정보기술연구소 논문집, 제2집, pp. 7~15, July 2000
Journal of The Institute of Information and Technology

데이타마이닝 기법을 이용한 효율적인 웹 검색엔진의 설계 및 구현

최민희, 오형재, 홍의경
서울시립대학교 전산통계학과

Design and Implementation of An Efficient Web Search Engine Using Data Mining Techniques

Min Hee Choi, Hyung Jae Oh, Eui Kyeong Hong
Dept. of Computer Science and Statistics, University of Seoul

Abstract

In this paper, an efficient web search engine has been designed and implemented by use of data mining techniques. A data model named as TLD(transformed log data) set has been developed by reforming the access-to-page log data occurred by the user-inputted queries. The results can be summarized as follows: firstly, in case of user access pattern analyses, non-accessed sites are exposed naturally to be inadequate, secondly, the dead-link is automatically informed to web master for the warning of dead-link deletion criteria required, and lastly, the searching time has been reduced considerably for the users of high access frequencies, with the aid of re-arranging the access data order in pages, and for the highly associated sites, with the aid of data quality reflection.

Key Words : 웹 검색엔진, 데이타마이닝, 연관규칙

1. 서론

데이타마이닝[1]은 대량의 실제 데이타로부터, 이전에 잘 알려지지는 않았지만 묵시적이고 잠재적으로 유용한 정보를 추출하는 작업으로서, 사용자가 한 사이트에 접근하면 대용량의 데이타에 숨겨진 유용한 패턴을 추출하는 방법론을 의미한다[9].

지금까지의 웹 검색엔진은 이러한 접근 패턴을 분석하지 않고, 단순히 질의에 따른 응답페이지의 신속한 생성에 초점을 맞추어 왔다. 그 결과 질의에 대한 응답 시 사이트들의 연관성과 사용자의 이력을 고려하지 않는 비효율적 결과 페이지까지 생성하게 되므로 사용자가 원하는 정보 검색을 어렵게 만들었다.

데이타마이닝 방법은 방대한 자료로부터 숨겨진 어떠한 정보 패턴을 추출하는데 관심이 있는 단계로 마이닝의 목적에 따라 여러 가지로 분류될 수 있다.

데이타마이닝 방법의 대표적인 예로는 분류(classification) 방법, 클러스터링(clustering) 방법, 연관규칙(association rule)과 일반화(generalization)를 포함하는 요약(summarization) 방법, 그리고 시계열(time-series), 예측(prediction) 모델링, 회귀분석(regression analysis)을 포함하는 변화(change) 방법들을 들 수 있다[11].

그러나 지금까지의 웹 로그에 대한 데이타마이닝 연구는 웹 서버의 트래픽을 모니터링하는 정도에 머무르고 있다[3]. 즉, 하나의 응답 페이지가 만들어 졌을 때,

그 페이지 안의 여러 링크들을 접근하는 패턴에 대한 연구는 연구 시작단계에 불과하다[4,8].

본 논문에서는 사용자가 입력한 질의와 응답 페이지를 접근할 때 로그 화일[2,6]에 수집되는 데이타들을 조합하여 그것을 데이타마이닝 기법중의 하나인 연관규칙 탐사 기법[3,4]으로 분석하여 접근 패턴을 발견하는 방법을 소개하고, 생성된 연관규칙을 해석하며 웹 검색엔진에 적용시킴으로써 직접 로그 데이타를 분석하여 효율적인 응답 페이지를 생성하는 시스템을 구현했다.

제 2장에서는 데이타 마이닝 기술에 대해서 기술하고, 제 3장에서는 본 연구의 동기가 된 웹 검색 엔진의 문제점을 살펴본다. 제 4장에서는 효율적인 웹 검색 엔진을 설계하고, 제 5장에서는 설계된 웹 검색엔진의 구현에 대해서 설명한다. 마지막으로 제 6장에서 결론과 향후 연구 방향에 대해서 언급하고 본 논문을 맺게 된다.

2. 연관규칙

2.1 연관규칙의 정의

연관규칙이란 어떤 사건이 일어나면 다른 사건이 일어나는 트랜잭션 내의 항목간의 연관성을 의미한다[3,5,10]. 주어진 트랜잭션의 집합을 {item-1, …, item-n}이라 하였을 때 서로 소의 관계인 두 개의 부분집합 A = {item-11, item-12, …, item-1m}과 B = {item-21, item-22, …, item-2k}의 연관규칙은 "A→B"로 표시된다. 이때 전제 항목들의 집합 A는 결론 항목들의 집합 B를 야기한다고 정의한다. Fig. 2.1에서 빵과 버터를 구매하는 사람은 우유를 구매한다는 연관규칙이 있음을 알 수 있다.

[빵], [버터] → [우유](support: 12.5%, confidence 90%)

Fig. 2.1. 연관규칙 예

2.2 연관규칙의 척도

1) 지지도(support degree) : 지지도란 생성된 연관규칙이 전체 아이템에서 차지하는 비율을 말한다. 즉 데이타베이스에 속한 전체 트랜잭션의 개수 중, 그 연관규칙을 지지하는 트랜잭션의 비율을 의미한다. 예를 들면, Fig. 2.1에서 전체 트랜잭션에 대해 빵과 버터를 함께 구매한 트랜잭션 수의 비율은 지지도 12.5%이다.

2) 신뢰도(confidence degree) : 신뢰도는 연관규칙의 강도를 의미하며 전제부를 만족하는 트랜잭션이 결론부까지를 만족하는 비율을 말한다. 예를 들면, Fig. 2.1에서 빵과 버터를 구매한 고객들 중에 우유를 함께 구매한 트랜잭션의 신뢰도는 90%이다. X(전제부)를 규칙의 가정, Y(결과부)를 규칙의 결과라 할 때, 가정과 결과로 이루어지는 연관규칙을 갖는 항목들의 집합을 항목집합(itemset)이라 하고, 항목집합에 있는 항목들의 수를 항목집합의 길이라 하며, K길이를 가지는 항목집합은 k-항목집합이라 하는데, N개의 트랜잭션 T로 구성된 집합을 DB라 표기하면 연관규칙은 다음과 같은 규칙을 만족한다.

2.3 연관규칙 탐사

<기본 가정>

1) 항목집합 I의 부분 집합 X에 대해, $X \subseteq T$ 이면 T는 X를 만족한다.
2) $X, Y \subseteq I$에 대한 연관규칙 $X \rightarrow Y$는 $X \cap Y = \emptyset$의 특성을 갖는다.

<용어>

- 지지도: 연관규칙 $X \rightarrow Y$는 지지도 S를 가지며, S는 다음과 같이 표시된다.

  $$S = |X \cup Y| / |N|$$

  단, 위의 식에서 |N|은 항목집합 N($N \subseteq I$)를 만족시키는 DB의 트랜잭션 수를 의미한다.
- 신뢰도: 연관규칙 $X \rightarrow Y$는 신뢰도 C를 가진다.

  $$C = |X \cup Y| / |X|$$
- 항목집합: 가정(X)와 결과(Y)로 이루어지는 연관규칙을 갖는 항목들의 집합을 의미한다.
- 빈발 항목집합: 외부 입력 값인 최소 지지도(minconf) 이상의 지지도를 갖는 $X \subseteq I$ 를 의미한다.

<기호 설명 >

- C_k: 후보 k-항목집합(항목집합과 지지도 항목을 가짐).
- L_k: 빈발 k-항목집합

이상에서 기술한 내용을 토대로 기본적인 규칙 탐사 알고리즘의 구조를 묘사하기로 한다. 지금까지 연구된 알고리즘들은 서로 다름에도 불구하고 그들 모두는 기본적인 스키마를 사용한다. 이들 요소는 알고리즘에 따라 서로 다르게 배열될 수 있고 전 스키마가 반복적으로 적용될 수도 s있다. 주어진 데이타베이스에서 탐사되는 연관규칙은 사용자가 정의한 최소 지지도와 최소 신뢰도 이상의 값들을 가져야 하므로, 연관규칙을 탐사하는 문제는 기본적으로 다음의 두 단계로 구성된다.

단계 1 : 빈발 항목집합들을 찾아낸다.

단계 2 : 데이타베이스로부터 연관규칙을 생성하기 위하여 빈발 항목집합들을 사용하고, 모든 빈발 항목집합 L에 대해서 L의 모든 공집합이 아닌 부분집합들을 찾는다. 그러한 각 부분집합 A에 대하여, 만약 sup(A)(트랜잭션 T에서 항목 X의 지지도를 생략된 형태를 sup(X)로 표기함)에 대한 sup(L)의 비율이 적어도 최소 신뢰도(minconf) 이상이면 (sup(L) / sup(A) ≥ minconf), A → (L-A) 형태의 규칙을 출력한다.

단계 1을 좀 더 자세히 기술하면 다음과 같다. 빈발 항목집합들을 찾아내는 과정에서, 잠재적인 빈발 항목집합들의 수는 고려될 항목들의 크기에 대하여 기하급수적으로 증가한다. 이것을 찾아내는 대부분의 알고리즘이 고려하는 기본적인 방법은 후보(candidates)라 지칭하는 빈발 가능성이 있는 항목집합들의 생성을 포함한다. 이들 후보 항목집합들 중에 실제로 빈발한(large) 항목들을 찾기 위해서는 각 후보 항목집합들에 대한 지지도가 데이타베이스를 읽어가면서 계산되어야 한다.

연관규칙 탐사의 전체 성능은 단계 1에서 결정된다. 먼저 빈발 항목집합들을 확인한 후에 해당되는 연관규칙을 단계 2의 방법으로 쉽게 유도할 수 있다. 빈발 항목집합들을 찾아내는 과정을 이해하기 위하여 간단한 데이타베이스에서 후보 항목집합의 생성과 그것에서 빈발 항목집합을 찾아내는 방법을 Fig. 2.2, Fig. 2.3을 통하여 설명하기로 한다.

지금까지 발표된 알고리즘 중에서 전형적인 방법론인 Apriori[1]가 적용된 방법론을 설명한다. Apriori에서는 각 패스에서 빈발 항목집합들의 후보 항목집합들을 구성한 후에 각 후보 항목집합의 발생 빈도수를 계산하고, 사용자가 정의한 최소 지지도를 기초로 하여 빈발 항목집합들을 결정한다. Fig. 2.2는 설명을 위한 데이타베이스이고, Fig. 2.3은 빈발 항목집합을 찾는 과정을 설명한다.

Database D

TID	Items
100	A C D
200	B C D E
300	A B C E
400	B E

Fig. 2.2. 트랜잭션 데이타베이스의 예

첫번째 단계에서 각 항목의 발생 빈도수를 세기 위하여 단순히 모든 트랜잭션들을 스캔하여 읽는다. 후보 1-항목집합들의 집합 C1은 Fig. 2.3에서와 같이 얻어진다. 최소 트랜잭션 지지도가 2라고 가정하면(minsup = 2), 필요로 하는 최소 지지도를 갖는 후보 1-항목집합으로 구성되는 빈발 1-항목집합들의 집합 L1이 결정될 수 있다.

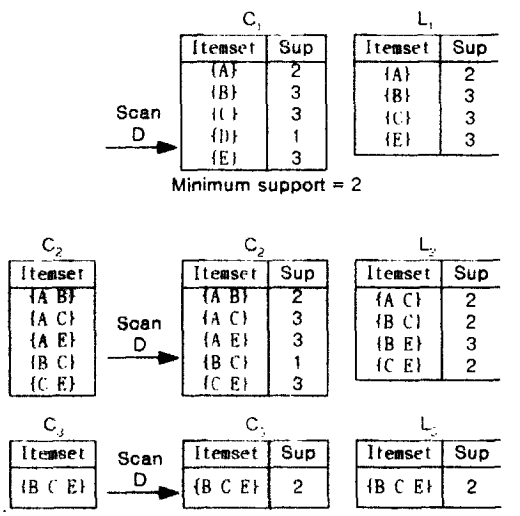

Fig. 2.3. 후보 항목집합의 생성과 빈발 항목집합

빈발 2-항목집합들의 집합을 탐사하기 위해서는, 모든 부분집합도 역시 최소 지지도를 가져야 한다는 사실에 입각하여 Apriori는 후보 항목집합들의 집합 C2를 생성하기 위해 L1 * L1을 사용하였다. 여기서 *는 집합 연산자다. C2는 (|L1| / 2) 개의 2-항목집합들로 이루어진다. L1이 크게 되면 (|L1| / 2)는 매우 큰 숫자가 됨을 알 수 있다.

다음으로 데이타베이스 D에 속한 네 개의 트랜잭션들이 스캔되어 읽히고 C2에 속한 각 후보 항목집합들의 지지도가 계산된다. Fig. 2.3에서 두 번째 행의 가운데 테이블은 C2에 속한 후보 항목집합들의 지지도 계산 결과를 나타낸다. 빈발 2-항목집합들의 집합 L2는 C2에 속한 각 후보 2-항목집합들의 지지도에 기초하여 결정된다.

후보 항목집합들의 집합 C3는 L2에서 다음과 같이 생성된다. L2에서 첫 항목이 같은 두 개의 빈발 2-항목집합들을 먼저 확인한다. 예를 들면, {BC}와 {BE}에서는 B가 동일한 항목이다. 다음으로, Apriori는 {BC}와 {BE}의 두 번째 항목들로 구성된 2-항목집합 {CE}가 L2의 원소로 빈발 집합이므로, {BCE}의 모든 부분집합들은 빈발하다는 것을 알았고, 그러므로 {BCE}는 후보 3-항목집합이 된다. L2에서 더 이상의 다른 후보 3-항목집합들을 구할 수 없다. 그러면, Apriori는 모든 트랜잭션들을 스캔하면서 Fig. 2.3에서와 같이 빈발 3-항목집합들을 구성한다. C3를 기본으로 하여 D를 스캔하여 L3를 찾아낸다. L3에서부터 구성될 수 있는 후보 4-항목집합들이 없으므로, 여기서 빈발 항목집합을 발견하는 과정을 마친다.

3. 웹 검색엔진의 문제점

여러 검색엔진 사이트에서 서비스되는 웹 검색엔진은 다량의 데이타에서 문자열이 포함되어 있는 데이타의 신속한 검색결과 도출에 중점을 두어 개발되어 왔다[4]. 현재의 기술은 각종 인덱싱 기법 등의 기술 발달과 서버측 하드웨어의 발전으로 말미암아 엔진의 속도면에서는 사용자들은 거의 불편을 느끼지 않는다.

웹 검색엔진의 문제점을 데이타베이스 관리와 로그 데이타 활용의 두 가지 측면에서 살펴보면 다음과 같다.

- DB 관리상의 문제

웹 검색엔진의 구축은 로봇을 이용한 방식과 사용자들이 직접 등록하는 방법이 있다. 이중에서 사용자들이 등록하는 방법을 살펴보면, 홈페이지 관리자(사용자)가 검색엔진에 홈페이지를 등록하고 변경되거나 홈페이지가 없어지면 내용을 변경하거나 삭제하도록 되어있다. 그러나 등록은 신중히 하지만 관리자(사용자)가 스스로 홈페이지의 변경 사항을 웹 검색엔진에서 변경(삭제)하는 일은 거의 없기 때문에, 홈페이지의 내용이 변경되거나 없어진 사이트도 여전히 나타난다. 그 결과 웹 검색엔진 사용자들은 변경되기 이전의 사이트 혹은 존재하지 않는 페이지를 방문하는 경우가 발생한다.

- 로그 데이타 활용상의 문제

1) 사용자의 피드백이 없는 단방향 서비스

사용자는 검색엔진에 질의를 요청하고, 그에 대한 결과로 나온 사이트를 보는 것에 그친다. 앞부분에서 언급한 바와 같이 이러한 과정에서 대량으로 발생하는 사용자들의 접근 패턴을 나타내는 로그 데이타가 생성된다. 로그 데이타를 분석하면 다음 질의에 대한 효율적인 응답 페이지를 생성할 수 있다.

2) 질의에 치우친 서비스

웹 검색엔진의 서비스는 검색을 요청한 질의에 중심을 둔 응답 페이지를 생성한다. 즉 이전에 질의를 누가(사용자), 어디서(IP 주소), 어떤 질의를 요청하였는 지에 대한 이력을 고려하지 않는다.

4. 효율적인 웹 검색엔진의 설계

4.1. 전체 시스템 구조

시스템은 네 개의 주요 모듈로 구성되어 있다. 사용자 세션 정보를 관리하는 웹 서버, 변형 로그 데이타 분석을 담당하는 오프라인 모듈과 동적인 페이지를 생성해 제공하는 온라인 모듈로 되어 있다. 각각의 모듈을 설명하면 다음과 같다.

- 웹 서버는 NCSA httpd 서버와 같이 HTTP 프로토콜을 지원한다. 차이점은 사용자와 웹 서버간의 지속적인 상호작용을 지원하도록 사용자 세션을 관리한다.
- 오프라인 모듈에서 전처리기는 주기적으로 사용자 접근 로그로부터 사용자 세션의 레코드 생성을 위

해 정보를 추출한다. 레코드에는 각 세션에서의 사용자 접근패턴이 등록된다. 레코드는 파싱과 필터링 과정을 거쳐 규칙 생성 관리자가 필요로 하는 데이타 형태로 가공하여 넘겨준다.

- 연관규칙 탐색기는 각각의 규칙 생성에 맞게 입력된 정보를 처리하여 연관규칙을 생성한다.
- 마지막으로 온라인 모듈인 동적 페이지 생성기는 웹 서버로부터 넘겨받은 데이타를 이용하여 검색엔진 데이타베이스에 저장된 정보를 검색해 문서를 만든다.

이때 사용자의 접근 패턴에 관한 정보를 연관규칙 탐색기를 통해서 읽어들여 동적으로 문서를 생성하여 서비스한다. 생성된 응답 페이지는 기존의 정적인 문서가 아니라 각 사용자마다, 그리고 연관된 질의에 따라 변하는 동적인 문서이다.

4.2 웹 검색엔진 사용자의 접근 패턴

웹 검색엔진 사용자는 원하는 정보를 찾기 위해 후보 사이트들 중에서 원하는 정보가 있는 곳을 찾을 때까지 여러 사이트에 접근한다. 이 과정에서 원하는 정보가 없는 사이트를 방문했을 경우에는 다시 검색엔진으로 돌아온다. 이때 데이타마이닝 수행시 외부 입력으로 받아들여 질의에의 사이트 부적합 여부를 판단한다. 찾는 정보가 있는 사이트 방문시는 어느 정도 오랜 시간 동안 그 사이트에 접속하게 되고, 다시는 웹 검색엔진으로 돌아오지 않아도 된다. 이 경우는 사용자가 원하는 사이트에 접근하였다고 판단할 수 있다.

4.3 데이타 필터링 및 모델링

웹 검색엔진에 접속하는 사용자들은 서버의 로그 화일에 접속 데이타를 남긴다. 로그 화일은 CERN과 NCSA에서 HTTP프로토콜로 규정한 표준 로그형식(Common Log Format)을 따른다. 이 표준을 따르는 로그 항목의 구조를 살펴보면, {<사용자의 IP 주소>, <사용자 ID>, <접근 권한>, <접근 시간>, <요구 방법>, <접근한 페이지 URL>, <데이타 전송을 위한 프로토콜>, <에러 코드 상태>, <전송된 데이타의 크기>}로 구성된다[2,4].

본 논문에서는 로그 데이타의 항목 중 일부분과 사용자가 제공한 질의를 덧붙여 생성한 변형된 로그 데이타로 연관규칙 탐사를 위한 데이타 모델을 구축했다. 또한 본 모델 구축은 사용자들이 특정 질의에 따라 생성된 결과 페이지를 어떠한 패턴을 가지고 접근하는가를 분석하여 보다 효율적인 결과 페이지를 생성하도록 하기 위한 것이다. 서버 로그에는 연관규칙 탐사와 관련이 없는 항목들도 포함되어 있다. 이와 같이 불필요한 로그 항목들을 제거하는 과정을 데이타 필터링이라 한다. 관심있는 항목들은 <사용자의 IP 주소>, <사용자 ID>, <접근 시간>, <접근한 페이지의 URL>이다. 로그 항목들이 필터링되면 로그 데이타는 연관규칙 탐사를 위한 데이타 모델에 알맞은 형태로 변환된다.

관심있는 로그 항목들에 사용자가 입력한 질의를 덧붙여서 연관규칙 탐사를 위한 데이타 모델로 하고, 이것을 **변형 로그 데이타**라 명명한다.

하나의 트랜잭션이 자연스럽게 정의되는 장바구니 분석과는 달리 웹 로그 데이타를 가지고서는 트랜잭션을 정의하기가 자연스럽지 않다. 본 논문에서는 사용자가 로그 화일에 남긴 <접근 시간> 항목에 기초하여 트랜잭션을 정의하였다. 즉 주어진 최대 시간 간격안에 같은 사용자(<사용자의 IP 주소>, <사용자 ID>, <질의>)가 기록한 로그 항목 중에서 URL을 사용하여 트랜잭션을 구성하였다.

연관규칙 탐사를 위한 데이타 모델인 변형 로그 데이타를 구성하는 항목들에 대해 살펴보면 아래와 같다.

- 변형 로그 데이타 집합을 L이라 하고 하나의 변형 로그 데이타 $l \in L$이 있을 때, l은 다음의 항목으로 이루어진다.
  - *l.ip* : 사용자의 IP 주소
  - *l.uid* : 사용자의 ID
  - *l.url* : 사용자가 접근한 페이지의 URL
  - *l.query* : 질의 집합

연관규칙 탐사를 위한 데이타 모델에 대해 간단하게 정의한다.

정의 1: 하나의 연관 트랜잭션 T는 다음과 같이 4개의 항목으로 구성된다.

트랜잭션 T : $<ip_t, uid_t, query_t, \{l_{t1}.url, \cdots, l_{tm}.url\}>$

단, $1 \le k < j \le m$, $l_j.time - l_k.time \le t_{max}$(최대 시간 간격), $l_{tk}.url \in url_{query}$, $l_k.ip = ip_t$,

$l_k.uid = uid_t$

정의 2: 연관규칙은 $X \xrightarrow{s,\ \alpha} Y$ 형태로 표현한다. 이때 $X \subseteq WS$, $Y \subseteq WS$이다.

단, s는 지지도로서 $\sigma(X \cup Y) / |T|$를 의미하고, α는 신뢰도로서 $\sigma(X \cup Y) / \sigma(X)$를 의미한다.

정의 3: $\{(l_1.url, l_1.time), (l_m.url, l_m.time)\}$을 URL_t라 할 때 연관 트랜잭션 집합 T를 $<ip_t, uid_t, query_t, URL_t>$로 표시한다.

정의 2에서 연관규칙 발견의 문제는 최소 임계치 s, α가 주어졌을 때, $X \xrightarrow{s,\ \alpha} Y$의 모든 규칙을 찾아내는 일이다. 이때, $query_t$에 따른 $\{URL_{query}$: 검색된 모든 URL$\}$이 존재하며, 그것을 WS(Web Space)로 표기한다.

4.4 전처리를 통한 변형 로그 데이타 생성

본 논문에서는 표준 로그형식과 사용자가 입력한 질의를 조합하여 연관규칙 탐색기의 입력 데이타로 사용한다. 전처리 단계에서 수행하는 일은 사용자 입력 질의로 생성된 페이지를 통해 얻어지는 로그에서 필요한 항목을 추출, 그것과 입력 질의를 합하여 테이블 형태로 저장하고 데이타마이닝 작업의 입력으로 사용할 수 있도록 데이타를 변형하는 일이다.

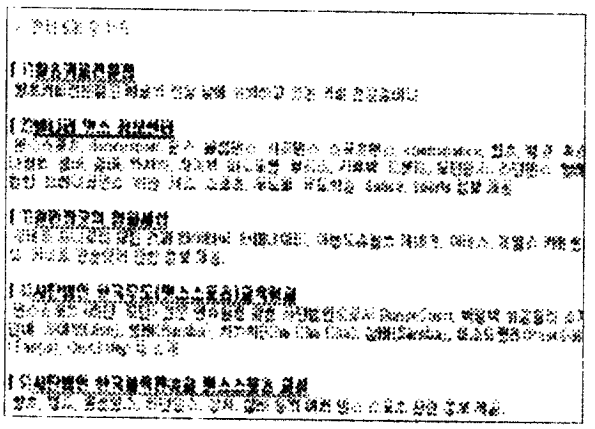

Fig. 4.1. 질의 "왈츠"로 검색한 결과 페이지

Fig. 4.1과 같은 웹 검색 결과 화면이 있고, 사용자는 결과 화면에서 "2.별나라 댄스 정보센터, 4.사단법인 한국무도(댄스스포츠)교육협회, 5.사단법인 한국체육진흥회 댄스 스포츠 교실"의 세 링크를 탐색했다고 할 때, 생성되는 로그 데이타는 Fig. 4.2와 같다. 로그 데이타의 여러 항목들 중에서 요구방법(post, get), 전송 화일 크기 등의 항목은 전처리 과정에서 제거된다. Fig. 4.2는 로그에서 필요한 항목인 사용자 IP 주소, 사용자 ID, URL, 접근 시간 등을 추출하여 질의("왈츠")와 덧붙여서 트랜잭션 T를 생성함을 보여주고 있다. 생성된 트랜잭션들을 테이블 형태로 구성한 것이 Fig. 4.2이다.

```
winger.uos.ac.kr mhchoi93 - [19/Oct/1999:13:40:32 -0800]\
GET http://www.dancesports.org/ HTTP/1.0  200 1502
winger.uos.ac.kr mhchoi93 - [19/Oct/1999:13:45:20 -0800]\
GET http://www.dancesport-korea.com/ HTTP/1.0  200 662
winger.uos.ac.kr mhchoi93 - [19/Oct/1999:13:40:32 -0800]\
GET http://dancekorea.pe.kr/ HTTP/1.0  200 2980
```

Fig. 4.2. 로그 데이타

5. 구 현

5.1 연관규칙 생성

앞에서 생성한 로그 데이타는 연관규칙 탐사를 통해 유용한 정보를 생성한다. 생성된 정보를 바탕으로 효율적인 웹 검색엔진을 구성한다. 변형 로그 데이타로 구성된 트랜잭션 T가 주어지면, 연관규칙 탐사를 통해서 사용자가 지정한 최소 지지도(minsup)와 최소 신뢰도(minconf)보다 큰 지지도와 신뢰도를 갖는 모든 연관규칙을 생성할 수 있다. 여기서 연관규칙을 찾는 알고리즘은 Apriori 방법을 토대로 하였다.

Apriori 방법을 통한 연관규칙 탐사는 두 단계로 시행된다.

첫 단계에서는 빈발 항목집합들을 찾는다. 이때 미리 결정된 최소 지지도 minsup 이상의 트랜잭션 지지도를 가지는 항목집합들의 모든 집합들을 빈발 항목집합들이라 하고, 그 외 모든 항목집합들은 작은 항목집합들이라고 부른다.

다음 단계에서는 데이타베이스로부터 연관 규칙을 생성하기 위하여 빈발 항목집합을 사용한다. 즉 모든

빈발 항목집합 L에 대해서 L의 모든 공집합이 아닌 부분집합들을 찾는다. 그 후 각각의 부분집합 A에 대해 만약 supp(A)에 대한 supp(L)의 비율이 적어도 최소 신뢰도 minconf 이상이면(supp(L)/supp(A) ≥ minconf), A → (L-A)의 형태의 규칙을 출력한다. 이때 생성된 지지도는 supp(L)이고 신뢰도는 supp(L)/supp(A)이다.

마지막 단계에서는 생성된 후보 항목집합(C_k)이 실질적으로 빈발한지 지지도를 조사하여, 빈발하다는 결정이 내려지면 후보 항목집합(C_k)는 다음 단계의 seed(L_k)가 된다.

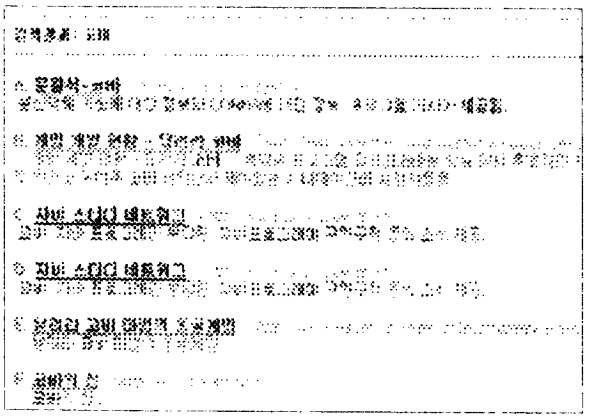

Fig. 5.1. 기존 검색엔진의 응답 페이지

Fig. 5.1은 본 논문의 실험 대상이다. 기존의 웹 검색엔진 서버에서 제공하는 응답 페이지를 보여주고 있다. 사용자는 A~F 사이트를 접근하는 동안 서버에 로그 데이타를 생성한다.

Table 5.1은 단위 시간 동안 각 사용자에 의해 접근된 페이지 항목을 보여준다. 각 항목에 대하여 1은 접근하였음을 의미하며, 0은 접근이 없었음을 의미한다.

Table 5.1. 연관규칙 탐사모델 예

트랜잭션 번호	사용자 IP	사용자 ID	실의	A	B	C	D	E	F
1001	203.249.110.61	winger	코바	1	0	1	0	0	0
1103	203.238.126.09	lion	코바	0	1	0	0	1	1
1406	203.249.110.83	shkim	코바	0	0	1	0	0	0
1674	203.249.110.34	database	코바	1	0	1	0	0	0
2081	168.134.6.54	yplee	코바	0	1	0	0	1	0
2411	137.68.1.94	jsko	코바	1	0	1	0	0	0
2590	201.44.136.65	dmseo	코바	0	1	0	0	1	1
2815	210.107.14.66	vision	코바	0	0	1	0	0	0

사용자 1001은 한번의 트랜잭션 안에서 A, C 항목의 페이지에 접근하였다는 사실을 알 수 있다. 최소 지지도를 만족시키는 횟수는 8개의 항목에 대해서 20% 이상이므로 1.6번 이상, 즉 2번 이상 접근된 항목은 최소 지지도 minsup를 만족시킨다고 말할 수 있다. 이때 빈발 항목집합을 구하기 위해 항목별로 지지도를 계산하여 최소 지지도 이하의 항목은 삭제한다.

Table 5.1은 연관규칙 탐사 예이다. 즉 트랜잭션을 분석하여 사용자가 접근한 페이지 항목별로 최소 지지도 minsup 20%를 만족하는 빈발 항목집합을 Table 5.2에 나타내고 있다.

Table 5.2. 최소지지도(minsup) : 20%를 기준으로 생성한 빈발 항목집합

Large Itemset	트랜잭션 수	지지도(%)
C	5	62.5
A	3	37.5
B	3	37.5
E	3	37.5
C,A	2	25
B,E	3	37.5
B,F	2	25
E,F	2	25
B,E,F	2	25

Table 5.3은 주요 항목집합으로부터 사용자가 지정한 최소 신뢰도 minconf 50%를 만족하는 연관규칙 생성을 보이고 있다. 분석된 데이타로부터 흥미 있는 현상을 발견할 수 있다.

Table 5.3. 최소 신뢰도(minconf) : 50%를 기준으로 생성한 연관 규칙

Rule No.	신뢰도(%)	지지도(%)	연관규칙
1	25	60	C→A
2	37.5	100	B→E
3	25	66	B→F
4	25	66	E→F
5	25	66	{B,E}→F

5.2 연관규칙의 적용

생성된 연관규칙을 적용하는 한 예는 연관규칙을 검색엔진 데이타베이스에 반영하는 것이다.

앞의 연관규칙 탐사 과정에서 D 사이트는 한 차례도

접근이 이루어지지 않았다. 실제로 접근이 없었던 이유는 Fig. 5.1에서 보는 바와 같이 C와 D가 중복되어 데이타베이스에 저장되어 있기 때문이다. 이처럼 무접근 사이트가 발견된 경우 그 사이트는 중복된 사이트이거나, 응답 페이지의 내용이 질의와 무관한 내용의 사이트이다. 이와 같은 결과를 해석하여 D 사이트의 정보를 웹 검색엔진 데이타베이스에서 삭제한다.

또한 F 사이트의 경우 접근 유지시간이 주어진 시간보다 짧았던 것을 발견하였다. 이것은 F 사이트가 실제로 존재하지 않거나 내용이 변경된 사이트임을 의미한다. 여기서 존재하지 않는 경우는 네트워크 오류 때문에도 발생하므로 이러한 경우에는 바로 삭제하지 않고 웹 서버 관리자에게 실제로 F 사이트가 데드링크인지 여부를 확인하도록 통보함으로써 데드링크의 삭제 기준을 제시하였다.

앞의 Table 5.3에서 발견된 연관규칙에 따라서 C, A 그리고 B, E가 각각 서로 상호 연관이 있음을 발견하였다. 여기서 F 사이트는 데드링크이므로 연관규칙을 해석하는 과정에서 데이타베이스에서 삭제하였다. 이 결과를 순서화, 시각적 주목성 가미 방법으로 웹 검색엔진의 응답 페이지에 적용할 수 있다. 각각에 대해서 살펴보면 다음과 같다

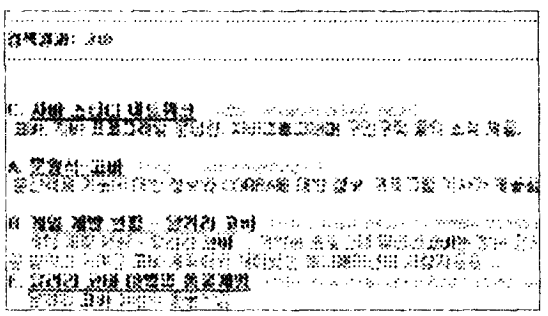

Fig. 5.2. 이력을 고려하지 않은 효율적인 응답 페이지

생성된 연관규칙 C → A, B → E를 바탕으로 사용자들의 검색 시간을 단축할 수 있도록 응답 페이지를 구성할 수 있다. 이때 응답 페이지의 구성은 사용자에 따라 동적으로 이루어진다. 즉, 접근 이력이 없는 사용자의 경우에는 C, A와 B, E를 순서에 상관없이 두 개씩 그룹화하여 응답 페이지에 보임으로써 각 사이트가 연관된 주제를 가지고 있음을 알린다. Fig. 5.2는 접근 이력이 없는 사용자의 응답 페이지의 예이다.

이전에 접근 이력이 있는 사용자의 경우에는 그 이력을 고려하여 응답 페이지를 생성할 수 있는데, 다음의 두 가지 기법을 사용한다.

- 순서화: 다음 응답시 같은 페이지 내에 있으며 서로 순서적으로 관련성있게 나타냄.
- 시각 주목성 가미: 연관된 사이트들을 폰트 크기, 혹은 폰트 굵기 등의 방법을 통하여 시각적 주목성을 높여 연관성이 있음을 표시.

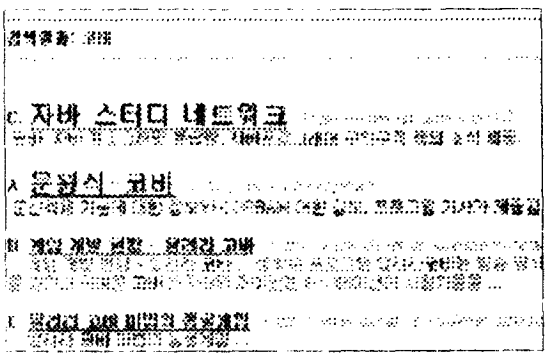

Fig. 5.3. 사용자의 이력을 고려한 동적인 응답 페이지

예를 들어, <사용자 IP>, <사용자 ID>가 각각 168.134.6.54, yplee인 사용자가 "코바"라는 질의로 검색을 요청했을 경우에는 이력이 있으므로(Table 5.1 트랜잭션 번호 2081), 이전에 방문했던 B, F 사이트를 적절히 순서화하여 페이지에 배치하고, 링크의 다른 색깔과 폰트 굵기를 조절하여 해당 사이트를 강조하는 시각적 주목성을 가미하여 사용자의 검색 시간을 단축시킬 수 있다. Fig. 5.3은 이 경우의 응답 페이지 화면이다.

Fig. 5.3에서 보는 것과 같이 이전의 이력이 있는 사이트 C와 A의 경우 글꼴의 색과 굵기를 바꾸고 순서를 앞쪽으로 그룹화하여 보여주고 있다.

6. 결론

본 논문에서는 데이타마이닝 기법을 이용하여 효율적인 웹 검색엔진을 설계 및 구현했다. 웹 검색엔진 서버에서 사용자가 입력한 질의와 그에 따라 생성된 페이지로의 접근 로그 데이타를 재구성하여 데이타 모델인 변형 로그 데이타 집합을 개발하였다. 이를 대상으로 효율적인 연관규칙 탐사 수행 과정을 보였으며, 이

때 생성된 연관규칙을 웹 검색엔진 서버 구성 및 운영에 적용하였다.

적용 내용을 세분화하면 다음과 같다.

1) 사용자의 접근 패턴 분석시 무접근 사이트의 경우에는 그 내용의 부적절성을 찾아낼 수 있도록 하였고, 그 경우 해당 항목은 데이타베이스에서 제거되어 다음 응답 페이지 작성시에는 포함되지 않는다.

2) 해당 사이트에 접근자의 짧은 접속 유지시간 사이트를 데드링크로 분류하여, 이를 웹 검색엔진 관리자에게 통보함으로써 기존의 서버에 데드링크 삭제 기준을 제시하였다.

3) 사용자의 접근 빈도가 기준치보다 큰 사이트를 발견하면 해당 사이트의 페이지 내의 위치를 적절히 조정함으로써 사용자의 정보 검색 시간을 단축시켰다.

4) 특히 사용자와 사이트간의 연관성을 발견하여 높은 연관성의 사이트들을 유형별로 그룹화하고, 각 그룹에 시각적 주목성을 가미함으로써 사용자들의 검색 시간을 단축시켰다.

향후 연구 방향은 복합 질의와 관련된 데이타 모델을 생성하여 그것에 대해 데이타마이닝 기법을 적용하는 방법에 대한 연구이다. 페이지의 접근시간과 관련된 데이타마이닝 기법인 순차 패턴분석을 통하여 사용자 접근 패턴을 분석하여 검색엔진을 구성하는 방법에 대한 연구와, 현재 서비스되고 있는 실제의 웹 검색엔진 서버에 본 시스템을 구축하여 실제의 효용에 대한 연구를 후속 과제로 남긴다.

참고문헌

[1] R. Agrawal and R. Srikant, "Fast Algorithm for Mining Association Rules," Proc. of 20th Int'l Conf. on VLDB, pp.487-499, 1994.

[2] M. C. Drott, "Using Web Server Logs to Improve Site Design," Proc. of 16th Annual Int'l Conf. on Computer Documentation, pp.43-50, 1998.

[3] U. Fayyad and R. Uthurusamy, eds., "Data Mining," Special Issue, Comm. of the ACM, 39(11), Nov. 1996.

[4] M. N. Garofalakis, R. Rastogi, S. Seshadri and K. Shim, "Data mining and the Web: Past, Present and Future," Proc. of 2nd Int'l Workshop on Web Information and Data Management, pp.43-47, 1999.

[5] J. Han, "Data Mining Techniques," Proc. of 1996 ACM SIGMOD Int'l Conf. on Management of Data, 1996.

[6] K. P. Joshi, A. Joshi, Y. Yesha and R. Krishnapuram, "Warehousing and Mining Web Logs," Proc. of 2nd Int'l Workshop on Web Information and Data Management, pp.63-68, 1999.

[7] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," Proc. of 21th Int'l Conf. on VLDB, pp.407-419, 1995.

[8] O.R. Zaiane, M. Xin, J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs," Proc. of Advances in Digital Libraries Conf. (ADL'98), Santa Barbara, CA, pp.19-29, April 1998.

[9] 김성자, 이도헌, "데이타마이닝 기술 및 연구동향," 한국정보과학회 정보과학회지, 제 16권, 제 9호, pp.6-14, 1998년 9월.

[10] 박종수, 유원경, 홍기형, "연관규칙 탐사와 그 응용," 한국정보과학회 정보과학회지, 제 16권, 제 9호, pp.37-44, 1998년 9월

[11] 윤종필, 김희숙, 최옥주, "데이타마이닝의 유용성," 한국정보과학회 정보과학회지, 제 16권, 제 9호, pp.15-23, 1998년 9월.