은 더 이상 문서 내에 존재하지 않을 것이다(제거 변경). 삭제된 내용이 다른 의견과 유사한 내용이면, 대부분의 삭제된 단어들은 여전히 해당 문서에 존재할 것이다(축소 변경). 일반적으로 유일한 정보의 변경이 중복되는 정보의 변경보다

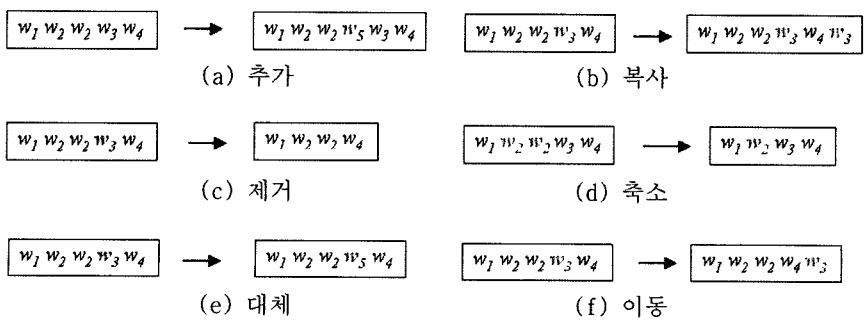


그림 1 웹 문서 변경의 여섯 가지 종류

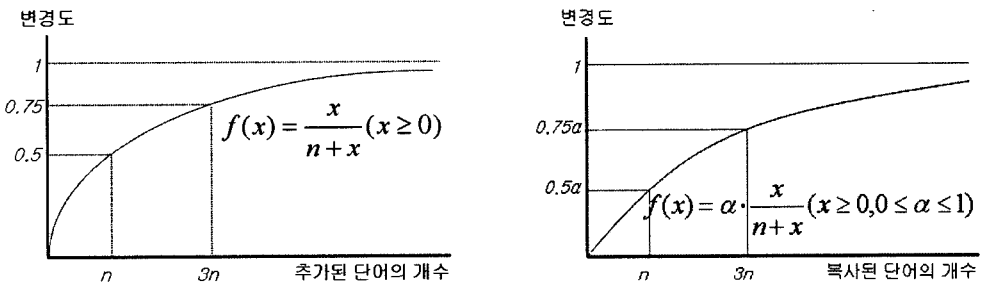
더 큰 의미를 가지기 때문에, 내용 삽입을 추가와 복사로(또한, 내용 삭제를 제거와 축소로) 구분할 필요가 있다. 도서 가격이 변동되면 기존 가격을 나타내는 단어는 새로운 가격을 나타내는 단어로 교체되게 되며(대체 변경), 도서의 인기도 변경은 도서 목록의 출력 순서에 변화를 주고 도서 정보를 나타내는 단어들의 위치 변경을 야기하게 된다(이동 변경).

그림 2, 3, 4는 위에서 분류된 여섯 가지 웹 문서 변경 종류 각각에 대한 척도를 나타낸다. x축은 변경된 단어의 개수를, y축은 해당 문서에 대한 변경도를 나타낸다. 각 변경 종류에 대한 척도 정의 식에서, n은 문서 p에 존재하는 전체 단어의 개수를, x는 변경되는 단어의 개수를 의미한다.

추가 변경에 대한 척도는 그림 2(a)와 같이 $x/(n+x)$로 정의한다. 예를 들어, n개 단어를 가진 문서 p에 n개의 단어가 추가될 때 척도는 $0.5(=n/(n+n))$가 된다. 같은 방법으로 n개 단어를 가진 문서 p에 $3n$개의 단어가 추가되면 척도는 $0.75(=3n/(3n+n))$가 된다. 추가되는 단어의 수가 증가할수록 척도는 1에 가까워진다. 복사 변경에 대한 척도는 그림 2(b)와 같이 $\alpha x/(n+x)$로 정의한다. α는 추가에 대한 복사의 가중치를 나타내는 사용자 정의 파라미터로, 0에서 1 사이의 값을 갖는다. 사용자가 복사 변경을 중요하게 생각할수록 α값을 크게 설정하며, 반대의 경우 작게 설정한다. 예를 들어, 어떤 사용자가 한 단어의 추가와 두 단어의 복사를 동일 변경으로 간주하면 α값을 1/2로 정한다. 만약 한 단어의 추가를 세 단어의 복사와 동일하게 간주하면 α값을 1/3로 한다.

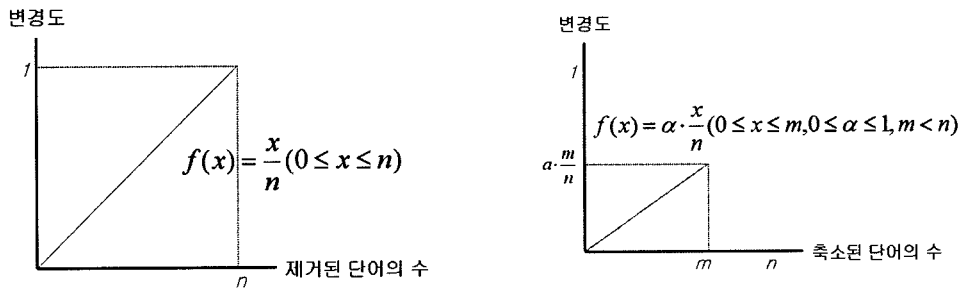
제거 변경에 대한 척도는 그림 3(a)와 같이 x/n로 정의한다. 예를 들어, n개 단어를 갖는 문서 p에서 n개의 단어가 제거될 때, 제거될 수 있는 최대 단어 수는 n이기 때문에 척도는 $1(=n/n)$이 된다. 축소 변경에 대한 척도는 그림 3(b)와 같이 $\alpha x/n$으로 정의한다. α는 복사 변경에서 정의되었던 사용자 정의 파라미터이며, 제거에 대한 축소의 가중치로서 재사용된다. 즉, 본 척도에서는 복사 변경과 축소 변경의 가중치가 항상 같다. m은 문서 p에 존재하는 중복 단어의 최대 개수를 의미한다. 따라서 축소될 수 있는 단어의 최대 개수는 m이 된다.

대체 변경에 대한 척도는 그림 4(a)와 같이 x/n로 정의한다. 예를 들어, n개 단어를 갖는 문서 p에서 n개, 즉 전체 단어가 다른 단어로 교체될 때 척도는 1 $(=n/n)$이 된다. 이동 변경에 대한 척도는 그림 4(b)와 같이 $\beta x/n$으로 정의한다. β는 대체에 대한 이동의 가중치를 나타내는 사용자 정의 파라미터이며, 0에서 1 사이의 값을 갖는다. 사용자가 이동 변경을 의미 있게 생



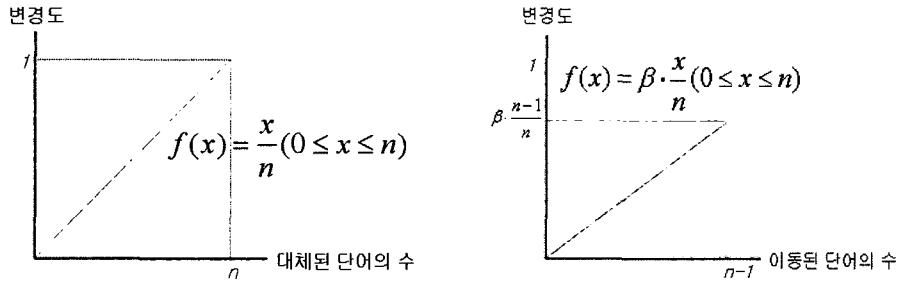
(a) 추가 (b) 복사

그림 2 "추가", "복사" 변경에 대한 척도



(a) 제거 (b) 축소

그림 3 "제거", "축소" 변경에 대한 척도



(a) 대체 (b) 이동

그림 4 "대체", "이동" 변경에 대한 척도

각할수록 β값은 점점 커진다. n개 단어를 갖는 문서에서 이동 가능한 최대 단어 수는 $n-1$이므로, 이동 변경에 대한 척도의 최대값은 $\beta\cdot\frac{n-1}{n}$이 된다.

정의된 척도는 다음 네 가지 특성을 갖는다.

특성 1. 웹 문서의 변경도는 항상 0에서 1 사이의 값으로 표현된다.

특성 2. 웹 문서 내에 변경되는 단어의 수가 많을수록 변경도가 커진다.

특성 3. 모든 단어는 의미적으로 동일한 중요도를 갖는다.

특성 4. 어떤 웹 문서 p가 p'으로 변경되었을 때의 변경도와 p'이 p로 변경되었을 때의 변경도는 같다.

4. 개선된 편집 거리(Improved Edit Distance) 연산

본 장에서는 기존의 편집 거리 방법을 확장하여 웹 문서의 여섯 가지 변경 종류를 효과적으로 표현할 수 있는 변경도 측정 방법을 제안한다. 제안된 측정 방법은 "개선된 편집 거리(improved edit distance: IED)"라 명명된다. IED는 고전적인 편집 거리 알고리즘(2장 참고)에서 사용하는 삽입(insertion), 삭제(deletion), 치환(substitution)의 세 가지 편집 연산을 추가(add), 복사(copy), 제거(drop), 축소(shrink), 대체(replace), 이동(move)의 여섯 가지 연산으로 확장한다. 각 편집 연산의 비용은 차례대로 1, s, 1, s, 1, t 이다. s와 t는 0에서 1 사이의 값을 가지며 사용자에 의해 조절될 수 있다. 확장된 편집 연산을 이용한 두 문서 사이의 확장 편집 거리(extended edit distance) δ_E는 식 (5)와 같다. k는 하나의 문서를 다른 문서로 변환할 수 있는 확장 편집 연산 시퀀스들의 경우의 수를 의미한다. $COST_i(op)$는 i번째 확장 편집 연산 시퀀스의 비용을 뜻한다.

$$\delta_E=\min_{i=1..k}\left\{\sum_{op\in\{add,\ drop,\ copy,\ shrink,\ replace,\ move\}}COST_i(op)\right\} \quad (5)$$

두 시퀀스 A와 B 사이의 확장 편집 거리 δ_E는 다음 과정을 통해 구한다.

단계 1 : 최장 공통 서브시퀀스(the longest common subsequence) 탐색

두 단어 시퀀스 A와 B로부터 최장 공통 서브시퀀스 $LCS(A,B)$를 구한다. 어떤 시퀀스에 대한 서브시퀀스란 주어진 시퀀스에서 일부 원소가 제거되거나 제거된 원소가 없는 시퀀스를 말한다. 예를 들어, 시퀀스 $<w_2,w_3,w_4,w_2>$는 시퀀스 $<w_1,w_2,w_3,w_2,w_4,w_1,w_2>$에서 2, 3, 5, 7번째 단어들로 구성된 서브시퀀스이다. $LCS(A,B)$를 구한 후, 두 단어 시퀀스 A, B로부터 각각 $LCS(A,B)$를 제외한 시퀀스 A', B'을 생성한다.

단계 2 : 이동 연산의 고려

이동 연산이 적용될 후보 단어(CW_{move})는 A'과 B'에 공통으로 존재하는 단어가 된다. 만약 t(이동 비용)가 $2s$(복사 비용과 축소 비용의 합)보다 크면, A와 B 각각에서 중복하여 존재하는 단어들을 CW_{move}로부터 제거한다. 제거된 단어들은 다음 단계에서 복사 연산 1회, 축소 연산 1회로써 처리되는데, 이는 확장 편집 연산 시퀀스의 비용을 최소로 하기 위함이다. 남아있는 CW_{move}는 모두 이동된 단어로 간주된다. 이동 연산의 수는 이동된 단어의 수가 되며, CW_{move}는 A'과 B'으로부터 제거된다.

단계 3 : 복사, 축소 연산의 고려

복사 연산이 적용될 후보 단어(CW_{copy})는 B'에 존재하면서 B에 중복 존재하는 단어가, 축소 연산에 의해 처리될 후보 단어(CW_{shrink})는 A'에 존재하면서 A에 중복 존재하는 단어가 된다. 먼저, CW_{copy}와 CW_{shrink}에 공통으로 존재하는 단어가 축소 후 복사된 단어로 간주된다. 이러한 단어는 앞 단계에서 처음 CW_{move}에서 제거된 단어이기 때문에, t가 $2s$보다 클 경우에만 존재한다. 따라서 복사, 축소 연산의 수는 각각 CW_{copy}와 CW_{shrink}의 공통 단어 수가 되며, 해당 단어는 A'과

B', CW_{copy}와 CW_{shrink}로부터 제거된다. 다음으로, CW_{copy}와 CW_{shrink}로부터 각각 B와 A에서 동일한 위치에 존재하는 단어가 있는지 확인한다. 다시 말해, CW_{copy} 각 단어의 B에서의 위치와 CW_{shrink} 각 단어의 A에서의 위치를 비교하여 서로의 위치가 같은지 확인한다. 만약 s가 0.5보다 작으면, 해당 단어들은 모두 축소 후 복사된 단어로 간주된다. 대체 연산 1회보다 축소, 복사 연산 각 1회의 비용이 더 작기 때문이다. 축소 연산과 복사 연산의 수는 각각 축소, 복사된 단어의 수만큼 증가되고, 해당 단어는 A'과 B', CW_{copy}와 CW_{shrink}로부터 제거된다. s가 0.5보다 크면, 해당 단어들은 다음 단계에서 대체된 단어로 간주되므로 CW_{copy}와 CW_{shrink}로부터만 제거된다. 끝으로, 남겨진 CW_{copy}와 CW_{shrink}가 모두 복사 및 축소된 단어로 간주되어 A'과 B'으로부터 제거된다. 복사, 축소 연산의 수는 CW_{copy}와 CW_{shrink} 단어의 수만큼 증가된다.

단계 4 : 대체, 추가, 제거 연산의 고려

A'과 B'에 남아있는 단어들이 각각 A와 B에서 동일한 위치에 있는지의 여부를 확인한다. 해당 단어는 대체된 단어로 간주되어 A'과 B'으로부터 제거되며, 대체 연산의 수는 해당 단어의 수가 된다. 다음으로, A'과 B'에 남아있는 단어가 각각 제거, 추가된 단어로 간주된다. 따라서 제거 연산의 수는 A' 내에 존재하는 단어 개수가, 추가 연산의 수는 B' 내에 존재하는 단어 개수가 된다.

단계 5 : 확장 편집 거리 계산

앞 단계에서 구하여진 각 확장 편집 연산의 수와 비용으로부터, 두 시퀀스 A와 B 사이의 확장 편집 거리를 계산한다.

예제 1. 어떤 웹 문서 $A=<w_1, w_2, w_2, w_2, w_3, w_3, w_4, w_5, w_2>$가 $B=<w_3, w_1, w_4, w_2, w_3, w_5, w_5, w_6, w_6, w_7>$로 변경되었다고 가정하자. s와 t의 값이 각각 0.4, 0.9일 때, A와 B 사이의 확장 편집 거리는 다음과 같이 계산된다.

단계 1에서, A와 B의 최장 공통 서브시퀀스를 구하고, A'과 B'을 생성한다.

$LCS(A, B) = <w_1, w_2, w_3, w_5>$

$A' = <w_2, w_2, w_3, w_4, w_2>$

$B' = <w_3, w_4, w_5, w_6, w_6, w_7>$

$|add| = 0, |drop| = 0, |copy| = 0, |shrink| = 0,$
$|replace| = 0, |move| = 0$

단계 2에서, CW_{move}는 A'과 B'에 공통적으로 존재하는 w_3, w_4가 된다. 그러나 w_3은 A, B 모두에서 중복 존재하며 t가 $2s$보다 크기 때문에 CW_{move}에서 제외된다. 따라서 w_4 한 단어만 이동된 단어로 간주되어, 이동 연산의 개수는 1이 되고 w_4는 A', B'으로부터 제거된다.

$A' = <w_2, w_2, w_3, w_2>$

$B' = <w_3, w_5, w_6, w_6, w_7>$

$|add| = 0, |drop| = 0, |copy| = 0, |shrink| = 0,$
$|replace| = 0, |move| = 1$

단계 3에서, CW_{copy}는 w_3, w_5, w_6이 되며 CW_{shrink}는 w_2, w_2, w_2, w_3이 된다. 먼저, w_3은 CW_{copy}와 CW_{shrink}의 공통 단어이므로, 축소 후 복사된 단어로 간주된다. 따라서 복사와 축소 연산은 각각 1이 되고 w_3은 A', B', CW_{copy}, CW_{shrink}으로부터 제거된다.

$A' = <w_2, w_2, w_2>$, $CW_{shrink} = w_2, w_2, w_2$

$B' = <w_5, w_6, w_6, w_7>$, $CW_{copy} = w_5, w_6$

$|add| = 0, |drop| = 0, |copy| = 1, |shrink| = 1,$
$|replace| = 0, |move| = 1$

다음으로, CW_{copy}와 CW_{shrink}로부터 각각 B, A에서 같은 위치에 있는 단어를 찾는다. w_2는 A에서 9번째에 위치하며 w_6은 B에서 9번째 위치에 존재한다. 이 때 s가 0.5보다 작으므로, w_2는 축소된 단어로, w_6은 복사된 단어로 간주된다. 따라서 복사, 축소 연산의 수는 1씩 증가하고, w_2는 A'과 CW_{shrink}로부터, w_6은 B'과 CW_{copy}로부터 제거된다.

$A' = <w_2, w_2>$, $CW_{shrink} = w_2, w_2$

$B' = <w_5, w_6, w_7>$, $CW_{copy} = w_5$

$|add| = 0, |drop| = 0, |copy| = 2, |shrink| = 2,$
$|replace| = 0, |move| = 1$

다음으로, CW_{copy}와 CW_{shrink}는 각각 복사, 축소된 단어로 간주된다. 따라서 복사 연산은 1만큼, 축소 연산은 2만큼 증가된다. w_2와 w_2는 A'으로부터 제거되고 w_5는 B'으로부터 제거된다.

$A' = <>$

$B' = <w_6, w_7>$

$|add| = 0, |drop| = 0, |copy| = 3, |shrink| = 4,$
$|replace| = 0, |move| = 1$

단계 4에서, A'이 비어있기 때문에 대체나 제거 대상 단어는 존재하지 않는다. w_6과 w_7은 추가된 단어로 고려되어, 추가 연산의 수는 2가 된다. 그러므로 최종적으로 최소 비용을 갖는 확장 편집 연산 시퀀스의 각 연산 수는

$|add| = 2, |drop| = 0, |copy| = 3, |shrink| = 4,$
$|replace| = 0, |move| = 1$

이 된다.

단계 5에서, A와 B 사이의 확장 편집 거리는 다음과 같이 계산된다.

$\delta_E = (2\cdot1) + (0\cdot1) + (3\cdot0.4) + (4\cdot0.4) + (1\cdot0) + (1\cdot0.9) = 5.7$

□

IED에서, 두 웹 문서 p와 p' 사이의 거리는 식 (6)과 같이 정의된다. m과 n은 각각 p와 p' 내에 존재하는 단어의 개수를 의미한다.

$$D_{IED}(p,p') = \frac{\delta_E}{\max(m,n)} \quad (6)$$

5. 실 험

웹 문서 변경도 측정 방법들의 비교 평가를 위하여 두 가지 실험이 수행되었다. 첫째, 실제 수집한 웹 문서 데이타에 대해 각 측정 방법이 웹 문서의 동일한 변경을 얼마나 다르게 표현하는지를 조사하였다. 둘째, 정의된 척도 하에 여섯 가지 변경 종류에 대한 각 측정 방법의 효과성을 평가하였다. 실험에서 사용된 측정 방법은 바이트 단위 비교(BW), TF·IDF 코사인 거리(COS), 단어 거리(WD), 편집 거리(ED), 10-쉥글링(10SH), 개선된 편집 거리 방법(IED)이다. IED 방법의 파라미터 s와 t는 각각 0.75로 설정하였다. 본 실험에서는 웹 문서의 내용 변화만을 문서 변경으로 간주하고, HTML 마크업 정보의 변경은 고려하지 않았다[4,5,8].

첫 번째 실험을 위하여 2005년 8월 한국 웹에서 41,469개의 웹 문서를 무작위로(random) 선택하였다. 각 문서를 이틀 간격으로 다운로드하여 두 개의 버전으로 저장하고, 제안된 여섯 가지 측정 방법으로 웹 문서의 두 버전 사이의 거리(즉 변경도)를 측정하였다.

웹 문서 집합에 속한 모든 웹 문서들의 변경도 합을 집합 내의 문서 개수로 나눈 값을 웹 문서 집합의 변경도 라고 하자. 예를 들어, 모든 웹 문서의 변경도가 1일 때 웹 문서 집합의 변경도는 1이 되고, 모든 웹 문서의 변경도가 0일 때 웹 문서 집합의 변경도 또한 0이 된다. 웹 문서 집합의 변경도는 문서 집합 내에 속한 웹 문서들이 평균적으로 변경된 정도를 표현한다고 할 수 있다.

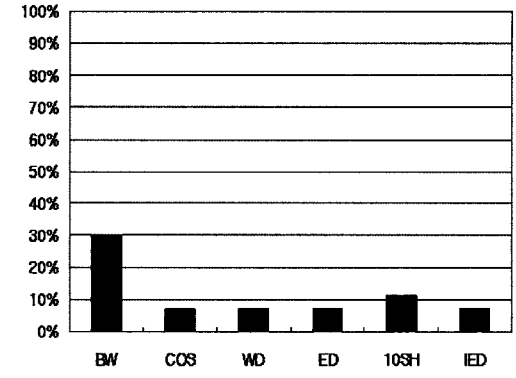
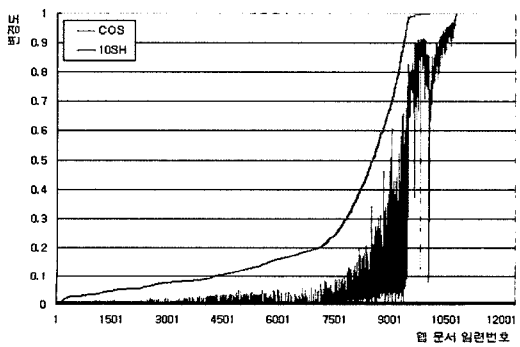


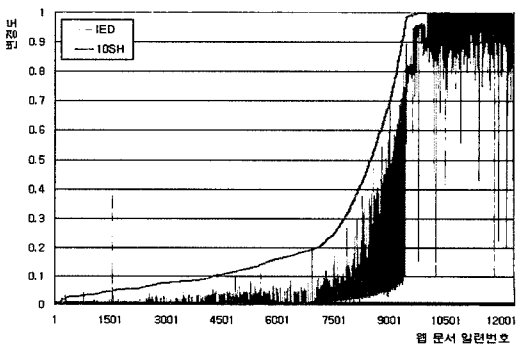
그림 5 웹 문서 집합의 변경도

그림 5는 여섯 가지 변경 측정 방법 사용에 따른 웹 문서 집합의 변경도를 나타낸다. BW 결과는 전체 웹 문서 중 약 30%가 변경되었다고 판단한다. BW는 매우 작은 변경에도 1을 반환하기 때문에 약 30%의 웹 문서는 완전히 변경되고 70%의 웹 문서는 전혀 변경되지 않은 것으로 볼 수 있다. COS, WD, ED, IED는 웹 문서 집합의 7%가 변경되었다고 판단하는 반면, 10SH은 약 12%가 변경되었다고 판단한다. BW가 웹 문서 변경에 가장 민감한 방법으로 나타나며, 10SH이 COS, WD, ED, IED에 비해 전반적으로 더 민감한 측정 방법이라는 것을 알 수 있다.

그림 6은 개별적인 웹 문서의 변경에 대한 변경도 표현 차이를 나타낸다. x축은 BW에 의해 변경된 것으로 판단된 웹 문서들(12,360개)의 일련번호를 의미하며, y축은 해당 번호의 웹 문서에 대한 변경도를 나타낸다. 각 측정 방법의 비교를 용이하게 하기 위하여, 그림 6(a), 6(b)에서의 웹 문서 일련번호는 각각 10SH 값에 대하여 오름차순으로 정렬되어 있다. 이 외의 다른 측정 방법들과의 비교결과는 위와 유사하게 나타났으므로 생략하였다. 그림에서 알 수 있듯이 각 측정 방법은 대부



(a) 10SH vs. COS



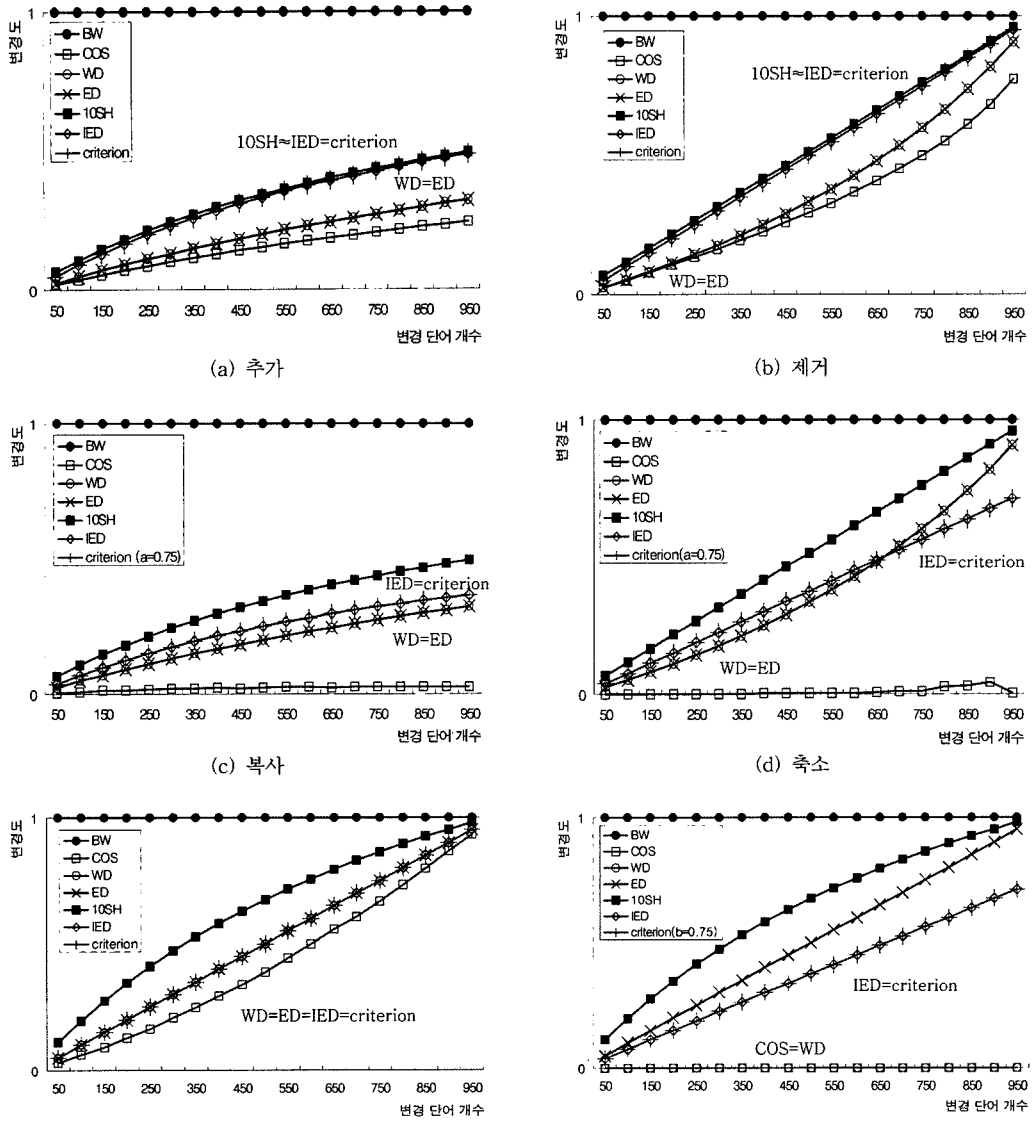
(b) 10SH vs. IED

그림 6 개별적인 웹 문서의 변경도

분의 웹 문서에 대해 서로 다른 변경도 값을 반환한다. 그림 6(a)에서 그 차이는 최대 약 0.92 정도로 매우 크다. 이는 웹 문서의 변경도가 사용되는 측정 방법에 의존적임을 내포하고 있다.

두 번째 실험에서는 여섯 가지의 문서 변경 종류의 관점에서 여섯 가지의 웹 문서 변경도 측정 방법의 효과성을 평가한다. 하나의 웹 문서 변경 종류에 대해, 척도와 유사한 값을 반환하는 변경도 측정 방법을 해당 변경 종류에 효과적이라 한다. 측정 방법의 결과가 척도보다 항상 높게 나타나면 민감도가 높은 방법이라 하며, 낮게 나타나면 민감도가 낮은 방법이라 한다. 평가 척도에서 복사와 축소의 가중치를 나타내는 파라미터 α와 이동의 가중치를 나타내는 파라미터 β는 모두 0.75로 설정하였다.

먼저 하나의 웹 문서에서 변경되는 단어 개수에 따라 각 측정 방법의 효과성을 비교한다. [4]에서는 전체 웹의 약 25%를 차지하는 문서들에 대략 1,000개의 단어가 포함되어 있음이 보고되었다. 본 실험에서는 1,000개의 단어로 구성된 문서의 변경되는 단어의 수를 5%에서 95%까지 변화 시키면서 각 측정 방법을 평가하였다. 또



(a) 추가 (b) 제거

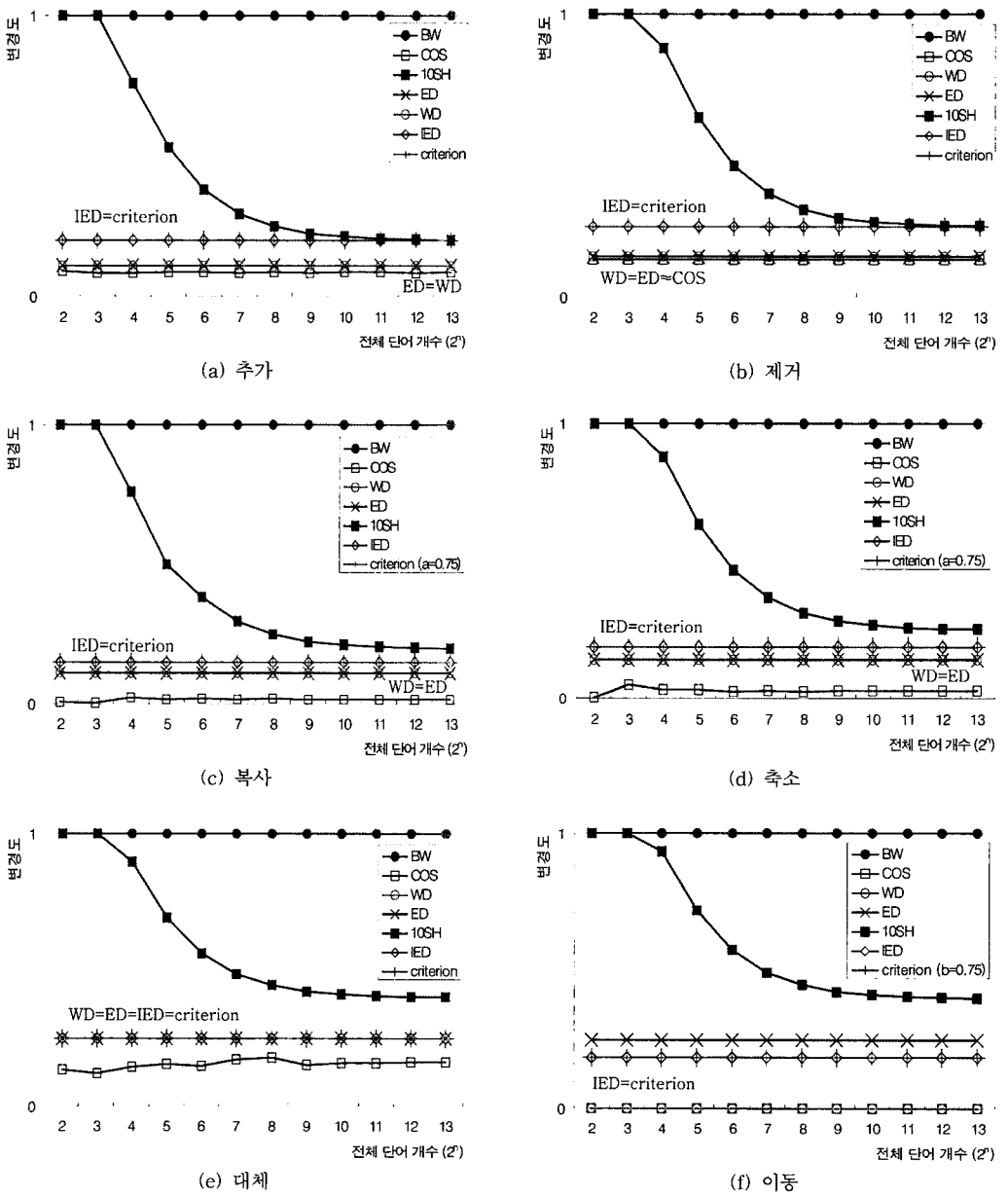
(c) 복사 (d) 축소

(e) 대체 (f) 이동

그림 7 변경되는 단어 수에 따른 효과성 평가

한 [8]에서는 웹 문서 내에 변경되는 단어들의 위치가 군집되어있음이 보고되었으며, 본 실험에서도 실험 문서 내의 변경되는 단어들의 위치가 군집되도록 하였다.

그림 7은 변경되는 단어 수에 따른 효과성 비교 결과를 나타낸다. x축은 1,000개 단어 중 변경된 단어의 개수를 의미하며, y축은 해당되는 문서의 변경도를 의미한다. BW는 변경 종류와 변경된 단어 개수에 상관없이 항상 변경도 1을 반환한다. 10SH은 추가, 제거 변경에 효과적이며, 복사, 축소, 대체, 이동 변경에는 민감한 결과를 보인다. 웹 문서의 변경도를 측정하는 환경에 따라 복사, 축소, 이동의 가중치 α,β는 다르게 간주될 수 있다. α,β값이 1에 가까워질수록 평가 척도가 반환하는 변경도와 10SH이 반환하는 변경도의 차이가 작아진다. α값이 1일 때 그림 7(c)와 그림 7(d)에서 척도의 그래프는 10SH의 그래프와 유사하게 (평균 약 0.01 차이) 나타났다. 이는 복사와 축소의 변경을 추가와 제거 변경



(a) 추가 (b) 제거

(c) 복사 (d) 축소

(e) 대체 (f) 이동

그림 8 문서 크기에 따른 효과성 평가

과 동일하게 간주하는 환경에서 10SH이 효과적인 웹 문서 변경도 측정 방법이 될 수 있음을 의미한다. β값이 1일 때 그림 7(f)에서 척도 그래프는 ED의 그래프와 유사하게 나타났다. 이는 대체와 이동의 변경을 동일하게 간주하는 환경일지라도, 10SH이 이동 변경에 여전히 민감함을 의미한다. COS는 여섯 가지 변경 종류 모두에 대해 둔감한 결과를 보인다. 특히 그림 7(c)와 그림 7(d)에서 변경된 단어 개수에 관계없이 매우 낮은 변경도(평균 약 0.01)를 보이며, 복사와 축소 변경을 거의 고려하지 못함을 나타내고 있다. 이동 변경에 대해서는 항상 0을 반환하여, 앞에서 기술한대로 단어순서의 변경을 전혀 고려하지 못한다는 것을 확인할 수 있다. WD는 대체 변경에 효과적이지만, 나머지 다섯 가지 변경 종류에는 낮은 민감도를 나타낸다. 그러나 α값이 0.5이면 복사와 축소 변경에 대해서도 효과적인 결과를 보인다. 즉, WD는 추가와 제거를 각각 복사와 축소의 약 2배 정도 큰 변경으로 간주한다. WD 역시 COS과 마찬가지로 이동 변경에서 항상 0을 반환하며 단어순서의 변경을 무시한다. ED는 이동을 대체와 동일한 변경으로 간주한다. 다시 말해, β가 1일 경우 이동 변경에 효과적이다. 이동을 제외한 변경 종류에서는 WD와 유사한 결과를 보인다. IED는 여섯 가지 변경 종류에 대해 모두 효과적이다. 이 사실은 본 논문에서 제안한 IED가 웹 문서의 여섯 가지 변경을 성공적으로 잘 반영하도록 고안되었음을 뒷받침해준다. IED는 척도의 가중치 파라미터 α와 β가 다른 값으로 설정된다 할지라도, IED의 입력 파라미터 s와 t를 동일하게 조절하여 반영할 수 있다. 예를 들어, 어떤 사용자가 α와 β를 각각 0.3, 0.75로 설정하였다고 가정해 보자. 위 실험 결과로부터 유추할 수 있듯이 10SH, COS, WD, ED 모두 복사, 축소, 이동 변경에서 효과적이지 못하게 된다. 하지만 IED는 s와 t 값의 입력을 각각 0.3, 0.75로 설정함으로써 여전히 효과적일 수 있다.

그림 8은 문서 크기에 따른 효과성을 평가한 결과이다. [4]에서는 2^2~2^{13}개 단어로 구성된 웹 문서가 실제 웹의 95%를 차지한다는 연구 결과가 보고되었으며, 본 실험에서는 단어 개수가 2^2, 2^3, 2^4, …, 2^{13}로 구성된 문서를 대상으로 각 크기 문서의 1/4의 단어가 변경되도록 하였다. 즉, 2^2개 단어 문서에서는 1개, 2^3개 단어 문서에서는 2개, 2^4개 단어 문서에서는 4개 단어가 변경되도록 설정하였다. 그림 8에서 x축은 변경 전 문서에 있는 전체 단어의 개수를 의미한다. 10SH은 여섯 가지 변경 종류 모두에 대해 문서의 크기가 작을수록 민감도가 매우 크게 나타난다. 작은 크기의 문서들로 구성된 웹 문서 집합에서 웹 문서 변경도 측정 방법으로서 쉥글링을 사용하는 것은 적절하지 못함을 의미한다. 다른 측정 방법들은 문서 크기에 관계없이 비교적 민감도가 일정하게 나타난다.

지금까지의 효과성 평가 실험에 대한 결과를 요약하면 표 1과 같다.

6. 결 론

본 논문에서는 웹 문서의 변경을 추가, 복사, 제거, 축소, 대체, 이동의 여섯 가지로 분류하여, 각 변경 종류에 대한 척도를 정의하였다. 또한 정의된 척도를 반영하도록 고안된 새로운 웹 문서 변경도 측정 방법을 제안하였다. 실험을 통해, 기존 연구에서 웹 문서의 변경도를 측정하기 위해 사용된 바이트 단위 비교, TF·IDF 코사인 거리, 단어 거리, 편집 거리, 쉥글링 방법과 본 논문에서 제안한 측정 방법의 효과성을 비교 평가하였다. 실험 결과로부터, 각 측정 방법이 웹 문서의 어떠한 변경 종류에 효과적인지 또는 해당 변경 종류를 얼마나 큰 변경으로 고려하는지 등의 특성을 알 수 있었다. 이러한 정보는 웹 문서의 변화를 조사할 때 어떤 측정 방법을 사용할 것인가에 대한 지침이 될 수 있다. 본 논문에서 제안한 측정 방법은 여섯 가지 모든 변경 종류에서 효과적인 결과를 보여, 척도를 성공적으로 반영하도록 고안되었음을 확인할 수 있었다.

표 1 평가 결과 요약

<table>
<tr><th>측정 방법 / 변경 종류</th><th>COS</th><th>WD</th><th>ED</th><th>10SH</th><th>IED</th></tr>
<tr><td>추가</td><td colspan="3" rowspan="2">낮은 민감도</td><td>효과적
(작은 크기의 문서 제외)</td><td rowspan="5">효과적</td></tr>
<tr><td>제거</td><td rowspan="2">$\alpha \approx 1$일 경우 효과적</td></tr>
<tr><td>복사</td><td rowspan="2">$\alpha \approx 0.01$일 경우
효과적</td><td colspan="2">$\alpha \approx 0.5$일 경우 효과적</td></tr>
<tr><td>축소</td><td colspan="2">불규칙한 민감도</td></tr>
<tr><td>대체</td><td>낮은 민감도</td><td colspan="2">효과적</td><td rowspan="2">높은 민감도</td></tr>
<tr><td>이동</td><td colspan="2">$\beta = 0$일 경우 효과적
(전혀 고려하지 못함)</td><td>$\beta \approx 1$일 경우 효과적</td></tr>
<tr><td>비고</td><td></td><td></td><td></td><td>작은 크기의 문서에 대해 매우 높은 민감도</td><td>복사, 축소, 이동 변경의 가중치 조절 가능</td></tr>
</table>

본 논문에서 정의한 척도는 웹 문서의 여섯 가지 변경 각각에 대하여 정의되었다. 그러나 실제 웹에서는 여섯 가지 변경이 복합적으로 발생하기 때문에, 여러 변경 종류가 혼합된 형태에 대한 척도 연구가 필요하다. 또한, 본 논문에서 제안한 측정 방법이 여섯 가지 변경에 가장 효과적인 결과를 보였지만 시간 복잡도에 관한 분석이 수행되지 않았다. 실제 거대한 수의 웹 문서를 대상으로 사용하기 위해, 제안된 측정 방법의 성능 개선 연구를 향후 계획으로 남긴다.

참 고 문 헌

[1] Google Search Engine, http://www.google.com

[2] Yahoo Search Engine, http://www.yahoo.com

[3] J. Cho and H. Garcia-Molina, "Synchronizing a Database to Improve Freshness," the 26th ACM SIGMOD International Conference on Management of Data, pp. 117-128, 2000.

[4] D. Fetterly, M. Manasse, M. Najork, and J. L. Wiener, "A Large-Scale Study of the Evolution of Web Pages," Software: Practice & Experience, Vol. 34, No. 2, pp. 213-237, 2004.

[5] A. Ntoulas, J. Cho, and C. Olston, "What's New on the Web? The Evolution of the Web from a Search Engine Perspective," In Proceedings of the 13th International World Wide Web Conference, pp. 1-12, 2004.

[6] B. E. Brewington and G. Cybenko, "How Dynamic is the Web?" In Proceedings of the 9th International World Wide Web Conference, pp. 257-276, 2000.

[7] S. J. Kim and S. H. Lee, "An Empirical Study on the Change of Web Pages," In Proceedings of the 7th Asia-Pacific Web Conference, pp. 632-642, 2005.

[8] L. Lim, M. Wang, S. Padmanabhan, J. S. Vitter, and R. Agarwal, "Characterizing Web Document Change," In Proceedings of the 2nd International Conference on Advances in Web-Age Information Management, pp. 133-144, 2001.

[9] G. Salton and M. J. McGill, "Introduction to Modern Information Retrieval," McGraw-Hill, 1983.

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithm," the Massachusetts Institute of Technology, 2001.

권 신 영

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

김 성 진

1998년 숭실대학교 소프트웨어 공학과(학사). 2000년 숭실대학교 대학원 컴퓨터학과(석사). 2004년 숭실대학교 대학원 컴퓨터학과(박사). 2004년~2006년 서울대학교 전기컴퓨터공학부, 박사후과정연구원. 2006년~현재 UCLA 컴퓨터학과 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 한국전자통신연구원, 선임연구원. 1999년~2000년 미국 조지 메이슨대 소프트웨어 정보 공학과 교환 교수. 1992년~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 인터넷 데이타베이스, 데이타베이스 시스템 성능평가 및 튜닝