퍼지 집합 모델의 검색 효율 개선을 위한 퍼지 연산자의 분석

Fuzzy Operator Analyses to Imporve Retrieval Effectiveness of the Fuzzy Set Model

이준호(Joon Ho Lee)*
김원용(Won Yong Kim)**
이윤준(Yoon Joon Lee)**
김명호(Myoung Ho Kim)**

□목 차□

초 록

AND와 OR에 대한 연산식으로 MIN과 MAX를 사용하는 기존의 퍼지 집합 모델은 많은 경우에 사람이 생각하는 것과 다른 문서값을 생성하기 때문에 정보 검색 모델로서 부적합하다고 비판되어 왔다. 퍼지 집합 이론이 도입된 이후로 AND와 OR에 대한 연산식으로 다양한 퍼지 연산자들이 개발되어 왔다. 본 논문에서는 이러한 퍼지 연산자들의 문서값 생성 특성을 분석하고, MIN과 MAX 대신에 긍정적 보상 연산자라 불리는 퍼지 연산자를 사용할 것을 제안한다. 긍정적 보상 연산자를 사용하는 퍼지 집합 모델이 보다 우수한 검색 효율을 제공함을 실험을 통하여 입증한다.

ABSTRACT

The conventional fuzzy set model has been criticized as a retrieval model because the MIN and MAX operators have the properties adverse to effective calculation of document values. Since the first introduction of fuzzy set theory a variety of fuzzy operators have been developed, which can replace the MIN and MAX operators. We analyze their behavioral aspects of generating document values, and propose the enhanced fuzzy set model based on a class of fuzzy operators called positively compensatory operators. We also show through performance experiments that the proposed fuzzy set model provides higher retrieval effectiveness.

*한국과학기술원 인공지능연구센터
**한국과학기술원 전산학과

1. 서 론

오늘날 정보 검색 분야에서 가장 널리 사용되는 시스템은 불리안 검색 시스템(Boolean Retrieval System)이다. 이것은 불리안 검색 시스템이 짧은 검색 시간을 제공하고, 불리안 연산자들을 사용함으로써 비교적 쉽고 정확하게 질의를 표현할 수 있기 때문이다. 그러나 불리안 검색 시스템은 정보 검색 시스템의 중요한 역할중의 하나인 순위 결정 방법(Ranking Method)을 제공하지 않는다. 이러한 단점을 개선하기 위해 퍼지 집합 모델(Fuzzy Set Model)[1, 2, 3]이 제안되었다. 퍼지 집합 모델은 문서 내에서 색인어의 중요도를 나타내는 색인어 가중치를 이용하여 질의와 문서 사이의 유사도를 나타내는 문서값(Document Value)을 계산함으로써 문서의 순위를 결정한다. 그러나 퍼지 집합 모델은 AND와 OR 연산을 위하여 사용하는 MIN과 MAX 연산자가 많은 경우에 사람이 생각하는 것과 다른 문서값을 생성하기 때문에 정보 검색 모델로서 부적합하다고 알려졌다[4, 5].

퍼지 집합 이론이 도입된 이후로 AND OR에 대한 계산식으로 다양한 퍼지 연산자들이 제안되어 왔으며, 이러한 퍼지 연산자들은 T-연산자(T-Operator)와 평균 연산자(Averaging Operator)로 분류된다[6]. 본 논문에서는 기존의 퍼지 집합 모델의 문제점을 MIN과 MAX 대신에 T-연산자를 사용할지라도 해결할 수 없음을 설명한다. 또한 긍정적 보상 연산자(Positively Compensation Operator)라 불리는 일부의 평균 연산자들은 높은 검색 효율(Retrieval Effectiveness)을 제공하는 특성들을 지니고 있음을 기술하고, 긍정적 보상 연산자를 불리안 연산자 계산식으로 사용할 것을 제안한다. 긍정적 보상 연산자의 이항 연산식은 탐색어의 중요도를 왜곡하여 문서값을 계산하기 때문에, 이를 다항 연산이 가능하도록 확장함으로써 문서값의 왜곡을 감소시킬 수 있음을 설명한다. 마지막으로 실험을 통하여 긍정적 보상 연사자를 사용하는 퍼지 집합 모델은 다른 연산자를 사용하는 퍼지 집합 모델보다 높은 검색 효율을 제공함을 입증한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 기존의 퍼지 집합 모델의 문제점을 기술한다. Ⅲ장에서는 퍼지 집합 이론에서 AND와 OR 연산을 위하여 지금까지 개발된 다양한 퍼지 연산자들이 정보 검색 시스템의 검색 효율에 미치는 영향을 설명하고, 이를 기반으로 불리안 연산자 계산식을 제안한다. Ⅳ장에서는 실험을 통하여 제안하는 방법의 검색 효율을 평가하며, 끝으로 Ⅴ장에서 결론과 앞으로의 연구 방향을 제시한다.

2. 퍼지 집합 모델

퍼지 집합 모델은 색인어가 문서에서 갖는 중요도를 반영하는 색인어 가중치를 이용한다. 퍼지 집합 모델을 기반으로 하는 정보 검색 시스템은 〈T, Q, D, F〉로 정의될 수 있다. T는 질의와 문서를 표현하기 위해 사용되는 색인어들의 집합이고, Q는 시스템이 인식할 수 있는 질의들의 집합이다. Q에 속하는 각각의 질의 q는 색인어들과 불리안 연산자 AND, OR 그리고 NOT으로 구성된 불리안 수식이다. D는 문서들의 집합이고, D에 속하는 각각의 문서 d는 w_i가 색인어 t_i의 가중치일 때$((t_1, w_1),\cdots,(t_n, w_n))$와 같이 표현된다. 색인어 가중치 w_i는 0부터 1사이의 값을 갖는다.

F는 문서값을 계산하는 검색 함수(Retrieval

Function)로서 $F: D \times Q \rightarrow [0, 1]$과 같이 정의된다. 검색 함수 F는 각 쌍의 (d, q)에 0부터 1사이의 값을 할당한다. 이 값은 문서 d와 질의 q사이의 유사도를 의미하며 질의 q에 대한 문서 d의 문서값이다. 검색 함수 F(d, q)는 다음과 같은 2단계 과정을 거쳐서 계산된다.

1. 질의에 나타난 각각의 탐색어 t_i에 대하여, $F(d, t_i)$는 문서 d에서 색인어 t_i의 가중치 w_i로 정의된다.
2. 불리안 연산자 AND, OR 그리고 NOT은 다음의 식을 이용하여 계산된다.

$F(d, t_1 \text{ AND } t_2) = MIN(F(d, t_1), F(d, t_2))$

$F(d, t_1 \text{ OR } t_2 = MAX(F(d, t_1), F(d, t_2))$

$F(d, \text{NOT } t_1) = 1 - F(d, t_1)$

두개 이상의 불리안 연산자를 포함하는 불리안 질의는 가장 안쪽에 위치하는 절부터 순환적으로 계산된다.

문서의 순위는 질의에 대해서 계산된 문서값의 크기에 따라 결정된다. 그러나 퍼지 집합 모델은 많은 경우에 부정확한 문서값을 생성하기 때문에 정보 검색 모델로서 부적합하다고 알려져 왔다[4, 5]. 보기 1과 2는 AND 연산을 위해 MIN 연산자를 사용하는 기존의 퍼지 집합 모델이 사람이 생각하는 것과 다르게 문서의 순위를 결정함을 보여준다. (MAX 연산자도 MIN 연산자와 유사한 문제점들을 발생시킨다.)

보기1 : 색인어와 가중치의 쌍으로 표현된 두개의 문서 d_1, d_2와 불리안 질의 q_1이 다음과 같이 주어졌다고 가정하자.

$d_1 = \{(\text{Thesaurus}, 0.40), (\text{Clustering}, 0.40)\}$

$d_2 = \{(\text{Thesaurus}, 0.99), (\text{Clustering}, 0.39)\}$

$q_1 = \text{Thesaurus AND Clustering}$

MIN 연산자가 AND 연산을 위해 사용되었을 때, 질의 q_1에 대한 d_1과 d_2의 문서값은 각각 0.40과 0.39이다. 따라서 기존이 퍼지 집합 모델은 d_1이 d_2보다 높은 순위를 갖는 것으로 결정한다. 그러나 대부분의 사람들은 d_2의 질의에 대한 만족도가 d_1보다 높은 것으로 결정할 것이다.

보기2 : 두개의 문서 d_3와 d_4, 그리고 질의 q_2가 다음과 같이 주어졌다고 가정하자.

$d_3 = \{(t_1, 0), (t_2, 1), (t_3, 1), \cdots, (t_{100}, 0)\}$

$d_4 = \{(t_1, 0), (t_2, 0), (t_3, 0), \cdots, (t_{100}, 0)\}$

$q_2 = t_1 \text{ AND } t_2 \text{ AND } \cdots t_{100}$

이때 기존의 퍼지 집합 모델은 질의 q_2에 대한 d_3의 문서값과 d_4의 문서값이 동일한 것으로 결정한다. 또한 문서 d_3가 질의에 지정된 99개의 색인어를 포함하고 있음에도 불구하고 d_3의 문서값을 0으로 계산한다.

보기 1과 2에서 설명된 기존의 퍼지 집합 모델의 문제점은 MIN과 MAX 연산자가 두개의 피연산자를 모두 고려하지 않고 단지 하나의 피연산자에 전적으로 의존하는 결과값을 생성하기 때문에 발생한다. MIN과 MAX 연산자가 발생시키는 이러한 문

제를 "단일 피연산자 의존 문제(Single Operand Dependency Problem)"라 한다.

3. 검색 효율 개선을 위한 퍼지 연산자들의 분석

3.1 퍼지 연산자의 분류

퍼지 집합 이론은 0부터 1사이의 소속값(Membership Value)을 갖는 원소들의 집합에, 집합 이론에서 정의된 집합 연산자에 대응하는 새로운 퍼지 연산자를 정의함으로써 개발되어 왔다. 일반적으로 하나의 집합 연산자에 대하여 이에 대응하는 다수의 퍼지 연산자가 개발되고 있으며, 서로 다른 퍼지 연산자는 서로 다른 특성을 지닌다. 이러한 퍼지 연산자들은 T-nofrm, T-conorm 그리고 평균연산자로 구분된다[6]. 그림 1은 지금까지 개발된 T-norm, T-conorm을 나타내고 그림 2는 평균연산자를 나타낸다. T-연산자라 불리는 T-norm과 T-conorm은 각각 AND와 OR에 대한 계산식으로 사용되며, A_4를 제외한 평균연산자는 매개변수 값을 조절하여 AND와 OR에 대한 계산식으로 사용된다.

3.2 T-연산자의 사용에 대한 문제점

MIN과 MAX 이외의 T-연산자들은 다음과 같은 두가지 공통적 특성을 지닌다. 첫째, 하나의 피연산자가 0 또는 1일 경우 MIN, MAX와 마찬가지로 하나의 피연산자에 전적으로 의존하는 결과값을 생성한다. 둘째, 피연산자의 값들이 0과 1이 아닐 경우 두개의 피연산자를 모두 고려하여 결과값을 계산하며, 계산된 결과값은 피연산자 값들의 최소값보다 작거가 최대값x보다 크다.

퍼지 집합 모델에서 MIN과 MAX 이외의 T-연산자들의 사용은 위에서 언급된 첫번째 공통적 특성으로 인하여 보기 2에서 기술된 문제점을 여전히 발생시키지만 두개의 피연산자를 모두 고려하여 결과값을 계산하기 때문에 보기 1에서 기술된 문제점을 완화시킨다. 예를 들어, 보기 1에서 MIN 연산자 대신에 곱하기 연산자 T_2를 사용한다고 가정하자. 이때 d_1과 d_2에 대한 문서값은 각각 0.16과 0.39로 계산되기 때문에, d_2가 d_1보다 높은 순위를 부여받는다. 그러나 계산된 결과값이 피연산자 값들의 최소값보다 작거나 최대값보다 크다는 특성은 다음의 보기와 같은 새로운 문제점을 발생시킨다.

보기3 : 문서 d_5와 두개의 질의 q_1과 q_3가 다음과 같이 주어졌다고 가정하자.

d_5={(Thesaurus, 0.70), (Clustering, 0.70), (System, 0.70)}

q_1=Thesaurus AND Clustering

q_3=System

MIN과 MAX를 제외한 T-연산자들은 AND 연산에 있어서 두개의 피연산자들의 최소값보다 작은 값을 생성한다. 따라서 이러한 연산자들을 불리안 연산자 계산식으로 사용하는 퍼지 집합 모델은 문서 d_5와 질의 q_1 사이의 유사도가 d_5와 q_3 사이의 유사도보다 적은 것으로 결정할 것이다. 예를 들어, 곱하기 연산자 T_2가 사용되었을 때, 질의 q_1에 대한 문서 d_5의 문서값은 0.49이고 q_3에 대한 d_5의 문서값은 0.70이다. 이러한 결정은 대부분 사람들의 의사와 상반되는 것으로 정보 검색 시스템의 검색 효율을 저하시킨다.

MIN과 MAX 이외의 T-연산자들을 불리안 연산

	T(x, y)	Tc(x, y)	Comment
1	$MIN(x, y)$	$MAX(x, y)$	
2	$x \bullet y$	$x+y-xy$	
3	$MAX(x-y-1, 0)$	$MIN(x-y, 1)$	
4	$\frac{xy}{x+y-xy}$	$\frac{x+y-2xy}{1-xy}$	
5	$\begin{cases} x \text{ if } y=1 \\ y \text{ if } x=1 \\ 0 \text{ otherwise} \end{cases}$	$\begin{cases} x \text{ if } y=1 \\ y \text{ if } x=1 \\ 0 \text{ otherwise} \end{cases}$	
6	$\frac{\lambda xy}{1-(1-\lambda)(x+y-xy)}$	$\frac{\lambda(x+y)+xy(1-2\lambda}{\lambda+xy(1-\lambda)}$	$0 \le \lambda \le \infty$
7	$MAX(1-))1-x-p+(1-y)p)^{1/p}, 0)$	$MIN((xp-yp)^{1/p}, 1)$	$1 \le p \le \infty$
8	$\frac{1}{1+\left\{\left\lfloor \frac{1}{x}-1 \right\rfloor^{\lambda}+\left\lfloor \frac{1}{y}-1 \right\rfloor^{\lambda}\right\rfloor^{1/\lambda}}$	$\frac{1}{1+\left\{\left\lfloor \frac{1}{x}-1 \right\rfloor^{-\lambda}+\left\lfloor \frac{1}{y}-1 \right\rfloor^{-\lambda}\right\rfloor^{-1/\lambda}}$	$0 \le \lambda \le \infty$
9	$\frac{xy}{MAX(x, y, \lambda)}$	$1-\frac{(1-x)(1-y)}{MAX(1-x, 1-y, \lambda)}$	$0 \le \lambda \le \infty$
10	$MAX\left\lfloor \frac{x+y-1+\lambda xy}{1+\lambda}, 0 \right\rfloor$	$MIN(x+y+\lambda xy, 1)$	$-1 \le \lambda \le \infty$
11	$MAX((1+\lambda)(x-y-1)-\lambda xy, 0)$	$MIN(x+y+\lambda xy, 1)$	$-1 \le \lambda \le \infty$

[그림 1] T-연산자

(A_1) $(x \bullet y)^{1-\gamma} \bullet (x+y-x \bullet y)^{\gamma}$, $0 \le \gamma \le 1$

(A_2) $(1-\gamma) \bullet MIN(x, y)+\gamma \bullet MAX(x, y)$, $0 \le \gamma \le 1$

(A_3) $(1-\gamma) \bullet (x \bullet y)+\gamma \bullet (x+y-x \bullet y)$, $0 \le \gamma \le 1$

$(A_{4.AND})$ $\gamma \bullet MIN(x, y)+\frac{(1-\gamma)(x+y)}{2}$, $0 \le \gamma \le 1$

$(A_{4.OR})$ $\gamma \bullet MAX(x, y)+\frac{(1-\gamma)(x+y)}{2}$, $0 \le \gamma \le 1$

[그림 2] 평균연산자

자 계산식으로 사용하는 퍼지 집합 모델의 문제점들은 다음과 같이 요약될 수 있다. 첫째, 연산자들의 첫번째 공통적 특성은 보기 2와 같은 형태의 단일 피연산자 의존 문제를 발생시킨다. 둘째, MIN과 MAX 이외의 T-연산자들은 결과값을 계산하기 위해 두개의 피연산자 값들의 상호 보상(Compensation)을 허용한다. 그러나 이러한 보상이 검색 효율에 부정적인 방향으로 이루어지기 때문에 "부정적 보상 문제(Negative Compenstation Problem)"라고 하는 보기 3과 같은 새로운 문제를 발생시킨다.

3.3 긍정적 보상 연산자의 적합성

불리안 연산잔 계산식이 항상 피연산자들의 최소값과 최대값 사이의 값을 생성하는 특성을 지니고 있다면, 보기 1, 2 그리고 3을 통하여 설명된 단일 피연산자 의존 문제와 부정적 보상 문제를 극복할 수 있다. 본 논문에서는 이러한 특성을 갖는 연산자를 "긍정적 보상 연산자(Positively Compensatory Operator)"라고 지칭하고 퍼지 집합 모델의 불리안 연산자 계산식으로 사용할 것을 제안한다.

평균연산자 A_2와 A_4가 긍정적 보상 연산자에 포함되며 평균연산자 A_1과 A_3은 일부분의 경우에 단일 피연산자 의존 문제와 부정적 보상 문제를 발생시킨다. A_1은 하나의 피연산자 값이 0일 때 결과값으로 항상 0을 생성하므로, A_1을 기반으로 하는 퍼지 집합 모델은 보기 2와 같은 형태의 단일 피연산자 의존 문제를 발생시킨다. A_1과 A_3은 피연산자 값들의 일부 범위내에서, 피연산자들의 최소값보다 작거나 최대값보다 큰 결과값을 생성하는 부정적 보상을 허용한다. 따라서 A_1 또는 A_3을 기반으로 하는 퍼지 집합 모델은 보기 3에서 설명된 부정적 보상 문제를 회피할 수 없다.

평균연산자 A_2와 A_4는 각각 다른 사람에 의해 개발되었을지라도 수학적으로 같은 식임을 증명할 수 있다. A_2는 매개변수의 변위를 다음과 같이 분할하여 AND와 OR에 대한 연산자 계산식으로 사용된다.

$(A_{2.AND})\ (1-\gamma)\bullet MIN(x, y)+\gamma\bullet MAX(x, y),\quad 0\le\gamma\le 0.5$

$(A_{2.OR})\ (1-\gamma)\bullet MIN(x, y)+\gamma\bullet MAX(x, y),\quad 0.5\le\gamma\le 1$

$(A_{2.AND})$의 매개변수 γ를 $1-\gamma$로 치환하면 $(A_{2.AND})$와 $(A_{2.OR})$의 매개변수의 변위를 일치시킬 수 있다. 매개변수의 변위를 수정한 $(A_{2.AND})$의 식은 다음과 같다.

$(A'_{2.AND})(1-\gamma)\bullet MAX(x, y)+\gamma\bullet MIN(x, y),\quad 0.5\le\gamma\le 1$

이때 $A'_{2.AND}$와 $A_{2.OR}$의 매개변수 γ를 $(\gamma+1)/2$로 치환하면 $A_{4.AND}$와 $A_{4.OR}$의 식을 얻을 수 있다.

정보 검색 분야에서 확장된 불리안 모델은 효율적인 검색 모델로 알려져 왔다[7, 8]. 확장된 불리안 모델과 퍼지 집합 모델의 차이점은 문서값 계산을 위한 불리안 연산자 계산식이다. 질의 가중치를 고려하지 않을 경우 확장된 불리안 모델의 불리안 연산자 계산식은 다음과 같으며, 이들은 긍정적 보상 연산자이다.

$$(E_{AND})\quad 1-\left[\frac{(1-x)^p+(1-y)^p}{2}\right]^{1/p}\qquad 1\le p\le\infty$$

$$(E_{OR})\quad \left[\frac{x^p+y^p}{2}\right]^{1/p}\qquad 1\le p\le\infty$$

3.4 긍정적 보상 연산자의 다항 연산

질의(t_1 OR t_2 OR t_3)에 나타난 탐색어 t_1, t_2, t_3는 같은 중요도로 사용자가 요구하는 정보를 표현한다. 그러나 평균연산자 A_4는 매개변수 값이 1인 경우를 제외하고, 연산의 수행 순서에 따라 탐색어의 중요성을 왜곡하여 연산 결과값을 생성한다. 즉 (t_1 OR t_2 OR t_3)에 대해서 ((t_1 OR t_2) OR t_3)와 (t_1 OR(t_2 OR t_3))의 연산 결과값이 서로 다르다. ((t_1 OR t_2) OR t_3)에 대해서 평균연산자 A_4는 탐색어 t_1, t_2보다 t_3를 더 중요한 탐색어로 취급하며, (t_1 OR t_2 OR t_3))에 대해서는 t_2, t_3보다 t_1을 더 중요한 탐색어로 취급한다. AND 연산자도 연산의 수행 순서에 따라 다른 결과값을 생성한다.

매개변수 값이 0인 경우에, 연산 수행 순서에 따라 탐색어의 중요도를 왜곡하는 정도를 계산할 수 있다. 질의를 (t_1 OR t_2 OR...OR t_n)이라 하고, 문서 d_i에서 색인어 (t_1, t_2,...,t_n)들의 가중치를 (w_1, w_2,...,w_n)이라고 가정하자. 평균연산자 A_4를 적용하여 질의를 왼쪽에서 오른쪽으로 연산하면, 다음과 같은 문서 d_i의 문서값을 얻을 수 있다.

$$F(d_i, (t_1 \text{ OR } t_2 \text{ OR...OR } t_n))$$
$$= \frac{w_1}{2^{n-1}} + \frac{w_2}{2^{n-1}} + \frac{w_3}{2^{n-2}} + \frac{w_4}{2^{n-3}} + \ldots + \frac{w_n}{2}$$
$$= \frac{1}{2}\left(\frac{w_1}{2^{n-2}} + \frac{w_2}{2^{n-2}} + \frac{w_3}{2^{n-3}} + \frac{w_4}{2^{n-4}} + \ldots + w_n\right)$$

위의 결과로부터 먼저 수행된 탐색어와 나중에 수행된 탐색어의 중요도가 다르게 취급되고 있음을 알 수 있다. 탐색어 t_1과 t_n을 보면 질의에서는 같은 중요도로 질의를 표현하고 있다. 그러나 탐색어 t_1의 중요도를 1이라고 할 때, 연산결과에서 t_n의 중요도는 $2^{n-2}(n \geq 2)$이다. 결론적으로 평균연산자 A_4를 적용하여 연산을 수행하면, 먼저 연산된 탐색어는 나중에 연산된 탐색어보다 훨씬 작은 중요도로 취급되며, n의 값이 크면 클수록 이러한 왜곡의 정도는 더욱 커진다.

이러한 왜곡을 해결하는 방법으로는 세 가지가 있다. (1) 탐색어의 중요도를 왜곡하지 않는 불리안 연산자 계산식을 고안한다. (2) 탐색어의 중요도가 왜곡된 정도를 계산하여 보상한다. (3) 불리안 연산자 계산식을 다항 연산식으로 확장하여 모든 탐색어를 한번에 연산한다. 본 논문에서는 평균연산자 A_4를 다음과 같이 다항 연산식으로 확장하는 방법을 제안한다. 다항 연산자 $A_{4.N}$은 항상 피연산자들의 최소값과 최대값 사이의 결과를 생성한다. 즉 $A_{4.N}$은 단일 피연산자 의존 문제와 부정적 보상 문제를 발생시키지 않는다.

$$(A_{4.AND.N}) \quad \gamma \bullet MIN(w_1,\ldots,w_n) + \frac{(1-\gamma)(w_1+\ldots+w_n)}{n}, \quad 0 \leq \gamma \leq 1$$

$$(A_{4.OR.N}) \quad \gamma \bullet MAX(w_1,\ldots,w_n) + \frac{(1-\gamma)(w_1+\ldots+w_n)}{n}, \quad 0 \leq \gamma \leq 1$$

확장된 불리안 모델의 불리안 연산자 계산식도 평균연산자 A_4처럼 이항 연산을 할 경우 탐색어의 중요도를 왜곡하기 때문에 다항 연산을 가능하도록 하고 있다. 확장된 불리안 모델의 다항 연산식은 다음과 같다.

$$(E_{AND.N}) \quad 1-\left[\frac{(1-w_1)^p+\ldots+(1-w_n)^p}{n}\right]^{1/p}, \quad 1 \leq p \leq \infty$$

$$(E_{OR.N}) \quad \left[\frac{w_1^p+\ldots+w_n^p}{n}\right]^{1/p}, \quad 1 \leq p \leq \infty$$

4. 성능 평가 및 분석

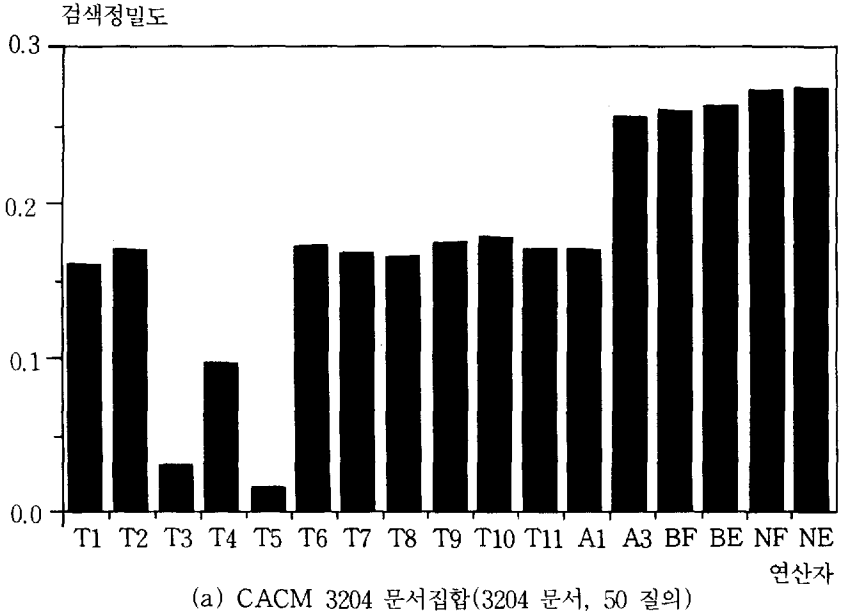
정보 검색 시스템의 검색 효율은 일반적으로 재현 능력도(Recall)와 검색 정밀도(Precision)를 이용하여 평가된다. 재현 능력도는 문서 집합에서 사용자가 원하는 문서를 어느 정도 검색하였는가를 나타내고, 검색 정밀도는 검색된 문서들 중에서 사용자가 원하는 문서가 얼마나 포함되어 있는가를 나타낸다. 예를 들어 200개의 문서로 구성된 문서 집합과 관련된 문서의 수가 5인 질의를 가정하자. 이때 사용자가 검색 시스템을 이용하여 6개의 문서를 검색하였고, 검색된 문서 중에서 질의에 관련된 문서가 4개 존재하였다면 재현 능력도와 검색 정밀도는 각각 0.8, 0.67이 된다.

문서의 순위 결정 방법을 제공하는 검색 시스템은 보간 기법을 사용하여 고정된 재현 능력도에 대한 검색 정밀도를 계산할 수 있다[9]. 본 논문에서는 검색 시스템의 검색효율을 평가하기 위하여 질의들에 대한 평균 검색 정밀도를 이용한다. 임의의 질의에 대한 검색 정밀도는 재현 능력도를 0.25(낮은 재현 능력도), 0.5(중간 재현 능력도), 0.75(높은 재현 능력도)에 고정시켜 계산된 검색 정밀도들의 평균값이다.

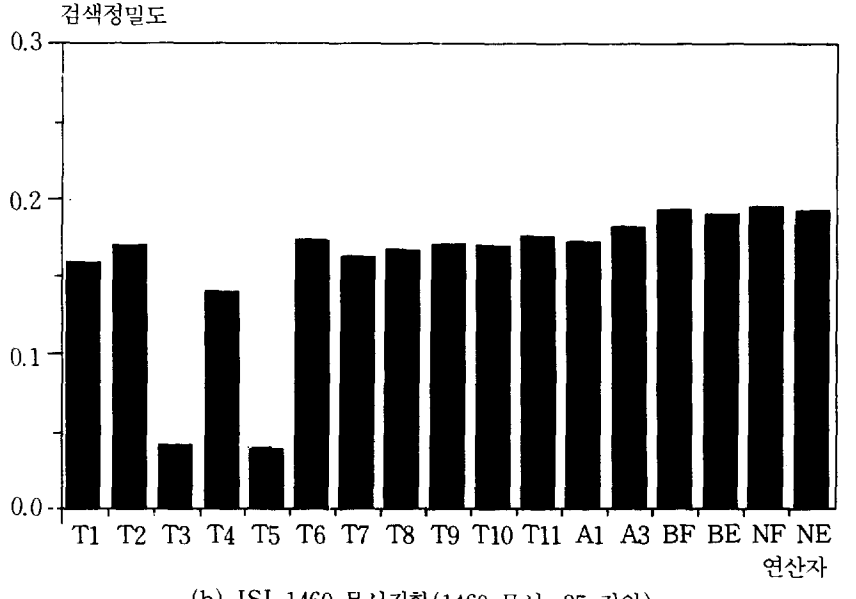
실험에 사용된 컴퓨터는 MIPS rc-3230이고, 데이타 집합으로 CACM 3204와 ISI 1460 두 종류를 사용하였다[7]. CACM은 3,204개의 문서와 64개의 질의로 구성되어 있으며, ISI는 1,460개의 문서와 35개의 질의로 구성되어 있다. 각 문서 집합의 질의에 대한 문서의 관련성 평가는 Cornell 대학에서 수행되었다. CACM 3204의 경우 일부 질의에 대하여 문서 관련성 평가가 없으며, 2개의 질의는 특정 저자가 쓴 문서를 검색하는 질의이다. 본 실험에서는 관련성 평가가 없는 질의와 특정 저자에 대한 질의를 제외한 CACM 3204의 50개와 ISI 1460의 35개의 질의를 사용하여 실험하였다.

그림 3은 퍼지 집합 모델에서 AND와 OR에 대한 연산자 계산식으로 퍼지 연산자를 사용하였을 때의 검색 효율을 비교한 실험 결과이다. 본 논문에서는 매개변수를 가지고 있는 연산자인 경우 가장 높은 검색 효율을 나타내는 매개변수 값을 다음과 같이 결정하여 비교하였다. (1) 매개변수 변위의 가장 작은 값에서부터 일정한 간격으로 가장 큰 값까지 증가시키면서 검색 효율을 측정한다. (2) 가장 좋은 검색 효율을 나타내는 매개변수 값의 변위를 찾아서 더 작은 간격으로 (1)과 같이 검색효율을 측정한다. (3) (2)를 검색 효율의 변화가 없을 때까지 반복하여 가장 좋은 검색효율을 나타내는 매개변수 값을 결정한다. 그림 3에서 T1－T11은 T-연산자들을 나타내고 A1과 A3은 평균연산자 A_1과 A_3을 나타낸다. 그리고 BF와 BE는 평균연산자 A_4와 확장된 불리안 모델의 이항 연산식이고 NF와 NE는 다항 연산식이다.

그림 3에서 ISI 1460 문서 집합의 경우 검색 효율의 변화가 CACM 3204보다 적다. 그 이유는 각 질의에 대하여 관련된 문서수의 평균이 CACM은 15개이고 ISI는 50개로써, CACM 보다 ISI가 광범위의 질의를 많이 포함하기 때문이다. 그림 3은 긍정적 보상 연산자가 다른 연산자들보다 높은 검색 효율을 제공하고 있음을 나타낸다. 이것은 긍정적 보상 연산자가 단일 피연산자 문제와 부정적 보상 문제를 발생시키지 않기 때문이다. A_3은 단일 피연산자 의존 문제를 발생시키지 않고 극히 일부분의 경우에만 부정적 보상 문제를 발생시키기 때문에 긍정적 보상 연산자와 유사한 검색 효율을 나타내고 있다. 또한 탐색어의 중요도를 왜곡하지 않



(a) CACM 3204 문서집합(3204 문서, 50 질의)



(b) ISI 1460 문서집합(1460 문서, 35 질의)

[그림 3] 퍼지 연산자의 검색 효율 비교

는 긍정적 보상 연산자의 다항 연산식이 이항연산식 보다 높은 검색 효율을 제공하고 있다.

5. 결론 및 앞으로의 연구

기존의 퍼지 집합 모델은 검색 효율에 대한 MIN과 MAX 연산자의 부정적 특성으로 인하여 부적합한 검색 모델로 알려져 왔다. 한편, 퍼지 집합 이론에서 MIN과 MAX 연산자를 대신할 수 있는 다양한 퍼지 연산자들이 개발되었다. 본 논문에서는 이러한 퍼지 연산자들의 특성을 분석함으로써 높은 검색 효율을 제공할 수 있는 긍정적 보상 연산자를 정의하고, 이들을 퍼지 집합 모델의 뷸리안 연산자 계산식으로 사용할 것을 제안하였다. 또한, 현재까지 개발된 긍정적 보상 연산자는 탐색어의 중요도를 왜곡하여 문서값을 계산하기 때문에, 긍정적 보상 연산자를 다항 연산식으로 확장하였다. 긍정적 보상 연산자를 사용하는 퍼지 집합 모델이 다른 연산자를 사용하는 퍼지 집합 모델보다 높은 검색 효율을 제공하고, 긍정적 보상 연산자의 다항 연산식이 이항 연산식보다 높은 금색 효율을 제공함을 실험을 통하여 입증하였다.

앞으로의 연구는 다음과 같다. 첫째, 본 논문에서 제안된 긍정적 보상 연산자는 결합, 배분법칙과 같은 유용한 특성들을 만족하지 않는다. 따라서 이러한 특성들을 만족하는 긍정적 보상 연산자의 존재 여부가 입증되어야 한다. 둘째, 보다 효율적인 검색을 위하여 탐색어 가중치의 연산 방법이 연구되어야 한다.

참고문헌

(1) A.Bookstein, "A Comparison of Two Weighting Schemes for Boolean Retrieval," *Journal of the Americal Society for Information Science*, Vol. 32, No. 4, pp. 275-279, 1981.

(2) T. Radecki, "Fuzzy Set Theoretical Approach to Document Retrieval," *Information Processin & Management*, Vol. 15, No. 5, pp 247-259, 1979.

(3) W.G. Waller and D.H. Kraft, "A Mathematical Model of a Weighted Boolean Retrieval System," Information Processing & Management, Vol. 15, pp. 235-245, 1979

(4) A. Bookstein, "Fuzzy Requests:An Approach to Weighted Boolean Seraches," *Journal of the American Society for Information Science*, Vol. 31, No. 4, pp. 240-247, 1980.

(5) S.E. Robertson, "On the Nature of Fuzz: A Diatribe," *Journal of the American Society for Information Science*, Vol.29, No. 6, pp. 304-307, 1978.

(6) H.J. Zimmermann, "Fuzzy Set Theory and ITs Applications," *2nd ed., Kluwer Academic Publishers*, 1991.

(7) G. Salton, E.A. Fox, and H. Wu, "Extended Boolean Information Retrieval," *Communications of the ACM*, Vol. 26, No. 11, pp. 1022-1036, 1983.

(8) G. Salton, "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer," *Addison Wesley*, 1989.

(9) G. Salton, Ed. "The Smart Retrieval System-Experiments in Automatic Document Processing", *Prentice Hall. Inc., Englewood Clifts, New Jersey,* 1971.