계층적 클러스터링 기법을 이용한 확장 불리언 모델의 적합성 피드백 방법

(Relevance Feedback Method of an Extended Boolean Model using Hierarchical Clustering Techniques)

최 종 필 † 김 민 구 ††
(JongPill Choi) (Minkoo Kim)

요 약 적합성 피드백 방법은 다음 검색 질의어와 검색 성능을 향상시키기 위해 사용자로부터 획득된 정보를 사용한다. 일반적으로 적합성 피드백 방법은 사용자로부터 획득된 정보를 새로운 질의어에 추가될 새로운 단어를 찾거나 질의어에 존재하는 단어의 가중치를 조정하는데 사용한다. 그러나 확장 불리언 검색 모델에서 적합성 피드백은 이것들뿐만 아니라 질의어에 있는 단어들을 적절하게 불리언 연산자(AND/OR)로 연결시켜야 한다. Salton과 그의 동료들은 확장 불리언 모델을 위한 DNF(disjunctive normal form) 방법이라 불리는 적합성 피드백 방법을 제안하였다. 그렇지만 이 방법은 질의어를 재구성할 때 심각한 문제점을 갖고 있다. 이 논문에서는 DNF 방법의 문제점을 조사하고 이러한 문제점을 극복하기 위해 계층적 클러스터링 기법을 이용한 적합성 피드백 방법을 제안한다. 그리고 두개의 실험 데이타 집합인 TREC 1의 DOE 컬렉션과 Web TREC 10 컬렉션을 이용하여 제안한 방법의 우수성을 보였다.

키워드 : 확장 불리언 모델, 적합성 피드백, 계층적 클러스터링

Abstract The relevance feedback process uses information obtained from a user about an initially retrieved set of documents to improve subsequent search formulations and retrieval performance. In the extended Boolean model, the relevance feedback implies not only that new query terms must be identified, but also that the terms must be connected with the Boolean AND/OR operators properly. Salton et al. proposed a relevance feedback method for the extended Boolean model, called the DNF (disjunctive normal form) method. However, this method has a critical problem in generating a reformulated queries. In this study, we investigate the problem of the DNF method and propose a relevance feedback method using hierarchical clustering techniques to solve the problem. We show the results of experiments which are performed on two data sets: the DOE collection in TREC 1 and the Web TREC 10 collection.

Key words : Extended Boolean Model, Relevance Feedback, Hierarchical Clustering

1. 서 론

오래 동안 연구되어온 적합성 피드백 방법의 주 아이디어는 검색된 문서들의 사용자 적합성 판단을 기반으로 질의어를 재구성함으로써 검색 성능을 향상시키는 것이다[1-3]. 이러한 방법들은 대부분 정보검색을 위한 벡터 모델에 기반을 둔다. 전통적인 불리언 모델의 대안으로써 벡터 모델은 문서와 질의어를 불리언 연산자를 사용하지 않고 벡터로 표현하도록 개발되었다. 그러나 불리언 연산자들에 위해 제공되는 구조가 없어졌기 때문에 벡터 모델에서 검색 방법론은 전통적인 불리언 방법과 호환성이 없어졌다. 이러한 문제점을 극복하고자, 즉, 벡터 모델의 장점들을 불리언 모델에 적용하기 위해 많은 연구가들은 불리언 검색 모델의 확장에 대하여 연구하였다[4-7]. 특히 Salton과 그의 동료들이 제안한 P-norm 확장 불리언 모델은 다른 어떤 확장 불리언 모델보다 뛰어난 성능을 갖는다는 것이 알려져 있다[8].

확장 불리언 모델에서 적합성 피드백 방법은 새로운 질의 단어들을 식별하고 가중치를 재조정하는 것뿐만 아니라 질의 단어들을 적절하게 불리언 AND/OR 연산

· 본 논문은 과학기술부의 국가지정연구실 사업의 일환으로 지원 받아 수행되었음(과제번호 M10302000087-03J0000-04400)
† 비 회 원 : 아주대학교 정보통신연구소 연구원
cjp@ajou.ac.kr
†† 종신회원 : 아주대학교 컴퓨터공학과 교수
minkoo@ajou.ac.kr
논문접수 : 2004년 3월 9일
심사완료 : 2004년 8월 16일

자로 연결해야만 한다. Salton 등은 DNF(disjunctive normal form) 방법이라 불리는 확장 불리언 모델을 위한 적합성 피드백 방법을 제안하였다[9]. 그러나 이 방법은 사용자가 재구성된 질의어로 검색될 문서의 수를 추정해야만 하는 심각한 가정을 포함하고 있다. 이 가정은 다음과 같은 두 가지 문제점을 불러일으킬 수 있다. 첫째로, 사용자가 검색된 문서의 수를 알지 못한다면 이 방법은 질의어를 적절하게 재구성할 수 없다. 둘째로, 사용자가 검색된 문서의 수를 추정할 수 있다고 하더라도 이 방법은 커다란 문서 집합의 경우 부적절한 질의어를 만들어낼 수 있다.

본 논문에서는 Salton의 DNF 방법의 문제점을 극복할 수 있는 적합성 피드백 방법을 제안한다. 이 방법에서 모든 불리언 질의어는 논리합 정규형(disjunctive normal form)으로 표현될 수 있고 단어들이 AND 연산자로 결합된 각 논리곱은 서로 다른 개념을 나타낸다고 가정한다. 이러한 가정 하에서 검색된 적합한 문서들을 서로 다른 그룹으로 클러스터링을 수행함으로써 개념들을 얻고 이렇게 얻어진 개념들을 OR 연산자로 연결함으로써 질의어를 재구성할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 간단하게 P-norm 기반 확장 불리언 모델과 이 모델을 위한 적합성 피드백 방법을 설명하고 3장에서는 계층적 클러스터링 기법을 이용하여 질의어를 재구성하는 방법을 제안한다. 4장에서는 데이타 집합 TREC 1의 DOE와 Web TREC 10을 이용하여 제안한 방법의 타당성을 살펴보고 5장에서 결론과 향후 연구에 대하여 언급한다.

2. 관련연구

먼저 P-norm 기반 확장 불리언 모델을 설명한다[5]. 이 모델은 어떤 다른 확장 불리언 모델보다 좋은 검색 성능을 얻는데 적합하다는 것이 널리 알려져 있다[8]. 다음으로 Salton 등이 확장 불리언 모델을 위해 제안한 적합성 피드백 방법인 DNF 방법과 이 방법의 문제점에 대하여 간단하게 설명하도록 한다[9].

2.1 P-Norm 기반 확장 불리언 모델

질의어나 문서를 표현할 때 단순 불리언 모델[1]에서는 AND/OR 연산자와 가중치가 없는 용어를 사용하여 표시한다. 그러나, 이러한 질의어는 가중치를 사용하지 못하므로 그 표현력이 약해 정확한 검색을 꾀할 수 없다. 이러한 문제를 해결하기 위하여 많은 확장 불리언 모델이 연구되었고[2,10-12] 그 중에서 본 논문은 Salton과 동료들이 제안한 P-norm을 이용한 확장 불리언 모델[11]을 사용한다.

P-norm 확장 불리언 모델에서 AND/OR 논리 연산자는 가중치를 갖는 문서와 질의어간의 유사도를 잘 정의하고 있다. 만약 어떤 시스템에서 사용되는 용어들을 t_1, t_2, ..., t_n라 하고, 문서 D에 있는 용어 t_i에 대한 가중치를 a_i, $1 \le i \le n$고 $0 \le a_i \le 1$,로 표시하면 문서 D는 $(a_1, a_2, \ldots, a_n)$으로 나타낼 수 있겠다. 또 OR 질의어와 AND 질의어를 $Q_{OR(p)} = OR^p(q_1, q_2, ..., q_n)$와 $Q_{AND(p)} = AND^p(q_1, q_2, ..., q_n)$의 형태로 각각 주어진다고 하자, 단 q_i는, $0 \le q_i \le 1$, 질의어 안에 있는 용어 t_i의 가중치이고 $1 \le p \le \infty$이다. 그러면, P-norm 확장 불리언 모델에서 문서 D와 이 두 질의어 사이의 유사도는 다음과 같이 정의된다.

$$\mathrm{sim}(D, Q_{OR(p)}) = \left[\frac{q_1^p a_1^p + q_2^p a_2^p + \ldots + q_n^p a_n^p}{q_1^p + q_2^p + \ldots + q_n^p} \right]^{1/p} \quad (1)$$

$$\mathrm{sim}(D, Q_{ANDp)}) = 1 - \left[\frac{q_1^p(1-a_1^p) + q_2^p(1-a_2^p) + \ldots + q_n^p(1-a_n^p)}{q_1^p + q_2^p + \ldots + q_n^p} \right]^{1/p} \quad (2)$$

이 모델에서, p = 1이면, 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned} sim(D, Q_{AND(1)}) &= 1 - \left[\frac{q_1(1-a_1) + q_2(1-a_2) + \ldots..+ q_n(1-a_n)}{q_1 + q_2 + \ldots..+ q_n} \right] \\ &= 1 - \left[\frac{(q_1 + q_2 + \ldots..+ q_n) - (q_1a_1 + q_2a_2 + \ldots..+ q_na_n)}{q_1 + q_2 + \ldots..+ q_n} \right] \\ &= \frac{q_1a_1 + q_2a_2 + \ldots..+ q_na_n}{q_1 + q_2 + \ldots..+ q_n} = sim(D, Q_{OR(1)}) \end{aligned}$$

이 경우 문서와 두 질의어(OR 질의어, AND 질의어) 사이의 유사도는 동일하고 그 값은 문서의 용어 가중치와 질의어의 용어 가중치의 내적(inner product)이 된다. 이것은 단순 벡터 모델이 얻어진다는 것을 의미한다. 반면에 p = ∞이고 질의어 용어 가중치가 모두 1이면 아래와 같은 식을 얻을 수 있다.

$$\begin{aligned} sim(D, Q_{AND(\infty)}) &= \lim_{p \to \infty} \; 1 - \left[\frac{q_1^p(1-a_1)^p + q_2^p(1-a_2)^p + \ldots..+ q_n^p(1-a_n)^p}{q_1^p + q_2^p + \ldots..+ q_n^p} \right]^{1/p} \\ &= 1 - \left[\frac{\max[(1-a_1), (1-a_2), \ldots.., (1-a_n)]}{\max[1,1,\ldots,1]} \right] \\ &= 1 - \max[(1-a_1), (1-a_2), \ldots.., (1-a_n)] = \min[a_1, a_2, \ldots.., a_n] \end{aligned}$$

$$\begin{aligned} sim(D, Q_{OR(\infty)}) &= \lim_{p \to \infty} \; \left[\frac{q_1^p a_1^p + q_2^p a_2^p + \ldots..+ q_n^p a_n^p}{q_1^p + q_2^p + \ldots..+ q_n^p} \right]^{1/p} \\ &= \frac{\max[a_1, a_2, \ldots., a_n]}{\max[1,1,\ldots,1]} = \max[a_1, a_2, \ldots., a_n] \end{aligned}$$

이 경우 $\mathrm{sim}(D, Q_{OR(p)}) = \max(a_1, a_2, \ldots, a_n)$와 $\mathrm{sim}(D, Q_{AND(p)}) = \min(a_1, a_2, \ldots, a_n)$이 되어 일반적인 불리언 모델이 된다. 위의 두 가지 경우에서 보는 것처럼 P-norm 확장 불리언 모델은 p의 값을 조정하여 단순 벡터 모델 (p = 1)과 일반적인 불리언 모델(p = ∞)의 중간 형태를 취할 수 있다.

2.2 확장 불리언 모델을 위한 적합성 피드백 방법

불리언 질의어의 적합성 피드백을 위한 적합한 방법을 찾기 위해 많은 연구가 있었다[9,13-15]. 이들 연구

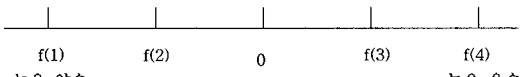
에서 대부분의 불리언 질의어의 적합성 피드백 방법은 크게 두 과정으로 나누어진다. 첫 번째 과정은 확장된 질의어에 사용될 용어를 구하는 것으로 적합성 피드백 방법에 따라 달라지지만 사용자 피드백 문서에 존재하는 용어들 중 중요한 용어를 선택하는 것이 일반적이다. 두 번째 과정은 첫 번째 과정에서 구한 용어들을 적당한 AND/OR 연산자를 이용하여 확장된 질의어를 구하는데 일반적으로 논리합 정규형의 질의어를 생성한다. 이 절에서는 Dillon이 제안한 방법[14]과 Salton등이 Dillon 방법을 개선하여 제안한 DNF(disjunctive normal form) 방법[9]을 통하여 기존에 연구된 불리언 질의어의 적합성 피드백 방법을 살펴본다.

2.2.1 Dillon 방법

Dillon 방법은 질의어 확장에 사용될 용어를 구하는 부분과 구한 용어를 이용하여 질의어를 만드는 부분으로 나누어진다[14]. 질의어에 사용될 용어는 질의어에 존재하는 용어를 배제하고 검새된 문서에 존재하는 용어들 중 적합한 문서를 식별하는 데 유용한 정도에 따라 선택된다. Dillon은 용어의 유용한 정도를 측정하기 위해 다음과 같은 용어 가중치($prev$)를 사용하였다.

$$prev_t = \frac{r_t/R - i_t/I}{\log freq_t}$$

이 식에서 r_t는 용어 t를 포함한 검색된 적합한 문서 수, R은 검색된 적합한 문서 수, i_t 는 용어 t를 포함한 검색된 부적합한 문서 수, I는 검색된 부적합한 문서 수, $freq_t$는 전체 문서 집합에서 용어 t를 포함한 문서 수를 의미한다. 두 번째 부분에서 불리언 연산자 AND, OR, NOT와 위 식을 이용하여 구한 가중치($prev$)를 갖는 용어들을 이용하여 새로운 질의어를 생성한다. 질의어를 생성하기위해 우선 용어들을 아래와 같이 일련의 가중치 플로어(f)에 따라 여러 영역으로 나눈다. 이론적으로 플로어의 수는 제한될 필요는 없다.



이렇게 나누어진 각 영역으로부터 다음과 같이 확장된 질의어를 구한다. 첫 번째 영역인 플로어 f(1)의 좌측에 존재하는 용어들로부터 (T(1) OR T(2) OR ... T(N)) 서브 질의어(G1)를 구한다. 두 번째 영역인 f(1)과 f(2) 사이에 존재하는 용어들로부터 (T(1) AND T(2)) 형태의 모든 것을 OR 연산자로 연결한 서브 질의어(G2)를 구한다. 세 번째 영역인 f(2)와 f(3) 사이에 존재하는 용어들은 서브 질의어 생성에서 제외된다. 네 번째 영역인 f(3)와 f(4) 사이에 존재하는 용어들로부터 NOT(T(1) AND (T(2)) 형태의 모든 것을 AND 연산자로 연결한 서브 질의어(G3)를 구한다. 다섯 번째 영역인 플로어 f(4)의 우측에 존재하는 용어들로부터 (NOTT(1) AND NOTT(2) AND ... AND NOTT(N)) 서브 질의어(G4)를 구한다. 이들 서브 질의어들 중 양수 가중치 값을 갖는 영역에서 구한 G1과 G2를 OR 연산자로 연결하여 (G1 OR G2)를 구하고 음수 가중치 값을 갖는 영역에서 구한 G3와 G4를 AND 연산자로 연결하여 (G3 AND G4)를 구하고 이 둘을 다시 AND 연산자로 연결하여 최종 확장된 질의어 (G1 OR G2) AND (G3 AND G4)를 구한다.

2.2.2 Salton의 DNF(Disjunctive Normal Form) 방법

Dillon 방법은 다음과 같은 특징을 갖는다[9]. 첫째 초기 질의어에 포함된 용어들은 자동으로 확장된 질의어에 사용되지 않는다. 둘째 용어의 중요도를 구하는 식에서 검색된 부적합한 문서를 이용함으로써 G3와 G4처럼 NOT 연산자를 포함하는 질의어를 얻을 수 있는데 이것은 정보 검색에서 NOT 연산자 고유의 문제에 직면할 수 있다. 셋째 단순히 용어의 중요도를 나타내는 가중치($prev$)와 플로어를 이용하여 서브 질의어를 구하므로 확장된 질의어에 너무 많은 절이 포함되어 검색 성능을 제어하기 어렵다. 넷째 자동으로 효과적인 플로어 값을 구하기 어렵다.

Salton은 이러한 성격을 갖는 Dillon 방법을 개선하여 P-norm 기반 확장 불리언 모델을 위한 적합성 피드백 방법인 DNF(disjunctive normal form) 방법을 제안하였다[9]. DNF 방법은 두개의 과정으로 구성된다. 첫 번째 과정은 초기 질의어와 검색된 적합한 문서에 존재하는 용어들로부터 좋은(good) 용어 절을 만드는 과정이다. 이 때 절은 하나의 용어 또는 두개 이상의 용어들이 AND 연산자(∧)로 연결되어 있는 것을 의미한다. 좋은 용어 절을 구하기 위해 Dillon 방법의 용어의 중요도를 나타내는 가중치($prev$)와 유사하게 시스템으로부터 적합한 문서를 검색하는데 유용한 정도를 나타내는 적합성 가중치를 이용한다. 임의의 절 c의 적합성 가중치는 다음 식을 이용하여 구한다.

$$relwt_c = \left[\frac{r_c}{R + qcount} - \frac{freq_c}{N} \right] \ln \left[\frac{N}{freq_c + 10} \right] \quad (3)$$

이 식에서 r_c는 절 c를 포함한 검색된 적합한 문서 수로 만약 절 c를 구성하는 용어가 초기 질의어에 존재할 경우 $qcount$ 만큼 더하여 질의에 존재하지 않는 용어보다 적합한 문서에 나타나는 빈도수를 증가시킨다. R은 검색된 적합한 문서 수, N는 검색 시스템에 있는 전체 문서 수, $freq_c$는 절 c로 검색이 기대되는 문서의 수로 기대 포스팅 빈도라 부른다. 이때 절 c가 하나의 용어 t인 경우 기대 포스팅 빈도는 전체 문서 집합에서 용어 t를 포함한 문서 수가된다. 하지만 절 c가 용어 s, t의 쌍인 s∧t (st로 표기) 또는 용어 r, s, t의 트리플인

r∧s∧t (rst로 표기)인 경우 절 c의 기대 포스팅 빈도수를 구하는 것은 용어 하나를 포함한 절의 빈도수를 구하는 것에 비교하여 어렵기 때문에 대략적으로 st, rst의 기대 포스팅 빈도수는 각각 n_s*n_t/N, $n_r*n_s*n_t/N^2$로 구한다. 단, n_r, n_s, n_t는 각각 전체 문서 집합에서 용어 r, s, t를 포함하는 문서의 수이다.

첫 번째 과정은 우선 초기 질의어와 검색된 적합한 문서에 존재하는 용어의 적합성 가중치를 식 (3)을 이용하여 구하고 적합성 가중치가 높은 상위 k개를 선택한다. 다음으로 이렇게 구한 k개의 용어들의 모든 가능한 쌍의 적합성 가중치를 구하고 적합성 가중치가 높은 상위 m개의 쌍을 선택한다. 마지막으로 앞에서 구한 k개의 용어와 m개의 쌍으로부터 얻어지는 트리플의 적합성 가중치를 구하고 가중치가 높은 상위 n개의 트리플을 선택한다.

두 번째 과정은 다음과 같이 첫 번째 과정에서 구한 좋은 절들을 적절하게 모아서 논리합 정규형의 확장 불리언 질의어를 생성하는 것이다. 먼저 사용자로부터 재구성된 질의어로 검색될 대략적인 문서의 수(T)를 입력받는다. 다음으로 질의어로 선택된 절들로 구해질 것으로 기대되는 문서 수가 T에 근접하도록 첫 번째 과정에서 구한 k개의 용어, m개의 쌍, n개의 트리플 절들 중에서 높은 적합한 절들을 선택한다. 이렇게 선택된 절들을 OR 연산자로 연결하여 논리합 정규형의 확장 불리언 질의어를 구한다. 이렇게 T를 이용하여 Dillon 방법의 특징 중 질의어에 포함되는 절의 수를 제어할 수 있다.

그러나, 이 방법은 사용자가 새로운 질의어의 검색될 문서의 수를 추정해야만 하는 심각한 가정을 한다. 이러한 가정은 두 가지 문제를 초래한다. 첫째로 사용자가 검색될 문서의 수를 알지 못한다면 이 방법은 적절하게 재구성된 질의어를 만들 수 없다. 둘째로 심지어 사용자가 검색될 문서의 수를 알 수 있어도 다음 세 가지 경우를 생각할 수 있다. 1) T의 값이 작다면 용어의 기대 포스팅 빈도 대부분은 T보다 커서 중요한 절들이 선택되기 힘들다. 2) T가 크다면 용어의 기대 포스팅 빈도 대부분이 T보다 작기 때문에 중요하지 않은 절들이 선택될 수 있다. 3) T가 적당한 크기라도 검색 결과는 컬렉션의 크기에 민감할 수 있다. 컬렉션의 크기가 큰 경우 T가 컬렉션의 크기에 상대적으로 작기 때문에 1)과 같은 현상이 나타날 수 있다. 반면에 컬렉션의 크기가 작다면 2)와 같은 결과가 나올 수 있다. 4절의 실험을 통하여 이러한 현상을 살펴보기로 한다.

3. 계층적인 클러스터링 적합성 피드백

이 절에선 계층적인 클러스터링 기법에 기반한 확장 불리언 모델의 새로운 적합성 피드백 방법을 제안한다. 제안하는 방법의 아이디어는 다음과 같다. 확장 불리언 정보검색 모델에서 문서들은 가중치를 갖는 용어들의 리스트로 표현되고 질의어는 용어들의 논리곱(Conjunction)의 논리합(Disjunction)으로 표현될 수 있다. 만약 논리곱이 서로 다른 개념들을 나타낸다고 가정하면 아래의 아이디어를 기반으로 검색된 적합한 문서들로부터 개념들을 구하여 불리언 질의어를 확장하는 것을 생각할 수 있다.

첫째, 질의어 q가 하나의 논리합으로 구성되어 있을 때 질의어 q와 두 문서 D_i와 D_j의 관계를 다이어그램으로 나타내면 아래와 같이 여러 가지 경우를 생각할 수 있다. 1)번 다이어그램은 질의어 q에 대하여 두 문서 D_i와 D_j가 유사한 것을 나타내는 반면에 다른 다이어그램들은 두 문서가 질의어 q에 대하여 유사하다는 것을 나타내지 못한다. 마찬가지로 두 문서 D_i와 D_j가 서로 공유하는 용어가 존재할 때, 두 문서를 유사하게 하는 질의어로 다이어그램 1), 3), 5) 중 1)번 다이어그램과 같이 두 문서가 공통적으로 갖는 용어들의 논리곱을 고려하는 것은 합당할 것이다.

둘째, 적합한 문서들에 대하여 클러스터링을 수행함으로써 적합한 문서들 중 다른 문서들과 비교하여 서로 좀 더 비슷한 문서들의 그룹을 만들 수 있다. 그리고 위에서 언급한 것과 같이 각 그룹에 존재하는 문서들을 유사하게 하는 질의어(용어들의 논리합)을 구하면 이러한 질의어는 사용자의 정보 요구(Information Need)를 반영하는 것을 기대할 수 있다.

셋째, 둘째에서 구한 각 그룹에 대응하는 질의어를 논리합 연산자로 결합함으로써 사용자의 정보 요구를 표현하는 논리합 정규형으로 표현된 확장된 질의어를 구할 수 있다.

이러한 아이디어를 바탕으로 제안하는 방법은 두개의 과정으로 이루어진다. 첫 번째 과정은 검색된 적합한 문서들에 존재하는 개념을 찾는 과정으로 본 논문에선 계층적인 클러스터링 방법을 이용한다. 이 과정은 트리 구

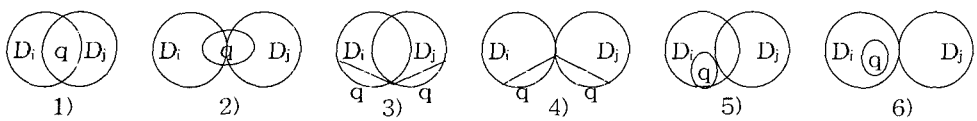


그림 1 문서 D_i, D_j와 질의어 q사이의 관계 다이어그램

조(dendrogram)와 같은 문서 클러스터를 생성하는데 우리는 이것을 클러스터 트리라고 부른다. 두 번째 과정은 생성된 클러스터 트리로부터 논리합 정규형의 확장 불리언 질의어를 생성하는 것이다. 아래에서 각 과정에 대하여 자세하게 살펴보도록 한다.

3.1 클러스터 트리 만들기

클러스터 트리를 생성하기 위해 분할 계층적인 클러스터링(divisive hierarchical clustering)을 수행한다. 이 클러스터링은 우선 모든 적합한 검색된 문서를 하나의 그룹에 넣고 그룹을 두개의 서브 그룹으로 나누는 작업을 반복 수행한다. 일반적으로 계층적인 클러스터링 방법은 클러스터링 과정을 멈출 때를 알아야하는데[16] 우리는 클러스터 트리의 깊이와 트리에 존재하는 노드에 포함될 최소 문서 수를 이용하여 하나의 그룹을 두개의 서브 그룹으로 나누는 작업을 제어한다. 클러스터 트리의 최대 깊이를 크게 설정하면 최종적으로 많은 수의 클러스터를 얻을 수 있다. 반면에 노드에 포함될 최소 문서 수를 크게 설정하면 적은 수의 클러스터를 얻을 수 있다. 또한 계층적인 클러스터링 방법은 하나의 그룹을 두개의 서브 그룹으로 나누는 방법이 필요하다. 일반적으로 분할 계층적인 클러스터링 방법은 그룹 안에 존재하는 모든 객체들 사이의 거리(distance)를 구하여 가장 멀리 떨어져 있는 두 객체를 서브 그룹의 씨드(seed)로 설정하고 나머지 객체들을 가장 가까운 서브 그룹에 배정하는 방법을 사용한다[16]. 하지만 우리는 첫째 아이디어로부터 최종적으로 생성된 그룹에 존재하는 개념을 구하기 위해 그룹을 대표할 수 있는 가장 유력한 용어를 선택하고 이 용어를 사용하여 두개의 서브 그룹을 구한다. 하나의 서브 그룹은 상위 그룹을 대표하는 용어를 포함하는 문서들로 구성되고 다른 서브 그룹은 이 용어를 포함하지 않는 문서들로 구성된다. 지금까지 설명한 방법을 정리하면 그림 2 클러스터 트리를 만드는 프로시저와 같다.

예를 들어 초기 검색된 적합한 문서 집합이 주어졌을 때 우선 이 전체 문서를 포함하는 루트 노드를 생성하여 스택에 넣는다. 스택에서 하나의 노드를 꺼내서 이 노드를 두 개의 서브 그룹으로 나눌 수 있는지 테스트한다. 이것은 앞에서 언급한 것처럼 현재 노드의 깊이가 클러스터 트리의 최대 깊이보다 작고 이 노드에 포함된 문서의 수가 클러스터 트리의 최소 문서 수 이상이면 두 개의 서브 그룹으로 나눌 수 있고 그렇지 않은 경우 나눌 수 없다. 테스트 결과 두 개의 서브 그룹으로 나눌 수 있다면 이 노드에 존재하는 문서들을 대표하는 용어를 선택한다. 용어를 선택하는 방법은 아래에 자세하게 설명하겠다. 선택된 용어가 t_l이라고 가정하면 t_l을 포함하는 문서들을 왼쪽 자식 노드에 배정하고 t_l을 포함하지 않는 문서들을 오른쪽 자식 노드에 배정한다. 이렇게 생성된 각 자식 노드를 스택에 넣는다. 이와 같은 작업을 스택이 빌 때까지 반복 수행하면 그림 3과 같은 클러스터 트리를 얻을 수 있다.

각 노드에 존재하는 용어 t_i는 노드를 대표하는 용어이다. 그룹을 대표할 수 있는 가장 유력한 용어를 선택하기 위해 우리는 적합성 피드백 분야에서 잘 알려진 용어 선택 방법인 Salton 방법, F4MODIFIED 방법, Porter 방법을 사용한다[17]. Salton 방법은 DNF 방법에서 적합성 가중치를 계산하기 위해 사용한 식 (3)을 이용하여 용어를 선택하는 것이고 다른 용어 선택 방법들은 다음과 같다.

```
1. 검색된 적합한 문서가 루트 노드에 들어있는 클러스터 트리를 만든다.
2. 루트 노드를 Node_Stack 스택에 넣는다.
3. while(Node_Stack 스택이 비어있지 않는 동안)
    i. Node_Stack에서 노드 하나를 꺼내고 이를 Current_Node라 한다.
    ii. if(Current_Node가 두 개의 서브 그룹으로 나누어 질 수 있다면)
        A. Current_Node를 대표할 수 있는 용어를 찾는다(Selected_Term).
        B. Current_Node의 왼쪽 자식 노드와 오른쪽 자식 노드를 만든다.
        C. Current_Node에 존재하는 문서를 Selected_Term을 포함하는 문서 집합(In_D)과
           포함하지 않는 문서 집합(Ex_D)으로 나눈다.
        D. In_D에 속하는 문서를 왼쪽 자식 노드에 할당하고
           Ex_D에 속한 문서를 오른쪽 자식 노드에 할당한다.
        E. 두 자식 노드를 Node_Stack 스택에 넣는다.
    iii. endif
4. endwhile
```

그림 2 클러스터 트리를 만드는 프로시저

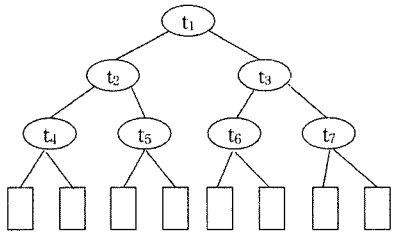


그림 3 클러스터 트리 예

F4MODIFIED 방법은 Robertson이 F4point-5 식을 수정한 것이다[18,19]. 이 방법은 오리지널 질의어에 새로운 용어를 추가하기 위해 사용된 것으로 다음과 같다.

$$w_t = \log \frac{(r+c)(N-n+R+1-c)}{(n-4+c)(R-r+1-c)}$$

이 식에서 $c = n/N$, N은 컬렉션에 있는 전체 문서 수이고, R은 사용자 피드백으로 주어진 적합한 문서 수이고, n은 용어 t를 갖는 문서 수이고, r은 용어 t를 갖는 사용자로부터 주어진 적합한 문서 수이다.

Porter 방법은 Porter and Galpin이 제안한 질의어 확장을 위한 용어의 순위를 구하는 알고리즘으로 다음과 같다[20].

$$w_t = \frac{r}{R} - \frac{n}{N}$$

여기에서 r, R, n, N은 각각 F4MODIFIED 방법에서 사용된 것과 동일하다.

3.2 DNF(Disjunctive Normal Form) 형태의 확장 불리언 질의어 생성하기

3.1에 설명한 방법으로 만들어진 클러스터 트리로부터 DNF 형태의 질의어를 생성한다. 첫째 아이디어를 이용하여 각 그룹에 대응하는 개념을 구하기 위해선 생성된 그룹이 정의되어야 한다. 그림 3과 같이 클러스터 트리가 생성되었다고 가정하자. 그림 3의 클러스터 트리에 존재하는 모든 단말 노드를 공통적으로 포함된 용어들의 논리곱으로 좌측에 있는 것부터 표현하면 아래와 같다.

$T_1 = t_1 \wedge t_2 \wedge t_4$

$T_2 = t_1 \wedge t_2$

$T_3 = t_1 \wedge t_5$

$T_4 = t_1$

$T_5 = t_3 \wedge t_6$

$T_6 = t_3$

$T_7 = t_7$

$T_8 =$

단말 노드 T_1과 T_2에서 T_2의 경우 T_1보다 좀 더 넓은 의미를 가지는 것을 볼 수 있다. 하지만 클러스터 트리를 생성할 때 T_1과 T_2의 부모 노드를 표현하는 대표 용어 t_4는 T_1이 T_2보다 더 중요한 개념을 갖는다는 것을 암묵적으로 내포하고 있다고 생각할 수 있다. 또한 P-norm 확장 불리언 모델의 유사도를 구하는 식 (2)를 살펴보면 T_1에 위해 T_2의 개념을 포함한 문서들을 검색할 수 있다. 나머지 단말 노드에 대하여 동일한 생각을 할 수 있고 본 논문에선 단말 노드의 부모 노드에 존재하는 문서 집합을 하나의 그룹으로 정의한다.

이렇게 정의된 클러스터는 트리의 루트로부터 두개의 말단 노드의 부모까지의 패스에 존재하는 용어들의 논리곱으로 표현할 수 있다. 그림 3의 클러스터 트리의 경우 그림 4와 같이 4 개의 클러스터가 존재하고 각 클러스터에 들어있는 개념을 용어들의 논리곱으로 표현하면 다음과 같다.

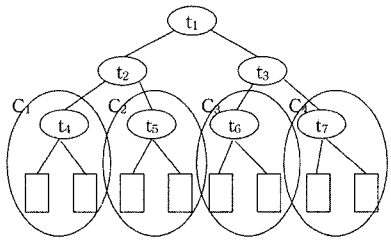


그림 4 클러스터 트리에 존재하는 클러스터

$C_1 = t_1 \wedge t_2 \wedge t_4$

$C_2 = t_1 \wedge t_5$

$C_3 = t_3 \wedge t_6$

$C_4 = t_7$

이렇게 구한 클러스터들의 개념을 이용하여 아래와 같은 확장 불리언 질의어를 구할 수 있다.

$$query = C_1 \vee C_2 \vee C_3 \vee C_4$$

확장 불리언 질의어를 생성하는 프로시저는 그림 5와 같이 요약할 수 있다.

```
1. for(클러스터 트리의 모든 단말 노드에 대하여)
    I. if(주어진 단말 노드가 부모 노드의 왼쪽 자식 노드면)
        A. 이 노드를 표현하는 용어 리스트(Term_Clause)를 구한다.
        B. Term_Clause를 Or_Clause에 추가한다.
    II. endif
2. endfor

3. Or_Clause에 있는 모든 Term_Clause를 OR 연산자로 연결하여 확장 질의어를 생성한다.
```

그림 5 확장 불리언 질의어 생성하는 프로지저

4. 실 험

본 연구에서 두 가지 실험을 수행하였다. 첫 번째 실험은 Salton의 DNF 방법의 문제점을 밝히기 위해 수행되었고 두 번째 실험은 제안한 확장 불리언 모델을 위한 계층적인 클러스터링 기법을 이용한 적합성 피드백 방법을 평가하기 위해 수행되었다. 이러한 실험을 위하여 두 개의 데이타 컬렉션이 사용되었는데 하나는 TREC 1에 있는 DOE(Department of Energy) 컬렉션이고 다른 하나는 Web TREC 10 컬렉션이다. DOE 컬렉션은 비교적 작은 데이타 집합으로 약 220,000 개의 문서를 포함하고 있다. TREC에서 제공한 65, 66, 68, 75, 82, 96, 111, 123, 134, 135 주제 10개를 가지고 DOE 컬렉션에서 실험을 수행하였다. 정확한 실험 결과를 얻기 위해 많은 수의 주제가 사용되어야 하는데 본 실험에서 사용한 DOE 컬렉션과 관련이 있는 주제의 수가 적어 위와 같이 10개 주제만을 사용하였다. 반면에 DOE 컬렉션과 비교하여 Web TREC 10 컬렉션은 비교적 커다란 데이타 집합이고 1,600,000 이상의 문서를 포함하고 있다. 이 데이타 컬렉션에서 실험을 수행하기 위해 TREC에서 제공하는 501~550 주제 50개를 이용하였다.

주제에 대한 초기 질의어는 데이타 컬렉션에 따라 수작업으로 또는 자동으로 만들어졌다. DOE 컬렉션의 주제들을 위한 초기 질의어는 각 주제의 설명 섹션을 보고 적당한 용어들과 불리언 연산자를 선택하여 초기 질의어를 수작업으로 만들었다. 반면에 Web TREC 10 컬렉션의 주제들에 대해서는 각 주제의 타이틀 섹션에 존재하는 모든 용어들을 논리곱 연산자로 연결하여 초기 질의어를 만들었다. 이 데이타 컬렉션에서 초기 질의어를 자동으로 만든 이유는 타이틀에 존재하는 용어의 수가 제한적이기 때문이다. 또한 질의어 용어의 가중치는 역문서 빈도수(Inverse Document Frequency)를 이용하여 구하였다. 예를 들어 다음과 같은 주제 65와 주제 501에 대하여 생성된 초기 질의어와 확장된 질의어를 살펴보면 아래와 같다.

주제 65:

<head> Tipster Topic Description
<num> Number: 065
<dom> Domain: Science and Technology
<title> Topic: Information Retrieval Systems
<desc> Description:
Document will identify a type of information retrieval system.
<smry> Summary:
A relevant document will identify a new information retrieval system, identify the company or person marketing the system, and identify some of the characteristics of the system.
<narr> Narrative:
A relevant document will identify an information retrieval system, identify the company or person marketing the system, and identify some of the characteristics of the system.
<con> Concept(s):
1. information retrieval system
2. storage, database, data, query
<fac> Factor(s):
<def> Definition(s):

주제 501:

<num> Number: 501
<title> deduction and induction in English?
<desc> Description:
What is the difference between deduction and induction in the process of reasoning?
<narr> Narrative:
A relevant document will contrast inductive and deductive reasoning.
A document that discusses only one or the other is not relevant.

주제 65로부터 수작업으로 작성된 "Information AND Retrieval"에 어근 추출(stemming) 작업과 역문서 빈도수를 적용하여 확장 불리언 모델의 초기 질의어를 다음과 같이 만들었고

inform(0.2343) AND retriev(0.4941)

이렇게 구한 초기 질의어를 Salton의 DNF 방법과 본 논문에서 제안한 방법을 이용하여 구한 확장된 질의어는 다음과 같다.

Salton의 DNF 방법으로 확장된 질의어:
inform(0.2194) OR retriev(0.7802) OR (inform (0.2343) AND retriev(0.4941))
계층적인 클러스터링 기법을 이용하여 확장된 질의어:
(retriev(1.0000) AND inform(0.2812) AND databas (0.2385)) OR
(retriev(1.0000) AND inform(0.2812) AND queri (0.2071)) OR
(retriev(1.0000) AND databas(0.2385)) OR
(retriev(1.0000) AND criteria(0.0818)) OR
(inform(0.2343) AND retriev(0.4941))

비슷하게 주제 501로부터 자동으로 작성된 "deduction AND induction AND english"의 초기 질의어와 확장된 질의어는 다음과 같다.

초기 질의어:
english(0.2243) AND induct(0.4089) AND deduct (0.3442)

Salton의 DNF 방법으로 확장된 질의어:
induct(1.0000) OR (english(0.2243) AND induct (0.4089) AND deduct(0.3442))

계층적인 클러스터링 기법을 이용하여 확장된 질의어:
(induct(1.0000) AND deduct(0.2947) AND logic (0.0735) AND peirc(0.0616)) OR
(induct(1.0000) AND orbitrecord(0.1538) AND halodust(0.1137)) OR
(induct(1.0000) AND gene(0.1563)) OR
(deduct(0.2947) AND exempt(0.0258)) OR
(english(0.2243) AND induct(0.4089) AND deduct (0.3442))

위에서 언급한 데이타 집합과 주제를 이용하여 실험에 대한 검색 성능 평가를 위해 고정된 재현율(recall) 레벨(0.1 단계로 0.1부터 1.0까지)에서 모든 주제에 대하여 평균 정확율(precision)을 구하는 재현율-정확율 평균(recall-precision average) 방법을 사용한다. 이 방법은 주로 TREC 컨퍼런스에서 사용되는 방법이다. 두 실험의 과정과 결과는 다음과 같다.

4.1 실험1

Salton의 DNF 방법은 재구성된 질의어로 검색될 문서 수를 사용자가 추정해야만 하는 심각한 가정을 가지고 있다. Salton과 동료들은 검색될 문서 수를 정확히 알 수 없기 때문에 추정 문서 수를 실험적으로 결정하였다. 이러한 사실은 잠재적으로 DNF 방법이 검색될 문서 수의 추정치에 상당히 민감하게 영향을 받는다는 것을 알려준다. 본 실험에서는 DNF 방법의 문제점을 밝히기 위해 다양한 추정된 검색될 문서 수를 두 데이타 집합에 적용하여 수행하고 결과를 살펴본다.

실험을 수행하기 위해 다음과 같은 매개변수들을 사용한다. *P-Value*는 P-norm 확장 불리언 모델에의 P 값으로 실험을 통하여 2.0에서 다른 값들보다 좋은 결과를 보였다. 사용자의 적합성 피드백 문서수(*Relevance feedback*)는 검색된 상위 100개 문서로 제한하였다. *qcount*는 2.2.2절에 설명한 것과 같이 질의어에 포함된 용어에 대하여 포함되지 않은 용어보다 적합한 문서에 출현한 빈도수를 증가시기 위해 사용되는 값으로 Salton이 실험에 이용한 것과 동일한 값인 2를 사용하였다. Salton의 DNF 방법에서 사용되는 추정된 검색될 문서 수(T)는 아래와 같이 다양한 값으로 실험을 수행하였다.

P-Value : 2.0
Relevance feedback : 상위 100 개 문서
qcount : 2
추정된 검색될 문서 수 리스트(T): 100, 500, 1000, 5000, 10000

또한 주어진 주제의 검색될 문서의 정확한 수를 데이타 컬렉션에 존재하는 그 주제의 총 적합한 문서 수라고 가정하고 실험을 수행하였다. DOE 컬렉션의 평균 적합한 문서 수는 120이고 Web TREC 10 컬렉션의 평균 적합한 문서 수는 1,408이다.

실험한 결과는 DOE 컬렉션과 Web TREC 10 컬렉션을 구별하여 아래와 같이 표로 정리하였다. 표에 *T:100, T:500* 등은 문서 추정 수를 100, 500로 하여 질의어를 확장한 경우를 말하고 *CorrectNum*은 각 주제에 대하여 전체 적합한 문서 수로 추정한 수를 말한다.

데이타 집합의 크기가 상대적으로 작은 DOE 컬렉션의 실험 결과를 보여주는 표 1과 2에서 다음과 같은 사실을 관찰할 수 있다.

가) 검색될 문서의 추정 수가 점차적으로 증가되는 경우 용어와 절의 평균수도 또한 증간된다.
나) 검색 효율은 T의 값이 클 때 나빠진다.

표 1 용어와 절의 평균 수 (DOE)

	용어의 평균 수	절의 평균 수
초기질의어	3.20	1.80
T:100	3.90	3.40
T:500	3.90	3.40
T:1000	5.60	5.40
T:5000	11.00	12.20
T:10000	12.20	14.90
T:CorrectNum	3.80	2.30

표 2 재현율-정확율 평균 (DOE)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.3912	0.2690	0.2213	0.1939	0.1745	0.1456	0.1068	0.0493	0.0232	0.0030	0.1578
T:100	0.4204	0.2864	0.2390	0.2087	0.1881	0.1552	0.1188	0.0616	0.0271	0.0030	0.1708
T:500	0.4204	0.2864	0.2390	0.2087	0.1881	0.1552	0.1188	0.0616	0.0271	0.0030	0.1708
T:1000	0.4256	0.3174	0.2420	0.2016	0.1777	0.1388	0.0978	0.0534	0.0249	0.0021	0.1681
T:5000	0.3679	0.2601	0.2108	0.1803	0.1621	0.1197	0.0930	0.0564	0.0261	0.0028	0.1479
T:10000	0.3448	0.2365	0.1971	0.1655	0.1471	0.1142	0.0877	0.0535	0.0198	0.0036	0.1370
T:CorrectNum	0.4145	0.2809	0.2294	0.2003	0.1811	0.1510	0.1125	0.0544	0.0254	0.0030	0.1653

표 3 용어와 절의 평균 수 (Web TREC 10)

	용어의 평균 수	절의 평균 수
초기질의어	2.42	1.00
T:100	2.90	1.48
T:500	3.38	2.42
T:1000	3.40	2.88
T:5000	6.78	7.50
T:10000	9.04	12.14
T:CorrectNum	3.86	4.04

다) 검색될 문서의 올바른 추정치의 경우 검색 성능은 초기 질의어 보다는 좀 더 좋지만 *T:100, T:500*의 경우보다 나쁜 것을 볼 수 있다.

데이타 집합의 크기가 상대적으로 큰 Web TREC 10 컬렉션의 실험 결과를 보여주는 표 3과 4에서 다음과 같은 사실을 관찰할 수 있다.

가) T의 값이 증가함에 따라 용어와 절의 평균수도 또한 늘어난다.

나) 검색 성능은 T의 값이 커질 때 나빠진다.

다) T 값에 관계없이 초기 질의어 보다 검색 성능이 떨어진다.

라) 검색될 문서의 올바른 추정치의 경우 검색 성능은 초기 질의어 보다 나쁜 것을 볼 수 있다.

게다가 T 값이 충분히 작은 경우 질의어 확장이 되지 않고 초기 질의어가 동일한 경우가 많이 발생한다는 것을 실험에서 발견할 수 있었다. 위의 실험 결과로부터 Salton의 DNF 방법의 성능은 검색될 문서의 추정 수에 종속적이라는 것을 알 수 있었다.

4.2 실험2

이 실험에선 계층적인 클러스터링 기법에 기반한 적합성 피드백 방법을 사용하여 질의어를 재구성하고 이렇게 확장된 질의어를 이용하여 검색을 수행한 결과와 DNF 방법의 결과를 비교한다. 이 실험에 이용된 제안한 적합성 피드백 방법과 DNF 방법을 위해 사용한 매개변수들은 다음과 같다. *P-Value*, 적합성 피드백 문서 수(*Relevance feedback*), *qcount*는 실험1에서 설명한 내용과 같고 Salton의 DNF 방법에서 사용되는 추정된 검색될 문서 수(T)는 500으로 사용하였다. 우리가 제안하는 방법에서 클러스터 트리를 생성할 때 클러스터링을 제어하기 위해 클러스터 트리의 최대 깊이는 4, 단말 노드에 존재하는 문서의 최소수는 5로 하였다.

P-Value : 2.0

Relevance feedback : 상위 100 개 문서

qcount : 2

검색될 문서의 추정 수(T) : 500

클러스터 트리의 깊이 : 4

단말 노드에 존재하는 문서의 최소 수 : 5

또한 3.1절에서 설명한 세 개의 용어 선택 방법(Salton, Porter, F4MODIFIED)을 사용하였다. 우리가 제안한 방법과 Salton이 제안한 방법의 차이점을 명백하게 하기 위해 두 데이타 컬렉션의 주제를 두개의 서브 그룹으로 나누었다. 하나는 초기 질의어 결과가 나쁜 그룹으로 DOE 컬렉션의 경우 검색된 상위 100개의 문서에 20개 이하의 적합한 문서를 포함한 주제가 그리고 Web TREC 10 컬렉션의 경우 50개 이하의 적합한 문서를 포함한 주제가 이 그룹에 속한다. 다른 한 그룹은 초기 질의어 결과가 좋은 그룹으로 초기 질의어 결과가 나쁜 그룹에 속하지 않는 나머지 주제로 이루어졌다. 이렇게 두 그룹으로 나눈 이유는 Salton의 DNF 방법과 우리 방법이 초기 검색 결과가 좋은 경우보다 나쁜 경우 질의어 확장이 잘되기 때문에 이렇게 두 그룹으로 나누어 성능 평가를 하면 두 방법의 성능 차이를 쉽게 볼 수 있다.

실험 결과로 다음과 같은 표 5, 6, 7, 8, 9, 10을 얻었다. 이 표의 *DNF*는 Salton의 DNF 방법을 이용하여 질의어를 확장한 경우를 말하고 *HCR1, HCR2, HCR3*는 용어 선택 방법인 Salton 방법, Porter 방법, F4MODIFIED 방법을 이용한 제안한 적합성 피드백 방법으로 질의어를 확장한 경우를 말한다.

먼저 데이타 크기가 상대적으로 작은 DOE 컬렉션의 실험 결과를 재현율 0.1과 평균에서 살펴보자. 초기 질의어 결과가 나쁜 그룹의 경우 초기 질의어와 비교하여 재현율 0.1에서 *DNF*는 22% 향상, HCR1은 51% 향상, HCR2는 124% 향상, HCR3는 78% 향상을 보였고 평균의 경우 *DNF*는 22% 향상, HCR1은 83% 향상,

표 4 재현율-정확율 평균 (Web TREC 10)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.5516	0.4227	0.3085	0.2347	0.1540	0.1134	0.0614	0.0249	0.0063	0.0000	0.1877
T:100	0.5460	0.4128	0.3058	0.2310	0.1528	0.1127	0.0615	0.0248	0.0066	0.0000	0.1854
T:500	0.5336	0.3977	0.2957	0.2237	0.1491	0.1105	0.0606	0.0245	0.0064	0.0000	0.1802
T:1000	0.5125	0.3829	0.2843	0.2130	0.1386	0.0998	0.0602	0.0244	0.0063	0.0000	0.1722
T:5000	0.4087	0.2635	0.1827	0.1288	0.0785	0.0493	0.0299	0.0178	0.0045	0.0000	0.1164
T:10000	0.4116	0.2805	0.1857	0.1261	0.0749	0.0447	0.0274	0.0137	0.0043	0.0000	0.1169
T:CorrectNum	0.5136	0.3736	0.2851	0.2187	0.1455	0.1082	0.0594	0.0240	0.0062	0.0000	0.1734

표 5 재현율-정확율 평균 (DOE의 주제 전체)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.3912	0.2690	0.2213	0.1939	0.1745	0.1456	0.1068	0.0493	0.0232	0.0030	0.1578
DNF	0.4145	0.2809	0.2294	0.2003	0.1811	0.1510	0.1125	0.0544	0.0254	0.0030	0.1653
HCR1	0.4112	0.3711	0.2789	0.2058	0.1878	0.1579	0.1201	0.0853	0.0411	0.0029	0.1862
HCR2	0.5269	0.3983	0.2756	0.2280	0.2139	0.1675	0.1186	0.0744	0.0252	0.0024	0.2031
HCR3	0.4476	0.3109	0.2428	0.1963	0.1813	0.1475	0.1149	0.0678	0.0380	0.0030	0.1750

표 6 재현율-정확율 평균 (DOE의 초기 질의어 결과가 나쁜 그룹)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.1770	0.1101	0.0823	0.0614	0.0558	0.0458	0.0270	0.0086	0.0005	0.0000	0.0569
DNF	0.2159	0.1299	0.0959	0.0719	0.0668	0.0546	0.0365	0.0172	0.0042	0.0000	0.0693
HCR1	0.2667	0.2413	0.1617	0.0944	0.0846	0.0797	0.0556	0.0425	0.0156	0.0000	0.1042
HCR2	0.3964	0.2608	0.1394	0.1113	0.0953	0.0753	0.0609	0.0288	0.0055	0.0000	0.1174
HCR3	0.3147	0.1885	0.1462	0.0815	0.0786	0.0594	0.0427	0.0164	0.0005	0.0000	0.0929

표 7 재현율-정확율 평균 (DOE의 초기 질의어 결과가 좋은 그룹)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.7125	0.5073	0.4298	0.3928	0.3525	0.2954	0.2265	0.1102	0.0573	0.0075	0.3092
DNF	0.7125	0.5073	0.4298	0.3928	0.3525	0.2954	0.2265	0.1102	0.0573	0.0075	0.3092
HCR1	0.6279	0.5659	0.4546	0.3728	0.3427	0.2751	0.2168	0.1496	0.0794	0.0073	0.3092
HCR2	0.7225	0.6046	0.4800	0.4030	0.3917	0.3058	0.2053	0.1429	0.0547	0.0061	0.3316
HCR3	0.6470	0.4946	0.3876	0.3684	0.3353	0.2797	0.2232	0.1449	0.0942	0.0074	0.2982

표 8 재현율-정확율 평균 (Web TREC 10의 주제 전체)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.5516	0.4227	0.3085	0.2347	0.1540	0.1134	0.0614	0.0249	0.0063	0.0000	0.1877
DNF	0.5136	0.3736	0.2851	0.2187	0.1455	0.1082	0.0594	0.0240	0.0062	0.0000	0.1734
HCR1	0.6237	0.4667	0.3570	0.2666	0.1708	0.1042	0.0558	0.0275	0.0071	0.0000	0.2079
HCR2	0.6282	0.4522	0.3158	0.2119	0.1293	0.0786	0.0420	0.0179	0.0051	0.0000	0.1881
HCR3	0.5477	0.4145	0.3003	0.2269	0.1506	0.1112	0.0603	0.0244	0.0063	0.0000	0.1842

표 9 bad 재현율-정확율 평균 (Web TREC 10의 초기 질의어 결과가 나쁜 그룹)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.2191	0.1966	0.1899	0.1856	0.1497	0.1466	0.0959	0.0394	0.0017	0.0000	0.1225
DNF	0.2453	0.2005	0.1903	0.1844	0.1476	0.1443	0.0961	0.0392	0.0017	0.0000	0.1249
HCR1	0.5580	0.4858	0.4108	0.3225	0.1979	0.1279	0.0704	0.0397	0.0015	0.0000	0.2214
HCR2	0.5705	0.4850	0.3437	0.2006	0.1333	0.0806	0.0459	0.0147	0.0021	0.0000	0.1876
HCR3	0.2015	0.1751	0.1698	0.1663	0.1448	0.1421	0.0925	0.0377	0.0017	0.0000	0.1131

표 10 good 재현율-정확율 평균 (Web TREC 10의 초기 질의어 결과가 좋은 그룹)

재현율	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	평균
초기질의어	0.6684	0.5022	0.3501	0.2519	0.1555	0.1018	0.0493	0.0198	0.0080	0.0000	0.2107
DNF	0.6079	0.4345	0.3183	0.2308	0.1448	0.0955	0.0466	0.0186	0.0077	0.0000	0.1905
HCR1	0.6469	0.4600	0.3381	0.2469	0.1613	0.0959	0.0507	0.0233	0.0091	0.0000	0.2032
HCR2	0.6485	0.4406	0.3060	0.2159	0.1279	0.0779	0.0407	0.0191	0.0061	0.0000	0.1883
HCR3	0.6693	0.4986	0.3462	0.2482	0.1526	0.1004	0.0490	0.0198	0.0080	0.0000	0.2092

HCR2는 106% 향상, HCR3는 63% 향상을 보였다. 이것은 제안한 방법이 *DNF* 방법에 비교하여 월등히 높은 성능을 보인다는 것을 보여준다.

반면에 초기 질의어 결과가 좋은 그룹의 경우 재현율 0,1에서 *DNF*는 0% 향상, HCR1은 -12% 향상, HCR2는 1% 향상, HCR3는 -9% 향상을 보였고 평균에서 *DNF*는 0% 향상, HCR1은 0% 향상, HCR2는 7% 향상, HCR3는 -4% 향상을 보였다. *DNF* 방법은 초기

질의어와 동일한 결과를 보였는데 우리가 제안한 방법은 초기 재현율에서 성능 감소를 보였지만 평균적으로 향상된 것을 보여주었다. *DNF* 방법의 경우 확장된 질의어가 초기 질의어와 동일하여 질의어 확장이 제대로 이루어 지지 않는 것을 살펴볼 수 있었다.

다음으로 데이타 크기가 큰 Web TREC 10 컬렉션의 실험 결과를 재현율 0.1과 평균에서 살펴보면 다음과 같다. 초기 질의어 결과가 나쁜 그룹의 경우 초기 질의어와 비교하여 재현율 0.1에서 *DNF*는 12% 향상, HCR1은 155% 향상, HCR2는 160% 향상, HCR3는 -8% 향상을 보였고 평균의 경우 *DNF*는 2% 향상, HCR1은 81% 향상, HCR2는 53% 향상, HCR3는 -8% 향상을 보였다. DOE 컬렉션과 같이 제안한 방법이 *DNF* 방법에 비교하여 월등히 높은 성능을 보인다는 것을 보여준다.

반면에 초기 질의어 결과가 좋은 그룹의 경우 재현율 0,1에서 *DNF*는 -9% 향상, HCR1은 -3% 향상, HCR2는 -3% 향상, HCR3는 0% 향상을 보였고 평균에서 *DNF*는 -10% 향상, HCR1은 -4% 향상, HCR2는 -11% 향상, HCR3는 -1% 향상을 보였다. 모든 방법이 초기 질의어 결과와 비교하여 성능이 떨어지는 것을 볼 수 있었지만 우리가 제안한 방법이 *DNF* 방법보다 좋은 성능을 보였다.

전체 주제에 대한 성능은 초기 질의어 결과가 나쁜 그룹과 좋은 그룹의 평균으로 DOE 컬렉션의 경우 *DNF*는 방법과 우리가 제안한 방법 모두 성능 향상을 보였고 Web TREC 10 컬렉션의 경우 *DNF*는 방법은 성능이 감소한 반면 제안한 방법은 성능이 향상되었다. 결론적으로 우리가 제안한 방법이 Salton의 *DNF* 방법과 비교하여 좋은 성능을 보이는 것을 실험을 통하여 확인하였다. 특히 초기 질의어 결과가 나쁜 그룹에서 초기 질의어와 비교하여 60% 이상의 성능 향상을 보였다. 하지만 초기 질의어 결과가 좋은 그룹에서 제안한 방법으로 확장된 질의어가 초기 질의어 보다 나쁜 결과를 보였고 용어를 선택 방법에 따라 성능의 차이가 발생하는 것도 알 수 있었다.

5. 결 론

본 연구에선 확장 불리언 검색 모델을 위한 Salton이 제안한 DNF 방법에서 사용한 검색될 문서 수의 추정치에 관련된 문제점을 보이고 이 문제점을 해결할 수 있는 계층적 클러스터링 기법을 이용한 적합성 피드백 방법을 제안하였다. 또한 두 방법을 상대적으로 작은 TREC 1의 DOE 컬렉션과 상대적으로 큰 Web TREC 10 컬렉션에서 비교 실험하여 제안한 방법의 우수성을 살펴보았다. 하지만 제안한 방법도 용어를 선택하는 방법에 따라 성능의 차이가 발생한다는 것을 알았다. 또한 이 연구에서 단순히 확장 불리언 검색 모델의 적합성 피드백 방법에서 용어를 선택하고 적절한 불리언 연산자(AND/OR)로 연결하여 질의어를 확장하는 데 집중하였을 뿐 적합성 피드백의 다른 부분인 용어의 가중치를 조정하는 방법에는 소홀 하였다. 추후 연구로 좀 더 좋은 성능을 갖는 용어 선택 방법을 찾고 확장된 질의어에 존재하는 용어의 가중치를 조정하는 방법에 대하여 연구하고 많은 데이타 집합에서 실험을 수행함으로써 뛰어난 성능을 갖는 적합성 피드백 방법을 찾겠다.

참 고 문 헌

[1] Ide, E. New experiments in relevance feedback. In Salton, G., ed., The Smart System - Experiments in Automatic Document Processing, pp. 337-354. Englewood Cliffs, NJ: Prentice-Hall Inc, 1971.

[2] Rocchio, J. J. Jr. Relevance feedback in information retrieval. In Salton, G., ed., The Smart System - Experiments in Automatic Document Processing, pp. 313-323. Englewood Cliffs, NJ: Prentice-Hall Inc, 1971.

[3] Salton, G. and Buckely, C. Improving retrieval performance by relevance feedback. J. of the American Society for Information Science, 41(4): pp. 288-297, 1990.

[4] Bookstein, A. Fuzzy requests: An approach to weighted Boolean searches, J. ASIS, Vol 31, No. 4, July, 1980, pp. 275-279.

[5] Salton, G., Fox, E. A., and Wu, H. Extended Boolean information retrieval, Vol. 36, No. 11, December 1983, Communication of the ACM, pp. 1022-1036.

[6] Waller, W. G. and Kraft, D. H. A mathematical model for a weighted Boolean retrieval system. Information Processing and Management, Vol 15, No. 5, 1979, pp. 235-245.

[7] Wong, S.K.M., Ziarko, W., Raghavan,V. V., and Wong, P. C. N. Extended Boolean query processing in the generalized vector space Model, Information Systems Vol. 14, No. 1, pp. 47-63, 1989.

[8] Joon Ho Lee. Properties of Extended Boolean Models in Information Retrieval. In Proceedings of ACM-SIGIR Conference, 1994, pp. 182-190.

[9] Salton, G., Fox, E. A., and Voorhees, E. Advanced feedback methods in information retrieval. J. of the American Society for Information Science, 36(3): pp. 200-210, 1985.

[10] Alsaffar, A. H., Deogun, J. S., Raghavan, V. V., and Sever, H. Concept-based retrieval with minimal term sets. In Z. W. Ras and A. Skowon, editors, Foundations of Intelligent Systems: Eleventh Int'l Symposium, ISMIS'99 proceedings, pp. 114-122, Springer, Warsaw, Poland, Jun, 1999.

[11] Raghavan, V. V. and Wong, S. A critical analysis of the vector space model for information retrieval. Journal of the American Society for Information Science 37(5): pp. 279-287, 1986.
[12] Salton, G. and McGill, M. J. Introduction to Modern Information Retrieval. McGraw Hill, New York, 1983.
[13] J. T. Rickman, Design Considerations for a Boolean Search system with Automatic Relevance Feedback Processing, Proc. National Meeting, Assoc. for Computing Machiner, New York, August 1971, p. 478-481.
[14] M. Dillon and J. Desper, Automatic relevance feedback in Boolean retrieval system, J. Documentation 1980. 36, 197-208.
[15] M. Dillon and J. Ulmschneider and J. Desper, A prevalence formula for automatic relevance feedback in Boolean retrieval system, Infor. Proc. Management 1983, 19(1), 27-36.
[16] A.K. Jain and R.C. Dubes. Algorithms for clustering Data, PrenticeHall, Upper Saddle River, NJ, 1988.
[17] Efthimis N. Efthimiadis. Query Expansion. Annual Review of Information System and Technology, v31, pp. 121-187, 1996.
[18] Robertson, Stephen E., Sparck Jones, Karen. Relevance Weighting of Search Terms. Journal of the American Society for Information Science, 27(3), pp. 129-146, 1976.
[19] Robertson, Stephen E. On Relevance Weight Estimation and Query Expansion. Journal of, 42(3), pp. 182-188, 1986.
[20] Porter M.F. and Galpin V. Relevance Feedback in a Public Access Catalogue for a Research Library: Muscat at the Scott Polar Research Institute. Program, 22(1), pp. 1-20, 1988.

최 종 필

1994년 아주대학교 공과대학 컴퓨터공학과 학사. 1999년 아주대학교 컴퓨터공학과 석사. 2001년 아주대학교 컴퓨터공학과 박사 수료. 2002년~현재 아주대학교 정보통신연구소 연구원, 프로그래밍 전문강사. 관심분야는 지능형 정보검색 시스템, 신경망, 기계학습, 온톨로지

김 민 구

1977년 서울대학교 계산통계학과(이학사) 1979년 한국과학기술원 전산학과(공학석사). 1989년 Pennsylvania 주립대 전산학과(박사). 1999년~2000년 Louisiana 대학 연구과학자. 1981년~현재 아주대학교 컴퓨터공학과(교수). 관심분야는 지능형 정보검색 시스템, 지능형 교수 시스템, 지능형 캐릭터 에이전트