전자 카탈로그에 대한 효율적인 색인어 통계 정보 관리 방법

Efficient Management of Statistical Information of Keywords on E-Catalogs

이동주(Dongjoo Lee)*, 황인범(Inbeom Hwang)*, 이상구(Sang-goo Lee)**

초 록

전자 카탈로그는 상품이나 서비스 정보를 저장하고 있는 전자 문서로, 전자 상거래에서 가장 중요한 자료 중 하나이다. 전자 카탈로그는 지속적으로 추가, 수정 혹은 삭제되면서 최신의 상태로 유지되게 되는데, 전자 카탈로그의 양이 많아지면서 중복이 발생하고, 부적합한 분류에 할당되는 등, 품질 유지 문제가 발생한다. 검색, 중복확인, 자동분류는 카탈로그 품질 관리를 위해 중요한 기능들인데, 이 기능을 구현하기 위해서 카탈로그에서 추출된 색인어들의 통계 정보를 활용한 확률 모델들이 제시되었다. 그러나 이들은 서로 독립적으로 다루어 졌기에, 카탈로그 관리 시스템이라는 하나의 시스템에서 구현될 수 있음에도 불구하고, 각 모델들이 공유하는 데이터와 이를 관리하기 위한 데이터 관리 기법에 관한 연구는 미흡하였다. 따라서 본 논문에서는 세 기능을 위한 확률 모델을 정리하고, 이를 관계형 데이터베이스 상에서 구현하고, 통계 정보를 효율적으로 관리하는 방법을 제시한다. 특히, 실체화 뷰를 이용하여 불필요한 응용의 개발 비용과 데이터 무결성 저해 요인을 제거하였다. 다량의 실제 전자 카탈로그 데이터베이스에 대한 실험을 통해 관계형 데이터베이스를 이용한 구현이 속도와 정확성에 있어 실용성이 있음을 보였고, 응용을 통한 통계 정보 갱신 방법과의 비교를 통해 실체화 뷰를 활용한 통계 정보 관리 기법의 효용성을 보였다.

ABSTRACT

E-Catalogs which describe products or services are one of the most important data for the electronic commerce. E-Catalogs are created, updated, and removed in order to keep up-to-date information in e-Catalog database. However, when the number of catalogs increases, information integrity is violated by the several reasons like catalog duplication and abnormal classification. Catalog search, duplication checking, and automatic classification are important functions to utilize e-Catalogs and keep the integrity of e-Catalog database. To implement these functions, probabilistic models that use statistics of index words extracted from e-Catalogs had been suggested and the feasibility of the methods had been shown in several papers. However, even though these functions are used together in the e-Catalog management system, there has not been enough consideration about how to share common data used for each function and how to effectively manage statistics of index words. In this paper, we suggest a method to implement these three functions by using simple SQL supported by relational database management system. In addition, we use materialized views to reduce the load for implementing an application that manages statistics of index words. This brings the efficiency of managing statistics of index words by putting database management systems optimize statistics updating. We showed that our method is feasible to implement three functions and effective to manage statistics of index words with empirical evaluation.

본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 육성・지원사업(IITA-2009-C1090-0902-0031)의 연구결과로 수행되었음

* 서울대학교 전기・컴퓨터공학부

** 서울대학교 전기・컴퓨터공학부 교수

2009년 09월 23일 접수, 2009년 10월 06일 심사완료 후 2009년 11월 06일 게재확정.

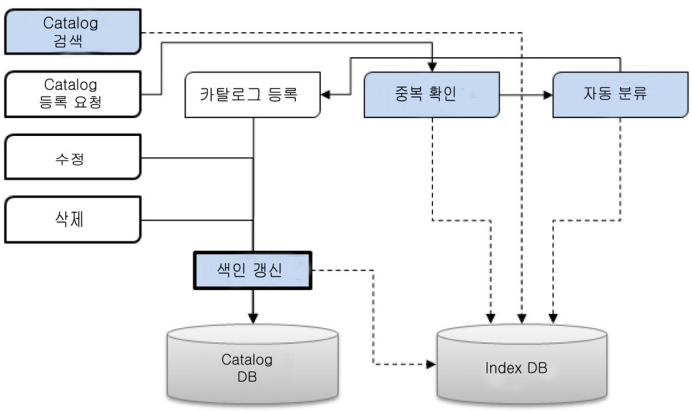
키워드 : 전자카탈로그, 관계형 데이터베이스, 정보검색, 중복확인, 자동분류, 실체화 뷰
E-Catalog, RDBMS, Information Retrieval, Duplication Check, Auto-Classification, Materialized View

1. 서 론

전자 상거래는 그 규모나 질 모두 비약적으로 발전하여 왔고, 전자 카탈로그의 유통은 그에 상응하게 증가하였다. 전자 카탈로그는, 전자 상거래 시스템에서 거래되는 상품이나 서비스 등의 품명과 코드, 생산 연월일, 규격, 특징 등의 제품 속성 정보와 가격, 배송 방법, 지급 방법 등의 판매에 필요한 다양한 정보를 저장하고 있는 목록으로 전자 상거래에서 가장 중요한 정보 중 하나이다.

전자 상거래 시스템을 이용하는 사용자는, 키워드를 이용하여 카탈로그를 '*검색*'함으로써, 상품에 대한 정확한 속성 정보나 값에 대한 지식 없이도, 순위를 기반으로 한 결과를 얻을 수 있다. 카탈로그 관리자는 관리 시스템을 이용해서 신규 카탈로그를 등록하거나 수정하고, 불필요한 카탈로그를 삭제하는 등의 카탈로그 관리 작업을 수행한다. 신규 카탈로그를 등록할 때, 관리자는 요청된 카탈로그 정보가 기존의 카탈로그와 동일한 정보를 나타내고 있는 지 확인하는 '*중복확인*' 작업을 거치고, 신규 카탈로그라고 판단되면, 어떠한 분류에 등록해야 하는지를 판단하게 된다. 이때에 최적 분류를 추천해주는 '*자동분류*'를 활용한다. <그림 1>은 이 같은 과정을 도식화 하여 보여주고 있다. 이같이 검색, 중복확인, 자동분류는 전자 상거래 시스템에서 전자 카탈로그를 활용하고 관리함에 있어 중요한 기능을 수행한다. 이전 연구에서, 이들을 구현하기 위해 확률 모델을 이용한 방법들이 제시되었고[6, 12-14], 확률 모델을 활용한 방법은 속도와 정확성에 있어서 실효성이 있음이 입증되었다.

전자 카탈로그는 대부분이 관계형 데이터베이스에서 관리되며, 이를 위한 다양한 모델



〈그림 1〉 카탈로그 등록 및 조회 프로세스 및 데이터베이스 접근

과 방법이 제시되었다[1, 2, 7]. 최근에는 온톨로지를 이용하여 그 활용성을 증대시키고자 하는 노력도 하고 있으나[3, 4, 8, 9], 여전히 관계형 데이터베이스가 전자 카탈로그를 저장하고 관리하는 주요 저장소로 사용되고 있다. 확률 모델을 이용하기 위해서는 <그림 1>에서 보이는 바와 같이 확률 모델을 위한 색인어 통계 정보를 추가적으로 저장해야 하고, 카탈로그 정보가 추가, 변경, 삭제될 때에 통계 정보 또한 지속적으로 갱신하여 카탈로그 정보와 통계 정보간의 일관성을 유지해야 한다. 색인어 통계 정보를 응용에서 사용하는 특정 파일 구조나 저장소에 저장할 경우, 응용에서 발생하는 에러나 예기치 않은 시스템의 문제가 발생하면 두 데이터 간의 정보 불일치로 인해 무결성이 저해될 수 있다. 또한, 이 세가지 기능을 따로 관리할 경우, 한 부분에서만 문제가 발생하더라도 데이터의 불일치가 발생하므로 무결성 저해 가능성이 증가하게 된다. 검색, 중복확인, 자동분류는 전자상거래 시스템에서 중요한 기능을 수행함에도 불구하고, 각기 따로 다루어져 왔고, 세 기능을 구현하는데 사용되는 통계정보를 관리하는 방법에 대해서도 심도 있게 다루어지지 않았다.

본 논문에서는 관계형 데이터베이스를 이용하여 위의 세 가지 기능을 통합하여 구현하는 방법을 제시하고자 한다. 특히, 관계형 데이터베이스에서 제공하는 실체화뷰를 이용하여 색인어 통계 정보를 관리함으로써, 응용 개발 비용을 줄일 뿐만 아니라, 응용에서 발생하는 데이터 무결성 저해 요인을 최소화하여 카탈로그 정보와 통계 정보간의 불일치를 최소화 하고자 한다. 이를 위해, 검색, 중복확인, 자동분류를 색인어 통계 정보를 저장하는 테이블에 대한 SQL질의로 표현하고, 각각의 질의를 위한 통계 정보를 저장하는 테이블을 정의하였다. 카탈로그 내의 색인어의 출현 빈도수나 카탈로그 내의 총 색인어 출현횟수와 같이 개별 카탈로그로부터 독립적으로 추출되는 정보를 *기초색인어통계정보*로 정의하고, 이를 테이블에 저장한다. 검색, 중복확인, 자동분류에서 필요한 각각의 통계 정보들은 이 기초색인어통계정보 테이블에 대한 실체화 뷰로 정의하여, 응용에서는 기초 색인어 통계 정보 테이블만을 갱신하도록 하여 응용에서의 처리 비용을 최소화 하였다. 다량의 실제 전자 카탈로그 데이터베이스를 이용하여 속도와 정확성에 대한 검증을 수행하였고, 응용에서 전체 색인어 통계 정보를 관리하는 방법과의 색인 갱신 속도를 비교하여 실체화 뷰를 이용한 방법의 효율성을 보였다.

본 논문의 구성은 다음과 같다. 먼저 제 2장에서 확률 모델에 기반한 검색, 중복확인, 자동분류가 확률 값 계산을 위한 통계 정보를 저장하는 테이블에 대한 SQL질의로 구현될 수 있음을 보인다. 제 3장에서는 통계 정보를 저장하는 테이블이 실체화 뷰로 관리될 수 있음을 보이고, 제 4장에서는 실험을 통해서 구현된 각 기능의 성능을 보이고, 실체화 뷰의 효용성을 검증한다. 끝으로 제 5장에서 결론을 맺는다.

2. 각 기능을 위한 확률 모델 및 SQL 질의를 통한 구현

전자 카탈로그는 상품을 기술하기 위한 속

성과 해당 속성에 대한 속성값으로 정의된다. 전자 카탈로그는 기업이나, 정의하는 단체에 따라서 약간씩의 차이를 보이지만, 대부분 UNSPSC[19]나 eCl@ss[16]와 같은 트리 구조의 분류체계 상에서 유사한 속성을 가진 상품들을 묶어서 관리할 수 있도록 하며, 분류에 따라서 상품의 특성을 기술하는 속성 집합이 달리 정의된다. 분류 또한 '소속 분류'라는 특별한 속성에 대한 속성값이라고 한다면, 전자 카탈로그 d는 속성과 속성값 쌍의 집합으로, 속성 집합 A를 n개의 속성을 가지는 속성 집합으로, $A=\{a_1, a_2, \cdots, a_n\}$와 같이 정의하면, 카탈로그는 각각의 속성에 대한 값을 가지는, 튜플(Tuple)로 $d=\{v_1, v_2, \cdots, v_n\}$와 같이 정의된다. 카탈로그 데이터베이스 D는 카탈로그의 집합으로 $D=\{d_1, d_2, \cdots, d_t\}$와 같이 정의된다. 이 같은 정의를 바탕으로, 전자카탈로그 데이터베이스를 대상으로 하는 검색, 중복확인, 자동분류를 정의한다. 본 절에서는 각 세 가지 기능을 지원하기 위한 확률 모델과, 이를 SQL질의로 구현하는 방법 및 이때에 필요한 색인어 통계 정보를 저장하는 색인 집합을 정의한다.

2.1 검 색

전자 카탈로그에 대한 검색은 주어진 키워드 질의 q로부터 질의가 나타낼 확률이 높은 카탈로그 목록 l_d를 카탈로그 데이터베이스 D로부터 반환하는 것으로 식 (1)과 같이 표현된다. 여기서 질의는 키워드의 집합으로 $q=\{kw_1, kw_2, \cdots\}$와 같이 정의된다.

$$l_d = search_p(D, q) \quad (1)$$

본 논문에서는 전자카탈로그에서의 검색 성능이 검증된[12] 신뢰망(Belief Network)[10]을 이용한 확률모델을 이용하여 주어진 키워드 질의에 대해서 정렬된 전자 카탈로그를 반환하도록 한다.

신뢰망을 이용하면, 주어진 질의 q가 카탈로그 d_j를 기술할 확률 $P(d_j|q)$은 식 (2)와 같이 정의된다.

$$P(d_j|q) = \frac{1}{P(q)} \sum_{\forall u} P(d_j \wedge q|u) \times P(u) \quad (2)$$
$$\sim \frac{1}{p(q)} \sum_{\forall u} p(d_j|u) \times P(q|u) \times P(u)$$

여기서 u는 신뢰망에서 질의와 카탈로그를 연결하는 색인어 집합이다. 이를 벡터공간(Vector Space)으로 확장하면 식(3)과 같이 변경된다.

$$P(d_j|q) \sim \frac{1}{p(q)} \sum_{\forall \vec{k}} P(d_j|\vec{k}) \times P(q|\vec{k}) \times P(\vec{k}) \quad (3)$$

여기서 $\vec{k_i}$를 단위 벡터로 $\vec{k}_i = \vec{k} | (g_i(\vec{k}) = 1 \wedge \forall_{j \neq i} g_i(\vec{k}) = 0)$와 같이 정의 하면, $P(d_j|\vec{k})$와 $P(q|\vec{k})$는 다음과 같이 정의된다($g_i(\vec{k}) = if\ w_i > 0\ then\ 1, otherwise\ 0$).

$$P(q|\vec{k}) = \begin{cases} \frac{w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2}} & if\ \vec{k} = \vec{k}_i \wedge g_i(q) = 1 \\ 0 & otherwise \end{cases}$$

$$\overline{P(q|\vec{k})} = 1 - P(q|\vec{k})$$

$$P(d_j|\vec{k}) = \begin{cases} \frac{w_{i,d_j}}{\sqrt{\sum_{i=1}^{t} w_{i,d_j}^2}} & if\ \vec{k} = \vec{k}_i \wedge g_i(d_j) = 1 \\ 0 & otherwise \end{cases}$$

$$\overline{P(d_j|\vec{k})} = 1 - P(d_j|\vec{k})$$

이를 바탕으로 $P(d_j|q)$는 식 (5)와 같이 표현될 수 있다.

$$P(d_j|q) \sim \frac{1}{P(q)} \sum_{i-1}^{t} \frac{w_{i,q}}{\sqrt{\sum_{k=1}^{t} w_{k,q}^2}} \times \frac{w_{i,d_j}}{\sqrt{\sum_{k=1}^{t} w_{k,d_j}^2}} \times P(\vec{k}_i) \quad (5)$$

여기서 질의와 전자 카탈로그는 질의와 전자 카탈로그를 기술하는 키워드에 대한 가중치를 가지는 벡터로 각각, $\vec{q} = (w_{q,k_1}, w_{q,k_2}, \cdots w_{q,k_t})$와 $\vec{d}_j = (w_{d_j,k_1}, w_{d_j,k_2}, \cdots w_{d_j,k_t})$와 같이 정의된다. 여기서 카탈로그 d에 대한 키워드 t에 대한 가중치 $w_{d,t}$는 $w_{d,t} = tf_{d,t} \cdot idf_t$와 같이 tf-idf[11]를 이용해서 구해지고, 각각 $tf_{d,t} = \frac{count(t \in d)}{\sum_{\forall t_i} count(t_i \in d)}$, $\mathrm{idf_t} = \log \frac{|\mathrm{D}|}{|\mathrm{t} \in \mathrm{d}|}$와 같이 색인어에 대한 통계 정보로부터 구해진다. 단위벡터의 출현 확률 $P(\vec{k}_i)$가 모든 단위벡터에 대해서 동일하다고 가정하면, $P(d_j|q)$ 값은 $\sum_{i-1}^{t} \frac{w_{i,q}}{\sqrt{\sum_{k=1}^{t} w_{k,q}^2}} \times \frac{w_{i,d_j}}{\sqrt{\sum_{k=1}^{t} w_{k,d_j}^2}}$에 의해서 결정되며, 질의와 카탈로그를 크기가 1인 벡터로 정규화하면, $search_p(D, q)$는 질의 벡터와 카탈로그 벡터의 내적을 기준으로 정렬하는 SQL질의로 다음과 같이 구현할 수 있다.

질의 벡터를 각 색인어의 구분자를 K_ID로 하고, 색인어에 대한 가중치를 W로 하여 S_QV(K_ID, W)의 형태를 가지는 테이블에 저장하고, 카탈로그 벡터를 카탈로그 구분자를 P_ID로 하여 S_CV(P_ID, K_ID, W)의 구조를 가지는 테이블에 저장한다면, $search_p(D, q)$는 아래와 같은 집계 SQL질의로 단순화 된다.

```
SELECT P_ID, SUM(S_QV.W*S_CV.W) PROB
    FROM S_QV JOIN S_CV ON S_QV.K_ID
    = S_CV.K_ID
  GROUP BY P_ID
  ORDER BY PROB DESC
```

따라서 질의 시점에 질의를 표현하는 S_QV가 생성된다면, 검색을 위해서는 S_CV를 최종형태로 하는 색인을 유지하면 된다. 이를 생성하고 관리하는 방법에 대해서는 제 3절에서 자세히 다룬다.

2.2 중복 확인

중복확인은 질의로 전자 카탈로그가 주어졌을 때, 기존의 카탈로그 데이터베이스에서 동일한 상품이나 서비스를 기술할 확률이 큰 카탈로그를 반환하는 것으로 식 (6)과 같이 정의된다.

$$l_d = checkDuplication_p(D, q) \quad (6)$$

여기서 질의는 카탈로그와 동일한 형태로 $d = < v_1, v_{2,} \cdots, v_n >$로 정의된다. 질의로 주어진 d가 기존에 저장된 상품 p를 기술할 확률은 베이스 정리(Bayes Theorem)에 의해서 다음과 같이 구해진다.

$$P(p|d) \sim \frac{P(p)}{P(d)} P(< v_1, v_2, \cdots, v_n > | p) \quad (7)$$

각 속성을 독립이라고 하면 이는 식 (8)과 같이 변형된다.

$$P(p|d) \sim \frac{P(p)}{p(d)} \prod_{i}^{n} P(a_i = v_i|p) \quad (8)$$

전자카탈로그 데이터베이스에 각 상품을 기술하는 카탈로그가 유일하게 존재한다면, 기존에 존재하는 카탈로그 $d_j = < v_{j,1}, v_{j,2,} \cdots, v_{j,n} >$가 d와 일치할 확률은 식 (8)에서 p를 d_j로 치환하는 것으로 구할 수 있다. 따라서 $P(a_i = v_i|d_j)$는 $P(v_i|d_j)$로 구할 수 있고, 이는 결국 $P(v_i|v_{j,i})$로 대체될 수 있다. 각 값에 대한 일치 확률은 검색에서와 마찬가지로 각 값으로부터 추출된 색인어로 확장된 신뢰망을 통해서 구해질 수 있고, 식 (9)와 같이 구해진다.

$$P(v_i|v_{j,i}) = \frac{1}{P(V_{j,i})} \sum_{\forall u} P(v_{j,i}|u) \times P(v_i|u) \times P(u) \quad (9)$$

식 (9)를 식 (8)과 결합하면 식 (10)을 얻을 수 있다.

$$P(d_j|d) \sim \frac{P(d_j)}{P(d)} \prod_i^n \cdot \left(\frac{P(\vec{k})}{P(v_{j,i})} \sum_{\forall k} p(v_{j,i}|\vec{k}) \times P(v_i|\vec{k}) \right) \quad (10)$$

검색에서와 마찬가지로 벡터 모델을 가정하면, 각 값은 추출된 키워드에 대한 가중치를 가지는 벡터로 $\vec{v} = < w_{k_1}, w_{k_2}, \cdots w_{k_t} >$와 같이 표현된다. 여기서 각 값의 색인어에 대한 가중치는 검색에서와 마찬가지로 tf-idf를 이용해서 $w_{v,t} = tf_{v,t} \cdot ivf_t$로 구해지고, 각 값은 $tf_{v,t} = \frac{count(t \in v)}{\sum_{\forall t_i} count(t_i \in v)}$와 $ivf_t = log \frac{|V|}{|\{t \in v\}|}$로 구해진다. 각 문서의 출현 확률이 $P(d_j) = P(d) = \frac{1}{|D|}$로 동일하다고 가정하고, 모든 색인 벡터의 출현 확률이 동일하고, 모든 값의 출현 확률이 동일하다고 가정하면, 식 (10)는 식 (11)과 같이 변경된다.

$$P(d_j|d) \sim \prod_i^n \left(\sum_{k=1}^t \frac{w_{i,k}}{\sqrt{\sum_{l=1}^t w_{l,q}^2}} \times \frac{w_{j,i,k}}{\sqrt{\sum_{l=1}^t w_{l,d_j}^2}} \right) \quad (11)$$

$checkDuplication_p(D, q)$는 $P(d_j|d)$를 기준으로 카탈로그를 정렬하는 것인데, SQL에서는 곱셈에 대한 집계를 지원하지 않으므로 이를 SQL로 표현할 수 없다. 그러나 식 (11)을 로그함수를 이용하여 식 (12)와 같이 변경하면, SQL을 이용해서 집계하는 것이 가능해진다. 또한 로그함수는 순증가 함수이므로 정렬 순서는 변하지 않는다.

$$log(P(d_j|d)) \sim \sum_i^n log \cdot \left(\sum_{k=1}^t \frac{w_{i,k}}{\sqrt{\sum_{l=1}^t w_{l,q}^2}} \times \frac{w_{j,i,k}}{\sqrt{\sum_{l=1}^t w_{l,d_j}^2}} \right) \quad (10)$$

질의 시에, 속성 구분자를 A_ID로 하여, 질의 벡터를 D_QV(A_ID, K_ID, W)의 형태의 테이블에 저장하고, 카탈로그를 D_CV(P_ID, A_ID, K_ID, W)에 저장하고 있다면, $checkDuplication_p(D, q)$은 다음과 같은 단순한 집계 SQL질의로 구현된다. 따라서 질의 시에 D_QV가 생성된다면, 중복확인을 위해서는 D_CV를 최종 색인으로 유지하면 된다.

```
SELECT P_ID, SUM(LOG(PROB)) LPROB
   FROM (
         SELECT P_ID, A_ID,
                SUM(S_QV.W*S_CV.W)
                PROB
```

```
            FROM D_QV JOIN D_CV ON
                 D_QV.A_ID =
                 D_CV.A_ID AND
                 D_QV.K_ID =
                 D_CV.K_ID
          GROUP BY D_CV.P_ID,
                 D_CV.A_ID
      )
GROUP BY P_ID
ORDER BY LPROB DESC
```

2.3 자동 분류

확률 모델을 이용한 카탈로그 자동분류는 주어진 카탈로그 d에 가장 적합한 분류 c_{MAX}를 찾는 것으로 식 (13)과 같이 정의된다.

$$C_{MAX} = \underset{c_j \in C}{argmax} P(c_j|d) \tag{13}$$

본 논문에서는 분류 기능을 질의로 주어진 카탈로그가 속할 확률이 큰 분류 목록 l_c을 반환하는 것으로 식 (14)와 같이 정의한다.

$$l_c = classify_p(D, q) \tag{14}$$

각 분류 c_j에 대해서 질의 카탈로그 $d = < v_1, v_2, \cdots, v_n >$가 속할 확률 $P(c_j|d)$는, 각 속성이 독립이라고 가정하면, 베이스 정리에 의해서 식 (15)와 같이 표현된다.

$$\begin{aligned} P(c_j|d) &= P(c_j| < v_1, v_2, \cdots, v_n >) \\ &\sim \frac{P(c_j)}{P(d)} \prod_i P(a_i = v_i|c_j) \end{aligned} \tag{15}$$

v_i가 n개의 색인어로 확장되어 $v_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,n}\}$와 같이 표현되고, 각각의 색인어가 독립이라고 가정하면 이는 식 (16)과 같이 표현될 수 있다($a_i = v_i$는 v_i로 줄여서 표현).

$$\begin{aligned} P(c_j|d) &\sim \frac{P(c_j)}{P(d)} \prod_i P(v_i|c_j) \\ &\sim \frac{P(c_j)}{P(d)} \prod_i \left(\prod_k P(t_{i,k}|c_j) \right) \end{aligned} \tag{16}$$

이에 필요한 각각의 확률 값은, $p(c_j) = \frac{|\{d \in c_j\}|}{|D|}$, $p(d) = \frac{1}{|D|}$, $P(t_{i,k}|c_j) = \frac{count(t_{i,k}, c_j)}{\sum_{\forall t_{i,k}} count(t_{i,k}, c_j)}$와 같이 정의되고, $classify_p(D, q)$는 $\log(P(c_j|d))$를 기준으로 정렬하는 집계 SQL질의로 구현된다.

$$\begin{aligned} log(P(c_j|d)) &\sim log|\{d \in c_j\}| \\ &+ \sum_i \sum_k log\left(\frac{count(t_{i,k}, c_j)}{\sum_{\forall t_{i,k}} count(t_{i,k}, c_j)} \right) \end{aligned} \tag{17}$$

또한 질의 카탈로그의 색인어 통계 정보를 이용한 스무딩 기법을 적용하여 다음과 같이 수정될 수 있다.

$$\begin{aligned} log(P(c_j|d)) &\sim log|\{d \in c_j\}| + \sum_i \sum_k log \\ &\cdot \left(\frac{count(t_{i,k}, c_j) + count(t_{i,k}, d)}{\sum_{\forall t_{i,k}} count(t_{i,k}, c_j) + \sum_{\forall t_{i,k}} count(t_{i,k}, d)} \right) \end{aligned} \tag{18}$$

질의 시점에 질의 카탈로그의 색인어 통계 정보가 C_QTF(A_ID, K_ID, CNT) 형태와 C_QLEN(A_ID, LEN)으로 주어지면, $classify_p(D, q)$는 각 분류에 속한 카탈로그의 수를 저장하는 색인 C_PCNT(C_ID, CNT), 각 분류의 속성에 각 색인어가 출현한 빈도수를 저장하는 색인 C_TF(C_ID, A_ID, K_ID, CNT), 각 분류의 각 속성에서 색인어의 총 출현횟수를 저장하는 색인 C_LEN(C_ID, A_ID, LEN)

을 이용해서 다음과 같은 집계 SQL질의로 구현할 수 있다. 여기서 C_ID는 분류 구분자이고, CNT는 색인어의 출현횟수, LEN은 총 출현횟수를 의미한다.

```
SELECT A.C_ID, LPROB + LOG(C_PCNT.
          CNT) LPROB
   FROM (
SELECT C_QTF.C_ID, SUM( LOG
          ((C_QTF.CNT + C_TF.CNT) /
          (C_QLEN.LEN + C_LEN.LEN))) )
          LPROB
   FROM C_QTF JOIN CQLEN ON
          C_QTF.A_ID = C_QLEN.A_ID
          JOIN C_TF ON C_QTF.A_ID =
          C_TF.A_ID AND C_TF.K_ID =
          C_QTF.K_ID
          JOIN C_LEN ON C_TF.A_ID =
          C_LEN.A_ID
GROUP BY C_QTF.C_ID ) A JOIN
         C_PCNT ON A.C_ID =
         C_PCNT.C_ID
ORDER BY LPROB DESC
```

3. 색인 집합 도출 및 실체화 뷰를 이용한 관리

일반적으로, 상품 정보는 공통 속성과 분류별 속성으로 구분된다. 공통 속성은 분류에 관계 없이 모든 카탈로그가 가지는 속성이고, 분류별 속성은 분류에 따라서 달리 가지는 속성으로 이를 관계형 데이터베이스에서 효율적으로 저장하고 관리하는 방법에 대한 연구가 있었다[7]. 공통 속성은 하나의 테이블의 각 컬럼에 속성값을 저장하도록 하고, 분류별 속성에 대한 저장 구조를 달리 한다. 공통 속성의 일종인 '소속 분류'는 카탈로그 구분자와 함께 공통 속성 테이블에 저장되는데, 여기서는 공통 속성 테이블을 P_COM(P_ID, C_ID, V_1, V_2, V_3, …)으로 정의한다. 여기서 P_ID는 카탈로그 구분자이고, C_ID는 분류 구분자를 말한다. V_i는 공통 속성 A_i에 대한 속성값이다.

카탈로그에서의 색인어 추출은 응용에서 수행되는데, 응용은 속성값에서의 특정 색인어의 출현횟수나 출현빈도와 같이 각 카탈로그별로 추출할 수 있는 색인어에 대한 통계 정보만을 추출하여 '기초색인어통계정보' 테이블에 저장한다. 이 같은 정책은 카탈로그에 대한 생성, 수정, 삭제가 발생할 때 각 기능에서 필요에 의해 정의한 통계 정보를 응용에서 직접 갱신하지 않고, 기초색인어통계정보 테이블에 대한 실체화 뷰로 생성되어 데이터베이스 관리 시스템에서 관리될 수 있도록 한다. 기초 색인어통계정보 테이블의 구조는 P_TF(P_ID, A_ID, K_ID, CNT, TF)로 정의되는데, 각 카탈로그(P_ID)의 각 속성값(A_ID)에서 추출된 색인어(K_ID)의 출현횟수(CNT)와 출현빈도(TF)를 저장한다.

아래의 각 하위 절에서는 검색, 중복확인, 자동분류를 위한 통계 정보를, 위에서 정의한 P_COM, P_TF에 대한 실체화 뷰로 정의할 수 있음을 보이고, 이를 위해 각 기능에서 필요한 정보로부터 뷰를 위해 필요한 테이블을 역으로 추적하면서 최종적으로는 P_COM, P_TF만으로 모두 생성할 수 있음을 보인다.

3.1 검색을 위한 색인 집합

검색을 위해서는 각 카탈로그의 정규화된 벡터 정보를 유지하는 테이블 S_CV(P_ID,

K_ID, W)가 필요하다. 이는 카탈로그의 벡터 테이블 S_COV(P_ID, K_ID, W)과 각 카탈로그 벡터의 크기를 저장하는 테이블 S_CAV (P_ID, AV)로부터 다음과 같이 정의된다.

```
SELECT S_COV.P_ID, S_COV.P_ID,
       S_COV.W/S_CAV.AV W
  FROM S_COV JOIN S_CAV ON
       S_COV.P_ID = S_CAV.P_ID
```

S_CAV는 S_COV에 대한 집계 SQL질의로 정의된다.

```
SELECT P_ID, POW(SUM(POW(W, 2)), 0.5)
       AV
  FROM S_COV
 GROUP P_ID
```

S_COV는 각 카탈로그의 어휘의 TF를 저장하는 S_TF(P_ID, K_ID, TF)와 각 어휘의 IOF를 저장하는 S_IOF(K_ID, IOF)로부터의 SELECT문으로 정의된다.

```
SELECT S_TF.P_ID, S_TF.K_ID, TF*IOF W
  FROM S_TF JOIN S_IOF ON S_TF.K_ID
       = S_IOF.K_ID
```

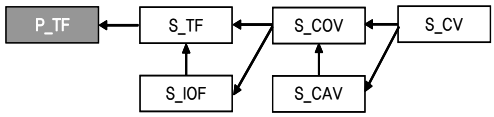
S_IOF는 S_TF에 대한 다음과 같은 집계 SQL질의로 정의된다.

```
SELECT K_ID, LN((SELECT COUNT(*)
FROM P_COM)/COUNT(P_ID)) IOF
  FROM S_TF
 GROUP BY K_ID
```

S_TF는 P_TF로부터 다음과 같은 SQL질의로 정의된다.

```
SELECT S_KCNT.P_ID P_ID, S_KCNT.K_ID
       K_ID, S_KCNT.CNT/S_PLEN.PLEN
       TF
  FROM (SELECT P_ID, K_ID, SUM(CNT)
       CNT FROM P_TF GROUP BY
       P_ID, K_ID) S_KCNT JOIN
       (SELECT P_ID, SUM(CNT) PLEN
       FROM P_TF GROUP BY P_ID)
       S_PLEN
       ON S_KCNT.P_ID = S_PLEN.P_ID
```

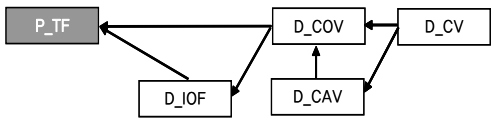
즉 최종적으로 필요한 색인 S_CV는 <그림 2>에서 보이는 바와 같이 여러 단계를 거쳐 P_TF로부터 실체화 뷰로 관리될 수 있다.



〈그림 2〉 검색을 위한 색인 집합의 구성 및 의존성

3.2 중복확인을 위한 색인 집합

중복확인은 검색과 유사하지만, 최종색인 D_CV(P_ID, A_ID, K_ID, W)가 S_CV와는 달리 키로 A_ID를 가지고 있다. 따라서 D_COV, D_CAV, D_IOF를 생성하기 위한 SELECT시에 GROUP절에 A_ID를 추가하는 것으로 쉽게 확장할 수 있다. 따라서 중복확인에서의 색인 집합 도출에 대한 자세한 설명은 생략한다. 최종적으로 <그림 3>에서 보이는 바와 같이 색인 집합들이 구성된다. 이때, 이미 P_TF가 각 속성별로 색인어의 출현횟수를 저장하기 때문에 검색에서와 같이 S_TF를 생성하는 과정이 필요 없다.



〈그림 3〉 중복확인을 위한 색인 집합의 구성 및 의존성

3.3 자동분류를 위한 색인 집합

자동 분류를 위해서는 제 2.3절에서 도출한 바와 같이 최종적으로 C_TF, C_LEN, C_PCNT의 세 가지 통계 정보를 유지해야 한다. 먼저, C_TF는 P_TF와 P_COM을 통한 다음과 같은 집계 SQL질의로 정의된다.

```
SELECT P_COM.C_ID, P_TF.A_ID,
       P_TF.K_ID, SUM(CNT) CNT
  FROM P_TF JOIN P_COM ON
       P_TF.P_ID = P_COM.P_ID
 GROUP BY P_COM.C_ID, P_TF.A_ID,
       P_TF.K_ID
```

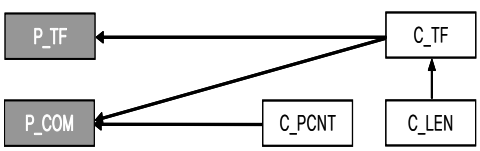
C_LEN은 C_TF에서의 집계 SQL질의로 정의된다.

```
SELECT C_ID, A_ID, SUM(CNT) LEN
   FROM C_TF
 GROUP C_ID, P_TF.A_ID
```

C_PCNT는 P_COM에 대한 집계 SQL질의로 정의된다.

```
SELECT C_ID, COUNT(*) PCNT
 FROM P_COM
 GROUP C_ID
```

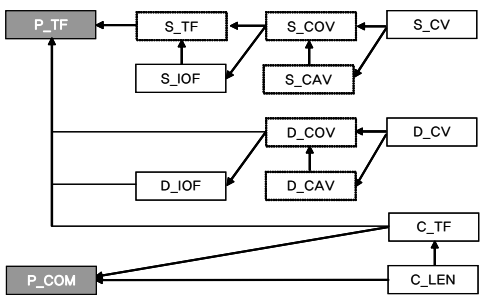
최종적으로 자동분류를 위한 색인어 집합은 <그림 4>와 같이 구성된다.



〈그림 4〉 자동분류를 위한 색인 집합의 구성 및 의존성

3.4 통합 색인 집합

위에서 도출한 각각의 색인집합을 통합하면, <그림 5>와 같이 통합 색인 집합을 구성할 수 있다. 여기서 회색으로 칠해진 테이블은 실제로 카탈로그의 생성, 변경, 삭제가 발생했을 때 응용에서 직접 그 값을 입력, 삭제, 수정하는 테이블을 나타내고, 흰색 상자는 실체화 뷰를 나타낸다. 실선 테두리의 실체화 뷰는 각 기능을 위한 SQL질의문에서 직접 사용되는 통계 정보를 저장하고, 점선 테두리의 실체화 뷰는 최종 색인을 생성하기 위한 중간 정보로만 사용된다.



〈그림 5〉 통합 색인 집합

본 논문에서 자세한 설명을 하지는 않지만, 검색을 위한 SQL문은 D_CV와 S_IOF를 이용해서 새로 작성될 수 있다. 또한 S_IOF는 P_TF로부터 생성될 수 있기 때문에 검색을 위한 색인 정보는 중복확인을 위한 색인과 통합되어 관리될 수 있다. Oracle과 같은 관계형 데이터베이스 관리시스템은 SELECT에서의 부분 집계를 지원하는데[18], 이를 활용하면, S_IOF, D_IOF, D_CV, C_TF, C_LEN, C_PCNT의 각 기능에서 필요로 하는 최소한의 색인 집합을 구축하고 유지할 수 있다.

각 기능에서 질의 시점에 질의로 제공된 키워드나, 카탈로그 정보로부터 색인어를 추출하고 추출된 색인어의 통계 정보와 S_IOF, D_IOF와 같은 기 구축된 통계 정보를 이용해서 최종적으로 필요한 값을 도출한다. 이 과정은 각 기능에서 각 카탈로그에 대한 최종적인 통계 정보를 생성하는 과정과 같으므로 생략한다.

4. 성능 실험

제시한 방법에 대한 효용성을 검증하기 위해서 다량의 실제 전자 카탈로그에 대한 실험을 수행하였다. 실험은 크게 두 가지로 나누어지는데, SQL질의로 구현된 각 기능의 정확도와 속도에 대한 것과 카탈로그가 갱신되었을 때 색인어 통계 정보를 갱신하는 속도에 대한 것이다.

4.1 실험환경

데이터는 조달청[17]에서 관리하고 있는 전자 카탈로그에서 추출한 50만개를 이용하였다. 전체 카탈로그가 직접적으로 속한 분류는 약 8,000개이고, 속성은 약 20,000개이다. 카탈로그 수에 따른 정확도와 성능을 달리 하기 위해서, 카탈로그를 임의로 추출하여 10,000개, 50,000개, 100,000개, 250,000개, 500,000개로 구분하여 실험을 수행하였다. <표 1>은 각 실험 데이터 집합의 간단한 통계 정보를 보여준다. 각 카탈로그당 생성된 기초색인어의 수는 약 30개이다.

세 가지 기능은 SQL질의로 구현되기 때문에, 데이터베이스에의 질의 응답 시간을 측정하는 것으로 속도를 측정하였다. DBMS는 Oracle 10G Enterprise Edition을 이용하였다. DBMS를 구동하는 서버는 Intel® CoreTM2 Quad CPU 2.83GHz를 탑재하고, 8.00GB의 메모리를 가진다. 실제로 질의는 개인용 컴퓨터에서 JDBC를 이용해서 다량의 질의를 지속적으로 수행하였으나, 통신속도나 개인용 컴퓨터의 성능에는 크게 영향을 받지 않기에 이에 대한 기술은 생략한다.

4.2 정확성 및 속도

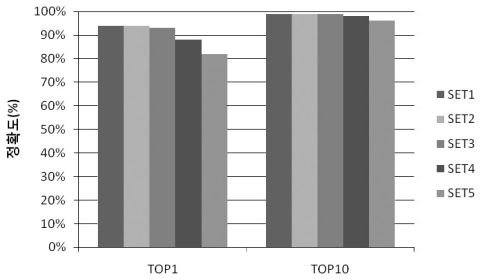
4.2.1 검색

검색 성능을 확인하기 위해서, 각 실험셋에 포함된 카탈로그의 공통 속성중에서 상품

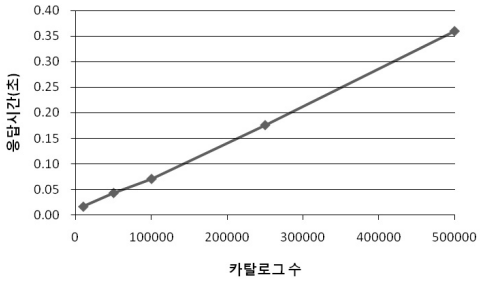
〈표 1〉 실험 데이터 특성

실험셋	카탈로그 수	분류수	속성수	기초색인어통계정보 테이블의 튜플수	상위 100개 색인어당 평균 카탈로그 수
SET1	10,000	2787	6845	307674	625
SET2	50,000	5312	13052	1524115	3054
SET3	100,000	6494	15931	3066546	6141
SET4	250,000	7506	18450	7601089	15291
SET5	500,000	7963	19633	15246576	30564

의 이름과 규격을 기술하는 속성값을 질의로 하여 검색 질의를 수행하였다. 추출한 해당 카탈로그가 정렬된 상품 집합에서의 상위1개(TOP1)와 상위10(TOP10)개 이내에 나오는지 여부로 정확도를 측정하였다. 속도는 각 질의가 완료되는 시간을 측정하였고, 총 100회의 질의를 수행하여 평균 수행 시간을 계산하였다.



〈그림 6〉 검색 정확도

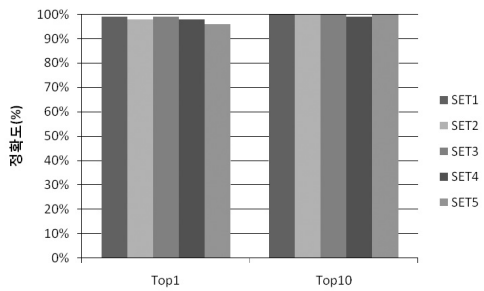


〈그림 7〉 검색 응답 시간

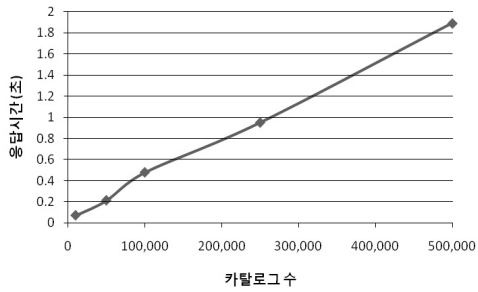
<그림 6>은 검색 정확도를 보여주는데, 상품명을 사용했을 때, 대부분의 경우 상위 10개 이내에서 해당 카탈로그를 확인할 수 있음을 보여주는데, 카탈로그 검색에 대한 확률 모델의 효율성을 확인할 수 있다. 그러나 여기서 대상 카탈로그 수가 증가할수록 TOP1 정확도가 떨어지는 것을 볼 수 있는데, 이는 카탈로그 수가 증가하면, 모델상에서 정의된 확률값 $P(d)=\frac{1}{|D|}$로 인해 질의가 특정 카탈로그를 지칭할 확률이 떨어지기 때문으로 해석할 수 있다. 그림7에서 보이는 바와 같이 응답 시간은 0.4초 이내였다. 이는 50만개의 대량의 카탈로그에 대해서도 실제로 사용할 때에 전혀 불편함이 없는 속도로, 대부분의 기업에서 특별한 검색 시스템을 도입하지 않고도 색인어를 추출하여 검색 기능을 구현할 수 있는 성능이다. 여기서 데이터의 수가 증가하면 응답시간이 선형으로 증가하는 것을 확인할 수 있는데, SQL의 실행 계획을 분석한 결과 검색된 카탈로그의 수가 증가하면, 이를 정렬하는 시간이 증가하여 나타나는 현상으로 해석된다.

4.2.2 중복확인

중복확인은 직접 데이터베이스에 포함된 카탈로그를 질의로 하여 성능을 확인하였다. 분류별 속성은 카탈로그의 구조적 특징을 반영하게 되므로, 공통 속성 값만을 질의로 하여 중복된 카탈로그를 선별할 수 있는지를 확인하였고, 이에 대한 100회 반복 질의의 평균 속도를 측정하였다.



〈그림 8〉 중복확인 정확도

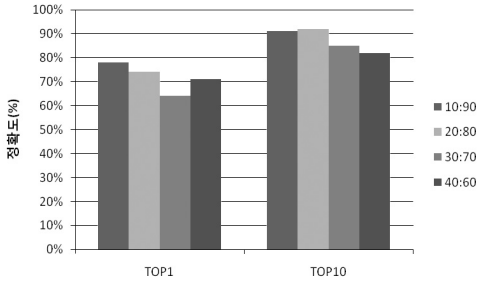


〈그림 9〉 중복확인 응답 시간

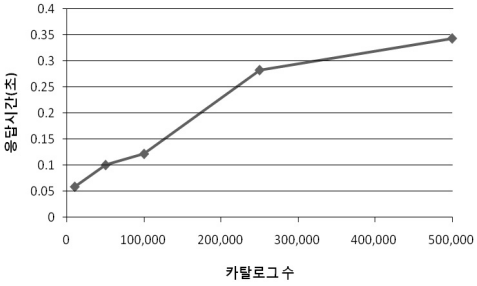
<그림 8>에서 보이는 바와 같이 전체 정보를 활용하지 않아도 거의 정확하게 동일한 카탈로그를 확인할 수 있었다. 그러나 시간은 검색보다는 오래 걸렸는데, 이는 검색에서보다 질의를 구성하는 색인어의 수가 많고, 정렬대상이 되는 카탈로그의 수가 많아지는 것 때문이라고 해석할 수 있다. 그러나 정확도를 확보할 수 있고, 2초 이내의 응답 시간을 갖기 때문에 충분히 활용할 수 있는 성능이다. 또한, 검색 질의를 구성하는 색인어 중에서 흔하게 출현하는 색인어를 제외하여 정렬 대상이 되는 카탈로그의 수를 줄이는 방법 등으로 응답 시간을 줄일 수 있을 것이라 판단한다.

4.2.3 자동분류

자동분류에 대한 응답 속도는 검색이나 중복확인과 동일하게 총 카탈로그의 수를 변경하면서 측정하였다. 그러나 자동분류의 정확도는 실험셋 3을 학습군과 실험군의 비율을 9 : 1, 8 : 2, 7 : 3, 6 : 4로 변경하면서 성능 변화를 측정하였다. 또한, 실험의 편의를 위해서 실험군 중에서 임의의 100개에 대해서만 질의를 수행하고 상위1개와 10개 이내에 해당 분류를 산출하는지 여부로 정확도를 측정하였다.



〈그림 10〉 자동분류 정확도

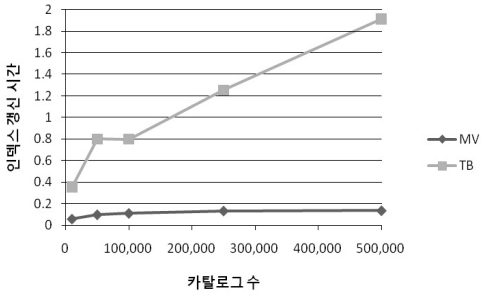


〈그림 11〉 자동분류 응답 시간

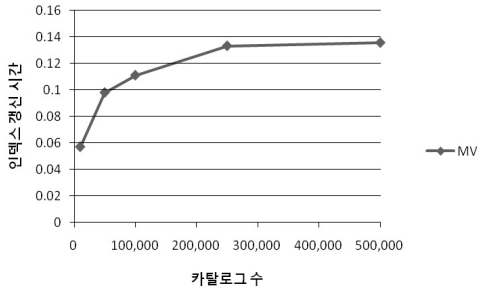
<그림 10>에서 보는 바와 같이 TOP1정확도는 학습군이 90%일 때, 대략 80%로 [14]에서 보이는 정확도와 비슷한 값을 가진다. 그러나 이는 [15]에서 제시한 개선된 방법을 적용하지는 않은 것으로 적용한다면 정확도가 개선될 것이라 예상한다. 본 논문에서는 정확도를 개선하는 것이 목적이 아니기에 중요하게 다루지 않는다. 속도는 카탈로그 수가 증가할수록 응답 시간이 증가하는 것을 확인할 수 있는데, 이는 검색이나 중복확인과 마찬가지로 정렬 하는 시간이 증가하기 때문인 것으로 해석된다. 응답 시간이 0.4초 이내로 매우 빠른 시간에 반응하므로 실질적으로 전자상거래 시스템에서 사용하는 데에 전혀 문제가 없는 속도이다.

4.3 실체화 뷰를 활용한 통계 정보의 갱신

실체화 뷰의 효율성을 보이기 위해서, 응용에서 각각의 색인어 통계를 담는 테이블에 대한 갱신 작업을 직접 처리하는 형태와 비교하였다. 카탈로그가 변경될 경우, 기존 색인어를 삭제하고 새로운 색인어를 추가한다. 따라서, 카탈로그의 추가와 삭제에 대해서만 색인 갱신 시간을 측정하였다. 임의의 순서로 카탈로그 추가와 삭제에 대해서 총 100회의 변경을 실험하였고 평균 색인 갱신 시간을 계산하였다.



〈그림 12〉 중복확인 정확도



〈그림 13〉 중복확인 응답 시간

<그림 12>에서 보이는 바와 같이 실체화 뷰를 이용한 경우(MV) 카탈로그의 수에 상관없이 0.2초 이내에 전체 색인어 통계 정보가 갱신되는 반면, 응용에서 처리하는 경우(TB)에는 이보다 훨씬 오래 걸림을 알 수 있다. 또한, 실체화 뷰를 이용한 경우에는 그림13에서 보이는 바와 같이 카탈로그 수가 증가해도 갱신 시간이 크게 증가하지 않는데, 이는 DBMS가 최적화하여 인덱스를 효율적으로 이용하기 때문인 것으로 파악할 수 있다. 실체화 뷰를 이용하는 경우에는 초기에 실체화 뷰를 생성하는 작업을 수행하면, 응용에서는 기초색인어통계정보 테이블만을 갱신하면 되기 때문에 응용의 코드 라인이 줄어든다. 실제로 실험에서 사용한, Java언어로 작성된 응용은, 실체화 뷰를 사용한 경우에는 기초색인어통계정보를 갱신하는 150줄 정도의 코드만이 필요했고, 응용에서 모두 처리하는 경우 350줄 정도의 코드가 추가로 필요하였다. 본 실험에서 확인한 바와 같이 색인어 통계 정보를 실체화 뷰를 이용하여 관리하는 것은 응용 개발 비용과 갱신 속도 모두에서 매우 효율적이다.

5. 결 론

본 논문에서는 전자 카탈로그를 관리하고 활용하는데 있어서 중요한 기능들인 검색, 중복확인, 자동분류를 관계형 데이터베이스의 SQL질의문을 이용해서 구현하는 방법을 제시하였다. 이는 확률 모델로 표현된 각 기능을 SQL질의로 구현하고, 각 모델에서 필요한 확률 값을 구하기 위한 통계 정보를 실체화 뷰를 이용하여 관리하는 방법이다. 실제 50만 개의 카탈로그를 포함하는 카탈로그 데이터베이스에서의 실험을 통해서 제시한 구현 방

법이 속도와 정확도 측면에서 모두 효용성이 있음을 입증하였다. 또한 실체화 뷰를 이용한 방법이 응용에서 직접 색인어 통계 정보를 갱신하는 방법보다 통계 정보를 관리하는데 있어서 속도와 응용 개발 비용 모든 측면에서 효율적임을 보였다.

전자 카탈로그에 대한 검색, 중복확인, 자동분류는 카탈로그를 활용하고 관리하는데 있어서 핵심적인 기능임에도 불구하고 기존의 연구들은 이를 따로 다루었다. 따라서 이 세 기능에 필요한 통계 정보를 정의하고 통합하여 통합 색인을 관리하고자 한 시도는 본 연구가 처음이다. SQL질의를 이용하여 각 기능을 구현하는 경우, 실체화 뷰나 질의 최적화와 같이 데이터베이스 관리 시스템에서 제공하는 기능을 쉽게 사용할 수 있다는 장점도 있지만, 질의를 수행할 수 있는 형태로 각 통계 정보를 정의해주어야 한다는 단점이 있다. 또한 Top-K질의 처리시 최적화를 적용하지 못하는 문제도 있다. 중복확인과 같이 다량의 데이터를 처리하는 경우에 Top-K 정렬 최적화를 수행하면, 성능 개선을 할 수 있음에도 이를 SQL질의로 사용하기 때문에 이 같은 제약이 따른다. 따라서 이를 극복할 수 있는 방법을 고안하거나, 데이터베이스 관리 시스템에서 이를 지원하게 되면 본 방법이 더욱 실용적일 것이라 생각한다.

참 고 문 헌

[1] Dongkyu Kim, Sang-goo Lee, “Catalog Management in e-Commerce Systems,” In Procd. CST 2003, 2003.

[2] Dongkyu Kim, Sang-goo Lee, Jonghoon Chun, Juhnyoung Lee, “A Semantic Classification Model for E-Catalogs,” In Procd. CEC 2004, 2004, pp. 85-92.

[3] Dongkyu Kim, Sang-goo Lee, Jonghoon Chun, Zoonky Lee, Heungsun Park, “A Practical Ontology for Product Information Management,” In Procd. of iiWAS 2005, 2005, pp. 217-222.

[4] Dongkyu Kim, Sang-goo Lee, Junho Shim, Jonghoon Chun, Zoonky Lee, Heungsun Park, “Practical Ontology Systems for Enterprise Application,” ASIAN 2005, 2005, pp. 79-89.

[5] Hesham Saadawi, “Universal e-catalog pattern,” In Procd. 2006 Conference on Pattern Languages of Programs, pp. 1-8, ACM Press, New York, 2006.

[6] Jae-won Lee, Taehee Lee, Sang-keun Lee, Ok-ran Jeong, Sang-goo Lee, “Massive Catalog Index based Search for e-Catalog Matching,” In Procd. CEC/EEE 2007, 2007, pp. 341-348.

[7] Kiryoong Kim, Dongkyu Kim, Jeuk Kim, Sang-uk Park, Ig-hoon Lee, Sang-goo Lee, “An Experimental Evaluation of Dynamic Electronic Catalog Models in Relational Database Systems,” Infomation Resources Management Association International Conference 2002, 2002, pp. 323-325.

[8] Martin Hepp, “ProdLight : A Lightwei-

ght Ontology for Product Description Based on Datatype Properties," LNCS Vol. 4439, Springer, 2007, pp. 260-272.

[9] Martin Hepp, "GoodRelations : An Ontology for Describing Products and Services Offers on the Web," LNCS, Vol. 5268, Springer, 2008, pp. 332-347.

[10] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, "Modern Information Retrieval," ACM Press, New York, 1999, pp. 56-60.

[11] Spärck Jones, Karen, "A statistical interpretation of term specificity and its application in retrieval," Journal of Documentation, Vol. 28, No 1, MCB UP Ltd, 1972, pp. 11-21.

[12] Taehee Lee, Jonghoon Chun, Junho Shim, "Sang-goo Lee, An Ontology-Based Product Recommender System for B2B Marketplaces," International Journal of Electronic Commerce, Vol. 11, No. 2, 2006, pp. 125-155.

[13] Y. Ding, M. Korotkiy, B. Omelayenko, V. Kartseva, V. Zykov, M. Klein, E. Schulten, D. Fensel, "GoldenBullet : Automated Classification of Product Data in E-commerce," BIS 2002, 2002.

[14] Young-gon Kim, Taehee Lee, Jonghoon Chun, Sang-goo Lee, "Modified Naive Bayes Classifier for E-Catalog Classification," LNCS Vol. 4055, 2006, pp. 246-257.

[15] Young-gon Kim, Taehee Lee, Sang-goo Lee, Jong-Heung Park, "Exploiting Attribute-Wise Distribution of Keywords and Category Dependent Attributes for E-Catalog Classification," LNCS, Vol. 5226, 2008, pp. 985-992.

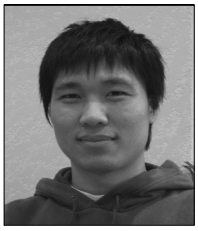
[16] eCl@ss : eCl@ss White Paper, V0.6, 2001, (Accessed on, Sept. 2009) http://www.eclass.de.

[17] KOCIS (Korea Ontology-based e-Catalog Information System) (Accessed on, April 2009), http://www.g2b.go.kr : 8100.

[18] Oracle® Database SQL Reference 10g Release 1 (10.1) Part Number B10759-01.

[19] UNSPSC : Why Coding and Classifying Products is Critical to Success in Electronic Commerce, Using the UNSPSC, White Paper, Granada Research, 2001.

저 자 소 개



이동주 (E-mail : therocks@europa.snu.ac.kr)
2003년 서울대학교 응용생물화학부 (학사)
2003년～현재 서울대학교 전기・컴퓨터공학부 (석박 통합 과정)
관심분야 e-Business Technology, Database, Semantic Web, Web 2.0 Context-Awareness, IR, NLP, Ontology

황인범 (E-mail : inbeom@europa.snu.ac.kr)
2009년 서울대학교 전기・컴퓨터공학부 (학사)
2009～현재 서울대학교 전기・컴퓨터공학부 (석사 과정)
관심분야 Semantic Web, IR, Web2.0 Development, NLP

이상구 (E-mail : sglee@europa.snu.ac.kr)
1985년 서울대학교 전산학과 (학사)
1987년 Computer Science, Northwestern University (석사)
1990년 Computer Science, Northwestern University (박사)
1992년～현재 서울대학교 전기・컴퓨터공학부 교수
2002년～현재 CEBT 센터장
관심분야 e-Business Technology, Database, Ontology, Semantic Web, Web2.0