정보검색시스템의 확률 및 벡터모델에 대한 질의 확장 검색 성능 평가†

(Extended Query Search Performance Evaluations for Vector Model and Probabilistic Model of Information System)

전 유 정*, 변 동 률**, 박 순 철***
(Yu-jung Jeon, Dong-ryul Byeon, Soon-cheol Park)

요 약 본 논문은 벡터모델과 확률모델의 성능 비교에 관한 연구이다. 벡터모델로써는 잠재적 의미를 적용한 검색 결과를 찾기 위해 사용되는 LSI 모델을 이용하였다. 확률모델로써는 현재 상용화 단계에 있는 콘도르 정보검색 시스템을 적용하였다. 각 모델 시스템의 검색 성능 비교를 위한 실험은 사용자가 입력한 원래 질의어에 관한 검색 결과를 바탕으로 성능을 비교한 후에, 사전적 의미를 적용한 확장 질의어에 대한 검색 결과를 추가하여 비교하였다. 본 연구에서는 입력된 질의어와 관련된 용어를 추가하여 검색하였을 경우, 확률모델에 비해 벡터모델에서 성능이 대부분의 질의어에 대해서 향상됨을 보인다.

핵심주제어 : 정보검색, 확률모델, 벡터모델, LSI

Abstract In this paper, we compare the vector model performance with the probabilistic model of information system. We use LSI(Latent Semantic Indexing) model for vector model, while Condor information search system that is ready to sell on business is used as a probabilistic model. Each model produces the search results from the original queries and the queries extended by a dictionary definition. We compare those results between two models and find out the vector model is much better than the probabilistic model for the most queries.

Key Words : Information Retrieval, Probabilistic Model, Vector Model, LSI

1. 서 론

정보검색 분야에서 가장 널리 사용되어 온 시스템은 불리언 검색 시스템이었다. 불리언 검색 시스템은 구현이 쉽고, 짧은 검색 시간을 제공하며, 간단한 불리언 연산자들을 사용함으로써 비교적 쉽고 정확하게 질의를 표현할수 있다는 장점이 있다.

그러나 수기가(Giga) 바이트 이상의 대용량 문서 데이터 중에서 보다 정확한 문서의 검색결과를 얻기 위해서는 상대적으로 복잡한 불리언 질의를 사용해야 하기 때문에 불리언 시스템이 적합하지 않다. 이런 경우 검색 결과에 대해 문서의 우선순위를 매겨 상위 문서들을 사용자에게 제공하는 것이 가장 효율적이다.

우선순위에 따른 검색 결과는 사용자에게 필요한 정보를 얻는 시간을 최소화 시켜준다는 장점이 있다. 이러한 문서 순위 결정을 지원하기 위해, 퍼지 집합 모델, 확률모델, 벡터 공간 모델, 확장 불리언 모델, 지

† 이 논문은 2003년 한국학술진흥재단의 공모과제 연구비에 의하여 수행되었음.
* 전북대학교 대학원 정보통신공학과 석사과정
** 전북대학교 대학원 정보통신공학과 박사과정
*** 전북대학교 전자정보공학부 교수

식기반 검색모델 등과 같은 다양한 문서 순위 결정 방법론들이 제시되었다. [1]

확률모델이 벡터모델보다 더 우월하다는 주장에는 몇 가지 반론이 있지만, Croft의 실험에 따르면 확률모델이 더 나은 검색 성능을 보였다. 그러나 이후에 여러 다른 측정을 통하여 Salton과 Buckley는 일반 컬렉션의 경우, 벡터모델이 확률모델보다 우월하다는 점을 보여주어 반론을 제기하였다. 이 점이 정보검색 연구자, 종사자 및 웹 공동체의 지배적인 생각으로 받아들여지고 있다. [2] 이에, 본 연구에서는 벡터모델로써 LSI 모델의 검색 성능을 확률모델의 성능과 비교하고 질의 확장을 통해 그 성능을 더욱 개선시킬 수 있음을 실험을 통해 증명한다.

본 논문의 구성은 다음과 같다. 2장과 3장에서는 확률모델을 적용한 콘도르 정보검색시스템과 벡터모델로서의 LSI 모델에 관해 각각 살펴본다. 4장은 각 모델 시스템의 성능 비교 실험을 설명하고, 5장에서 결론과 향후 과제를 언급하겠다.

2. 콘도르 정보검색시스템

'콘도르' 정보검색시스템은 전북대학교, ㈜서치라인, 그리고 카네기멜론 대학교가 컨소시엄 형태로 개발한 시스템이다. 이 시스템의 질의처리는 확률모델을 기반하고 있으며 최근 정보검색시스템에서 제공하는 문서 클러스터링 기능을 제공하고 있다. 특히 시스템의 특징은 다중 언어 질의를 처리하고 질의를 중심으로 온라인으로 문서를 요약하는 것이다. 본 시스템은 이미 국내의 3,000만개 웹 페이지서 보이는 것에 대한 테스트를 마쳤으며 그 안정성을 확보하고 있다. 이 절에서는 확률모델인 콘도르 정보검색시스템의 특징을 알아보기 위하여 확률모델의 특징과 콘도르 정보검색시스템의 구조에 관해 간단히 살펴본다.

2.1 확률모델

확률모델은 정보검색 문제를 확률적 틀로 해석하고 있으며, 다음의 기본적인 가정에 기초하고 있다. [3]

사용자 질의 q와 컬렉션의 문헌 d_j가 주어지면, 사용자가 문헌 d_j에 흥미(즉, 연관)가 있을 확률을 추정한다. 이 연관성에 관한 확률은 질의와 문헌상에 사용된 색인어에 종속된다고 가정한다. 또 사용자가 질의 q의 해답 집합으로서 선호하는 부분 문헌 집합이 있다고 가정하며, 이 이상적인 해답 집합을 R이라 하면, 사용자는 전체적인 연관 확률을 최대화하여 이상적인 해답 집합의 확률적 표현으로 개선하게 된다. 이것은 정보검색 문제를 클러스터링 문제로 해석하는 것과 유사한 개념이다. R 집합에 속한 문헌들은 질의에 관련이 있다고 가정하며, R에 속하지 않은 문헌들은 비 연관 문헌으로 간주된다.

이 모델에서는 주어진 질의에 대해서 먼저 검색된 관련 문헌에 나타난 용어가 나타나지 않은 용어보다 높은 가중치를 갖는다. [4] 질의 q에 대한 연관문헌과 비 연관 문헌에서 용어 t에 대한 분포가 다음의 <표 1>에 나타나 있다.

<표 1> 확률모델의 연관 문헌과 비 연관 문헌

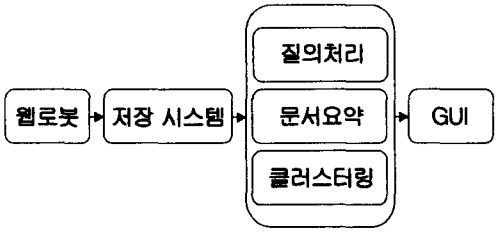
문헌 색인	문헌 연관성 +	문헌 연관성 -	
+	r	n-r	n
-	R-r	N-n-R+r	N-n
	R	N-R	N

N=전체 문헌 집합 수
R=질의 q에 대한 연관 문헌 수
n=용어 t가 있는 문헌 수
r=용어 t가 있는 연관 문헌 수

이 가정은 연관 확률 계산 방법을 명확히 밝히고 있지 않기 때문에 문제가 되는데, 본 연구에서는 확률모델을 바탕으로 구현된 콘도르 정보검색시스템에 적용하게 될 Data Set으로 사용한 KT-SET에서 구분된 그룹에 따라 그 연관성과 비연관성을 구분하였다.

2.2 콘도르 정보검색시스템

콘도르의 전체 구조는 크게 웹 로봇, 저장 시스템, 질의 처리부, 문서 요약, 클러스터링 그리고 GUI(Graphic User Interface) 부분으로 구성된다. [그림 1]은 콘도르의 각 기능에 따른 구조를 나타낸다.



[그림 1] 콘도르의 구조

각 모듈 별로 간단히 소개하면, 로봇과 저장 시스템은 일반 정보검색시스템과 유사하다. 질의어 처리를 위한 검색 모델은 확률모델을 따랐다. 또한 다중 언어(한국어, 중국어, 일본어, 영어) 질의 처리가 가능한 것도 콘도르의 특징이다.

문서 요약은 문서를 단순히 요약하는 오프라인 요약과 질의어에 포함된 용어를 중심으로 요약하는 온라인 요약을 포함한다. 특히 온라인 문서 요약은 현재까지 상용 정보검색시스템에서는 구현된 적이 없다.

문서 클러스터링의 구조는 계층 구조를 갖는다. 이러한 구조는 사용자의 판단에 따라 정확한 문서 집단을 택할 수 있도록 유도한다. 또한 클러스터링의 수는 임으로 정한 고정적인 것을 기본으로 하나 경우에 따라서는 그 수가 동적으로 변하는 알고리즘을 택하여 클러스터링의 결과를 좀더 정확하게 유지하도록 했다.

다른 시스템들과 마찬가지로 콘도르의 GUI 환경은 검색 시스템에 필요한 기능들을 사용자가 손쉽게 이용할 수 있도록 설계했다. 특히 클러스터링의 계층 구조를 가시화 했고 아울러 요약에는 온라인과 오프라인 요약을 구분하여 동시에 나타나도록 했다.

3. LSI 모델

3.1 LSI 모델

LSI (Latent Semantic Indexing)는 벡터 공간상에서 SVD (Singular Value Decomposition)를 이용한 개념 기반의 문서 검색 대수적 모델이다. 이를 이용하면 서술된 단어 자체뿐만 아니라 개념까지 비교가 가능하여 유사 단어까지 고려하기 때문에, 사용자의 질의에 대한 결과 문서 도출에 있어서 개념이 비슷한 문서까지 찾을 수 있는 장점이 있다.

기존의 개념 기반 검색에서는, 사용자의 질의어가 부정확할 경우, 수많은 정보들에 존재하는 내포된 의미를 파악하여 사용자가 원하는 관련된 정보를 검색한다는 것이 쉽지 않다. 질의어 벡터와 문서 벡터간의 유사성에 초점을 두었기 때문이다. 그리고 관련 문서를 검색하기 위해 질의어 용어와 의미가 비슷하거나 소리가 비슷한 용어를 이용한 질의어 확장을 했다.

본 논문에서는 질의어 벡터와 문서 벡터간의 유사성을 계산하기 이전에, 질의어 벡터와 용어 벡터간의 유사성을 먼저 측정하고, 질의어와 유사도가 높은 단어들을 구한다. 그리고 질의어의 사전적 의미를 이용한 질의어 확장을 한 후, 재현율, 정확률 측면에서 기존의 개념 기반 검색과 비교하였다.

문서에서 색인어는 그 자체보다 개념을 분석하는 것이 중요하다. 문서에 따라 개념은 같지만 다른 형태의 색인어들이 많다. 특히 관용구의 경우 사전적 의미나 구문론적 분석만을 따진다면 그 의미를 이해하기가 어렵다. 문서의 내용은 서술된 색인어보다는 그 안에 내포된 개념에 더 관련되어 있으므로 색인어 대신에 개념 기반이어야 한다. 이렇게 하면 문서들이 같은 색인어로 구성되어 있지 않더라도 연관성을 나타낼 수 있다. 어떤 문서가 다른 문서와 같은 개념을 공유한다면 유사한 문서라고 할 수 있다. 문서의 개념 기반에 대한 많은 시도들은 정보검색 엔진에 있어서 많은 발전을 가져왔다. 이러한 개념 분석에 있어서 두드러진 시도들 중 하나는 LSI 기술이었다. 이것은 단순한 색인어를 사용하는 것이 아니라 문서에 있어서 주요 개념들을 분석하기 위하여 다차원 확장을 이용하는 기술이다.[5][6]

LSI는 문서들이 색인어들의 벡터로 표현된 벡터 공간 검색 모델에 기반을 둔다. 하지만 이것은 색인어와 문서로 구성된 행렬을 SVD를 통해서 축소한다는 점에서 벡터 공간 모델과 다르다. [5]

3.2 SVD (Singular Value Decomposition)

m개의 용어, n개의 문서로 구성된 전체 집합을 A라고 할 때, Simple Query Matching 방법은 식 (1)과 같다.

$$Query = q^t * A \qquad (1)$$

식 (1)에서 q^t는 입력된 질의 전치행렬이다.

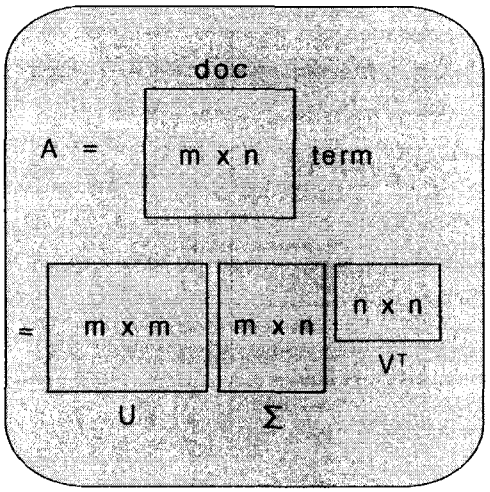
그런데, 일반적인 A 집합은 약 30만개의 용어와 3억 개의 문서로 이루어진다. 이렇게 큰 행렬을 이용하면 계산이 어려울 뿐만 아니라, 질의어 벡터의 정보에 비해 관련 없는 용어들과 문서들의 정보가 너무 많아진다는 문제점이 발생한다.

이를 해결하기 위해 수학적 Matrix Decomposition 중 SVD(Singular Value Decomposition)를 이용한다.

식 (2)는 행렬 A를 SVD 분해한 것을 나타낸다.

$$A = U\Sigma V^T \qquad (2)$$

식 (2)를 자세히 표현하면 [그림 2]와 같다. 식 (2)와 [그림 2]에서 U 행렬은 색인어간의 상관행렬, V 행렬은 문서간의 상관행렬, 그리고 Σ 행렬은 단일 값을 갖는 대각행렬이다. [6]



[그림 2] SVD 분해

SVD를 통해 분해 된 행렬을 이용해 LSI 모델을 적용한다. LSI 모델의 요점은 각 문서와 질의 벡터를 저차원 공간인 개념으로 사상시키는데 있다. 색인어 사이의 관계를 나타내는 행렬과 문서 사이의 관계를 나타내는 행렬을 같은 공간으로 사상시킨다.[7][8]

식 (3)를 이용하면 입력 질의어, q도 축소된 용어/문서와 같은 공간 안에서 표현될 수 있다.

$$q = q^T U_k \sum_k^{-1} \qquad (3)$$

따라서 같은 공간 내에 표현된 각 용어와 질의어 간의 유사성을 식 (4)의 코사인 유사도로 계산할 수 있다.

$$Sim(d,q) = \frac{\sum_k t_k \times q_k}{\sqrt{\sum_k t_k^2 \times \sum q_k^2}} \qquad (4)$$

4. 실험 환경 및 평가

4.1 실험 데이터

본 실험에서는 KT-SET93 문서들 중 문서번호 kt0001부터 kt0400번까지의 400개 문서를 실험 데이터로 이용하였다. <표 2>에서 용어 수는 콘도르 정보검색시스템의 인덱싱 모듈을 통하여 추출된 색인어의 개수이다

<표 2> 실험 데이터

문서집단	언어	분야	성격	문헌수	용어수
KT-SET 93	한국어 영어	전산학 정보학	초록	400	7782

400개의 문서들은 총 10개의 그룹으로 구분되는데, 본 실험에서는 그룹 별 해당 문서수를 고려하여 문서수가 너무 적거나 지나치게 많은 경우를 제외한 그룹을 대상으로 성능을 측정하였다. KT-SET 문서들은 콘도르의 DB 형식으로 변환하여 시스템에 입력하였다.

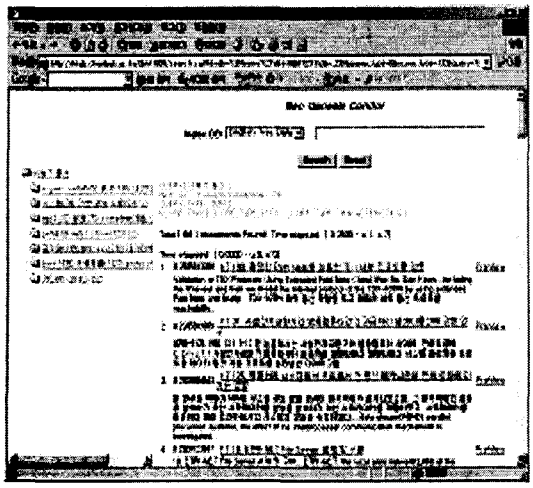
4.2 실험

정확률과 재현율등을 측정은 각 그룹별로 <표 3>의 연관 질의어를 입력하여 나온 검색 결과를 바탕으로 하였다.

<표 3> 입력 질의어

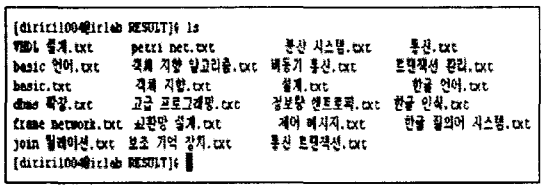
구분	질의	해당 그룹
q1	트랜잭션 관리	C
q2	분산 시스템	C
q3	제어 메시지	D
q4	비동기 통신	D
q5	DBMS 확장	H
q6	객체 지향 알고리즘	H

확률모델의 성능 평가를 위하여 입력 질의어를 콘도르 정보검색시스템에 직접 입력하여 결과를 구했다. [그림 3]은 콘도르의 질의어 입력 후 검색결과 화면이다.



[그림 3] 콘도르 검색결과 화면

확률모델의 성능 평가는 각 입력 질의어를 콘도르 정보검색시스템에 직접 입력하면, 시스템 내에 [그림 4]와 같이 검색 결과 문서 번호가 (질의어).txt로 생성되도록 하여 성능 측정 자료로 활용하였다.

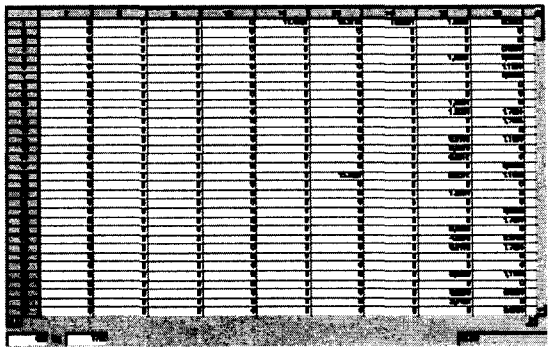


[그림 4] (질의어).txt 파일

벡터모델의 성능 평가는 400개의 문서와 7782개의 단어로 이루어진 벡터모델 공간을 구축하여 이루어졌다. term-by-document의 전체 집합 행렬 A를 생성하고 용어의 빈도수(tf)와 역문서빈도수(idf)값을 고려한 가중치 값을 적용하여 행렬 $A_{weighted}$를 구성하였다. [7]

[그림 5]는 각 용어의 가중치로 구성된 행렬 $A_{weighted}$에 대한 구현화면이다. $A_{weighted}$행렬은 LSI 개념을 적용하기 위하여 SVD 분해 되는데 이때, SVD 분해를 위해서는 Bioscience Division, Los Alamos National Laboratory의 Michael E. Wall등이 개발한 SVDMAN(Singular Value Decomposition Microarray ANalysis)을 사용하였다. [9]

행렬 A를 SVD 분해하면, term의 수가 7782개, document의 수가 400개 이므로, U 행렬은 size가 7782 × 7782 인 행렬, Σ 행렬은 size가 7782 × 400 인 행렬, V^T 행렬은 size가 400 × 400 인 행렬로 구해



[그림 5] 가중치가 적용된 weighted_A 행렬

진다. 여기에서 U 행렬은 단어와 단어간의 관계를 나타내며, Σ 행렬은 단어와 문서와의 관계, V^T 행렬은 문서와 문서간의 관계를 나타낸다.

마찬가지로 사용자 질의어에 사전적인 의미를 추가하여 확장한 질의어를 입력한 경우의 검색 성능 평가를 위한 실험도 동일한 환경에서 진행하였다. 다만, 콘도르 시스템에는 질의어의 사전 의미로서 추가되는 단어들을 입력하였고, $A_{weighted}$ 행렬에 적용하는 확장 질의벡터는 추가된 단어의 가중치를 구하여 질의에 적용함으로써 원 질의 벡터의 값을 갱신하도록 하였다.

4.3 실험 결과

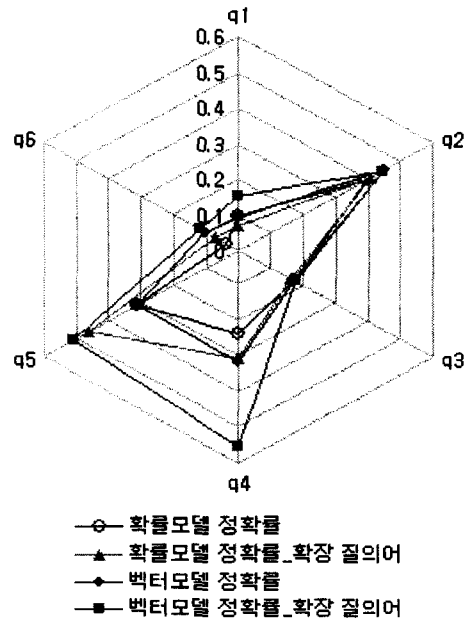
콘도르 정보검색시스템과 LSI 벡터모델 검색결과를 바탕으로 정확률과 재현율을 구하면 <표 4>와 같다. [2] <표 4>는 동일한 데이터 집합(KT-SET 93)을 대상으로 확률모델에 원 질의어를 입력한 검색결과와, 질의어의 사전적 의미를 갖는 용어를 추가하여 확장

<표 4> 검색 결과

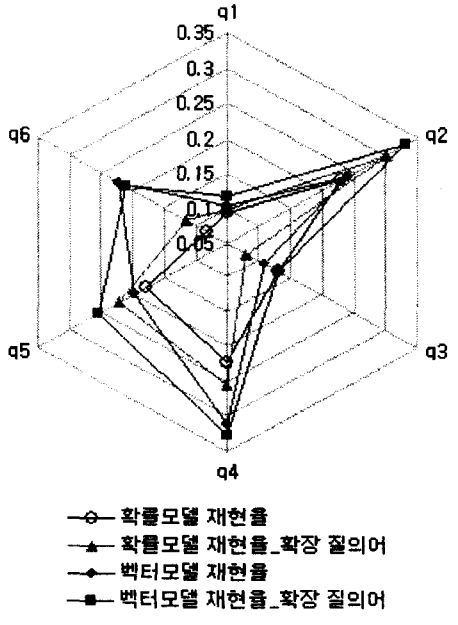
	질의	일반 질의		확장 질의	
	구분	정확률	재현율	정확률	재현율
확률모델 (콘도르)	q1	0.0972	0.0933	0.0689	0.0998
	q2	0.3947	0.2293	0.4083	0.3012
	q3	0.1724	0.1311	0.1633	0.0800
	q4	0.2356	0.2231	0.3060	0.2545
	q5	0.3111	0.1782	0.4598	0.2231
	q6	0.0374	0.0828	0.0697	0.1154
벡터모델 (LSI)	q1	0.0916	0.1035	0.1554	0.1176
	q2	0.4500	0.2409	0.4457	0.3323
	q3	0.1744	0.1082	0.1760	0.1324
	q4	0.3158	0.3101	0.5512	0.3265
	q5	0.3122	0.1988	0.5122	0.2557
	q6	0.1015	0.2227	0.1159	0.2113

질의어를 입력한 검색결과들을 벡터모델의 검색결과와 비교한 것이다.

<표 4>의 결과가 [그림 6]과 [그림 7]에 도시되어 있다. [그림 6]은 각 모델의 정확률을 비교한 것을 나타낸다. [그림 7]은 각 모델의 재현율을 비교한 것이다.



[그림 6] 정확률 비교



[그림 7] 재현율 비교

[그림 6]과 [그림 7]에서 보이는 것처럼, 벡터모델의 경우 대부분 모든 경우에서 뛰어난 성능을 보인다. 특히, 확장질의어를 사용한 벡터모델의 경우 재현율과 정확률 모두에서 제일 외곽선을 차지하고 있다.

벡터모델은 확률모델에 입력한 원 질의어의 검색결과에 비하여 평균적으로 정확률과 재현율에서 약 30% 이상의 성능 향상을 보이고 있다. 질의어를 확장한 검색 결과에서는 벡터모델이 확률모델보다 정확률에서 약 50%, 재현율에서 약 30% 이상까지 성능 향상을 보이고 있다.

5. 결 론

본 연구에서는 상용화 중에 있는 정보검색시스템 '콘도르'의 연구용 버전을 확률모델로 사용하여 실험하였다. 벡터모델의 구현은 LSI모델의 SVD 분해 기법을 이용하여 벡터계산을 단순화 시켰다. 각 모델의 검색 성능을 비교하기 위하여 동일한 데이터 집합(KT-SET93)을 각 모델 특성에 맞게 처리한 후 시스템에 적용하여 검색 성능을 측정하였다. 실험에는 원 질의어에 대한 검색 성능과 더불어, 질의어에 대한 사전적인 관련 의미를 추가하여 잠재적 연관성까지 고려한 검색 성능을 비교하였다.

원 질의어에 대한 검색 성능을 각 모델에서 비교하였을 때, 벡터모델의 성능이 비교적 우수했다. 그러나 사전 의미를 고려하여 질의어를 확장한 후 얻은 검색 결과는 벡터모델에서 두드러지게 향상됨을 보였다. 이러한 실험결과로부터 사용한 벡터모델인 LSI모델은 문서 간, 또는 단어간의 잠재적인 의미를 고려한 검색 결과를 도출해 내는데 적절한 시스템임이 확인되었다.

향후 연구과제는 원 질의어에 대한 사전적 의미뿐만 아니라 동의어를 추가함으로써 얻어지는 결과를 비교하여 정보검색시스템의 성능을 향상시키고자 한다.

참 고 문 헌

[1] http://www.kordic.re.kr/~news/letter/15/nl05.htm

[2] Richardo Baeza-Yates, Berthier Ribeiro-Neto, 'Modern Information Retrieval',Addison Wesley, pp. 36-46, 1999.

[3] William B.Frankes, Ricardo Baeza-Yates 원저, 류근호 외 공역, "정보검색", PRENTICE HALL, 시그마프레스 1995.07.
[4] W. Bruce Croft, "Advances in Information Retrieval", Kluwer Academic Publishers, pp. 60-66, 206, 2000.
[5] Richardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval. Addison Wesley, 1999.
[6] 고지현, 오형진, 박순철, "LSI를 이용한 가중치 변화에 따른 클러스터링 결과 분석", 정보처리학회지, 제 9권, 제 2호, pp. 1009-1012, 2002
[7] Gerald J. Kowalski, Mark T. Maybury, Information Storage And Retrieval Systems. Kluwer Academic Publishers, 2000.
[8] Michael W. Berry, Susan T. Dumais, Todd A. Letsche, "Computation Methods For Intelligent Information Access" ACM, 1995.
[9] http://public.lanl.gov/svdman/

전 유 정 (Yu-jung Jeon)

2002년 전북대학교 정보통신공학과 공학사
2004년 전북대학교 정보통신공학과 석사

(관심분야 : 정보검색, 멀티미디어 정보검색, 데이터베이스)

변 동 률 (Dong-ryul Byeon)

1998년 전북대학교 공학사
2003년 전북대학교 정보통신석사
2001년-2002년 미국 카네기멜론대학 언어기술연구소 방문연구

(관심분야 : 정보검색, 멀티미디어 정보검색, 데이터베이스)

박 순 철 (Soon-cheol Park)

1979년 인하대학교 공학사
1991년 미국 루이지아나주립대학 전산학박사
1993년-현재 전북대학교 전자정보공학부 부교수

(관심분야 : 정보검색, 알고리즘)