벡터와 신경망 모델에서 데이터 퓨전 기법을 이용한 정보검색의 효율성 향상

Improving the Effectiveness of Information Retrieval Using Data Fusion Method in the Vector and Neural Network Model

최성환(연세대학교 대학원 문헌정보학과)

Choi Sung-Hwan

Dept. of Library and Information Science, Graduate School of Yonsei University

본 논문에서는 벡터모델과 신경망 모델을 이용하여 데이터 퓨전의 관점에서 다중증거로서 가중치, 문헌분리가, 엔트로피, 공기유사도를 적절히 결합하여 질의를 확장하는 방법을 제안한다. 실험결과 코사인 정규화 가중치 알고리즘, 문서길이 정규화 가중치 알고리즘과 결합하여 질의를 확장 하는 것이 정규화시키지 않고 단순히 문헌빈도와 역문헌빈도의 조합을 이용한 가중치 알고리즘과 결합했을 때 보다 평균정확률 향상이 더 높게 나타났다. 또한 다양한 공기기반 유사도를 이용하여 질의확장을 한 결과 벡터모델과 신경망 모델에서 코사인 공기유사도에 기반하여 질의확장한 경우가 다른 공기유사도에 비해 더 좋은 성능을 보였다.

1. 서 론

최근에는 정보검색의 효율성을 높이기 위해 다중 문서(질의) 표현 방법이나 다중 검색기법들을 결합하는 '데이터 퓨전(data fusion)' 연구가 활발히 이루어지고 있다(Belkin etal, 1995).

데이터 퓨전 기법 연구의 당위성은 1980년대 이래 경험적·실험적인 연구결과들에서 그 근거를 찾을수 있다. 정보검색에서 다중증거 결합 연구의 필요성은 첫째, 주제형성은 사람(시스템)에 따라 달리 해석되기 때문에 단일표현이나 검색모델로 이용자의 정보요구를 완전하게 표현하기 어렵다. 특히 정보검색에서 하나의 질의에 대해서도 검색모델과 가중치 기법에 따라 다른 문서들을 검색하게 되는데 이는 다중증거들을 적절히 결합함으로써 적합문서들을 효과적으로 검색할 수 있다. 둘째, 질의와 문서간 하나의 적합한 증거만을 가지고는 적합한 문서들을 효과적으로 검색하기가 어렵다. 따라서 질의와 문서들간의 증거들이 많으면 많을수록 확률적으로 적합한 문서들을 많이 검색할 수 있다는 논리적 근거를 바탕으로 적합확률을 높일 필요가 있다. 셋째, 논리적으로 단일증거는 검색공간을 협소하게 만들 가능성이 있다. 그러므로 다양한 질의표현과 문서표현들을 이용하여 검색공간을 확장하고 재현율을 향상시키는 동시에 적절하게 이들 증거들을 결합함으로써 정확률을 향상시킬 필요가 있다.

지금까지 다중증거 결합을 이용하여 정보검색의 효율성을 향상시키는 방안이 많이 제안되어 왔으나, 가중치 퓨전을 통한 다양한 공기유사도에 기반하여 질의를 확장시키는 연구는 거의 이루어지지 않았다. 본 논문에서는 데이터 퓨전기법의 관점에서 검색효율성을 높이기 위해 가중치, 문헌분리가, 엔트로피,

문헌빈도, 공기유사도를 다중증거로서 적절히 결합하여 질의를 확장하는 방법을 제안한다.

2. 알고리즘

2.1 검색 모델

벡터공간모델(Vector Space Model)은 질의벡터와 문서벡터간의 유사도를 계산하는데 일반적으로 코사인 유사도 계수를 많이 이용하여 왔다. 이 모델은 많은 순위부여 검색실험에서 기본모델로 사용되었고, 특히 설튼(Salton)과 그의 동료가 스마트(SMART) 검색시스템 실험에서 채택한 모형이다. 본 논문에서는 다양한 유사도를 이용하여 실험한 결과 <수식1>과 같은 내적(inner product)를 이용한 것이 좋은 성능을 보여 질의와 문서간의 유사도 모델로 내적유사도를 이용하였다.

$$(d_j, q_k) = \sum_{i=1}^{n} (td_{ij} \times tq_{ik}) \qquad (1)$$

where td_{ij} :문헌벡터 j에서 i번째 용어

tq_{ik} :질의벡터 k에서 i번째 용어

n :고유단어수

신경망(Neural network)은 생물학적 뉴런의 구조와 기능을 단순화하여 수학적 모델로 표시하고 이 뉴런모델을 상호 연결시켜 망을 형성한 것을 말하며, 생물학적 신경망과 구별하여 인공 신경망(Artificial neural network)이라 하기도 한다. 신경망은 기본적으로 입력(Inputs), 연결선(weight), 결합함수(combining function), 전이함수(transfer function), 출력(outputs)의 5개 부분으로 구성된다. 결합함수는 입력되는 값과 연결선의 정보를 결합하여 전이함수로 보내게 되고, 전이함수는 결합함수로부터 나온 결과를 변환하여 현재 뉴론의 출력값을 결정한다. 본 논문에서는 일반적으로 많이 사용되는 결합함수로 누적합을 이용하였으며, 전이함수로 <수식2>와 같은 비선형적인 시그모이드(sigmoid) 함수를 이용하였다.

$$out = 1/(1+\exp(-anet+b)) \qquad (2)$$

where $a=1, b=0$

2.2 퓨전 연산

벡터모델에는 <수식11>, 신경망모델에는 <수식21>를 각 가중치와 결합하여 실험을 하였다. 질의확장에서 가중치 퓨전 연산은 기존의 가중치 결합방식과는 달리 고빈도의 영향을 줄이기 위해서 가중치를 결합할 때 문헌분리가와 엔트로피(정영미, 1993)가 모두 음수값을 가지면 질의와 문서간의 관계에 대한 증거로서 가치가 없다고 가정하고 가중치를 결합하지 않았다. 이것은 특히 수작업으로 만들어진 키워드 질의가 아닌 자연어 질의를 이용할 때 유효한 실험결과를 얻었다. 그리고 저빈도의 영향을 줄이기 위해 문헌빈도가 3이상인 용어만 가중치를 결합하였다.

2.3 공기기반 유사도

공기란 두 용어가 동일문서 혹은 몇 개의 단어 창안에서 같이 발생하는 것을 말하는데, 공기빈도수가 클수록 두용어가 밀접한 관련이 있다는 전제에 기반한다(Rijsbergen, 1977). 본 논문에서는 공기단위를 동일문서로 하였다.

<표 1> 공기기반 유사도

df_x : 용어 x를 포함한 문서수, df_y : 용어 y를 포함한 문서수, df_{xy} : 용어 x,y를 동시에 포함한 문서수, N : 총문서수, d_{ik} : 문서 d_k에서 용어 t_i의 색인가중치, d_{jk} : 문서 d_k에서 용어 t_j의 색인가중치(수식11 이용)

(3) $sim(x,y) = \frac{df_{xy}}{\sqrt{df_x \cdot df_y}}$	코사인(cos)
(4) $sim(x,y) = \frac{2df_{xy}}{df_x + df_y}$	다이스(dice)
(5) $sim(x,y) = \frac{df_{xy}}{df_x + df_y - df_{xy}}$	자카드(jac)
(6) $sim(x,y) = \log \frac{N \times df_{xy}}{df_x \times df_y} / \log N$	정규상호정보(nmi)
(7) $sim(x,y) = \frac{df_{xy}}{2}(\frac{1}{df_x} + \frac{1}{df_y})$	평균조건확률(acp)
(8) $sim(t_i,t_j) = \sum_{k=1}^{N} d_{ik} \cdot d_{jk}$	용어간유사도(wsim)

2.4 가중치

정보검색시스템에서 질의 혹은 문서에 순위를 부여할 수 있는데, 대표적인 방법으로 문서의 구성요소들을 조합하는 가중치를 사용하여 순위를 부여할 수 있다. 설튼과 버클리(Buckley)는 과거 20여 년 동안 스마트 검색시스템에서 행한 실험을 통해 문헌내 용어빈도수와 코사인 측정법에 의해 정규화된 역문헌빈도수를 조합하면 가장 좋은 문헌 용어가중치를 만들어 낼 수 있다는 것과 용어가중치를 갖는 질의에 대해서 개선된 질의용어가중치 부여방법을 사용하면 성능을 향상시킬 수 있다는 연구결과를 발표했다. 질의용어나 문헌용어의 가중치를 계산하는 방법은 그동안 많은 연구에서 제안되었고, 코사인 정규화를 많이 사용하고 있다. 최근 연구결과에서는 스마트의 코사인정규화는 Inquery의 최대단어빈도정규화와 Okapi의 바이트길이 정규화에 비해 낮은 검색효과를 보였으며, 코사인 정규화를 수정한 피벗코사인 정규화로 성능을 향상시킬 수 있다고 보고되었다. 본 논문에서는 색인어 가중치 부여 기법의 구성요소로 출현빈도, 장서빈도, 문헌빈도, 정규화 등을 조합한 가중치 알고리즘 중에서 일반적으로 많이 사용되는 것을 이용하였다.

<표 2> 가중치 알고리즘

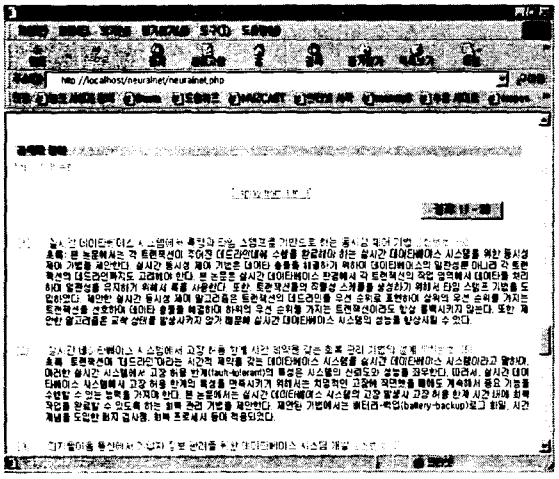
tf : 문헌(혹은 질의)내 용어 t의 출현빈도수, cf : 전체 컬렉션에서 용어 t의 총출현 빈도수
$tfnum$: 문헌(혹은 질의)내 유니크한 용어수, N : 총문헌수, n : 용어 t가 출현하는 문헌수
dl : 문헌길이, $avdl$: 평균문헌길이, $a=0.5$, $cutoff=\max df\times 0.8$

(9)	$tf\cdot\log\frac{N}{n}$	ntn	(10)	$\log(tf+1.0)\times\log\frac{N}{n}$	ltn
(11)	$\frac{\log(tf+1)\times(\log\frac{N}{n})}{\log tfnum}$	lts	(12)	$\frac{(cf/n)^a}{\log\max(cutoff,n)}$	avtf
(13)	$0.5+0.5\frac{tf}{\max tf}\times\log\frac{N}{n}$	atn	(14)	$\frac{1+\log(tf)}{1+\log(total\ tf)}\times\frac{\log(\frac{N}{n})}{\max idf}$	ltf
(15)	$\frac{tf\cdot\log\frac{N}{n}}{\sqrt{\sum(tf\cdot\log\frac{N}{n})^2}}$	ntc	(16)	$\frac{\log(tf)+1.0}{\sqrt{\sum(\log tf+1.0)^2}}$	lnc
(17)	$\frac{0.5+0.5\frac{tf}{\max tf}}{\sqrt{\sum(0.5+0.5\frac{tf}{\max tf})^2}}$	anc	(18)	$\frac{0.5+0.5\frac{tf}{\max tf}\times\log\frac{N}{n}}{\sqrt{\sum(0.5+0.5\frac{tf}{\max tf})^2\sum(\log\frac{N}{n})^2}}$	atc
(19)	$\frac{1+\log(1+\log tf)}{0.8+0.2\cdot\frac{dl(in\ byte)}{avdl(in\ byte)}}$	dnb	(20)	$\frac{\left[\frac{1+\log tf}{1+\log(avgtf)}\right]\times(\log\frac{N}{n}+1)}{0.8+0.2\cdot\frac{dl(in\ unique\ terms)}{avdl(in\ unique\ terms)}}$	dtus
(21)	$\frac{(1.0+\log tf)\cdot(\log\frac{N}{n})}{(1.0-slope)\times pivot+slope\cdot unique\ terms}$				ltu
(22)	$0.4+0.6\times\frac{tf}{tf+0.5+1.5\frac{dl}{avdl}}\times\frac{\log(\frac{N+0.5}{n})}{\log(N+1.0)}$				btu

3. 실험환경 및 성능평가

3.1 실험환경

실험집단은 KTSET 1000건을 대상으로 초록, 저자, 키워드를 추출하여 HAM으로 색인을 생성하였다. 데이터베이스로는 Mysql를 이용하였으며, 웹프로그래밍 언어인 PHP로 웹상에서 실시간으로 검색이 가능하도록 구현하였다. 실험은 구현된 검색시스템을 가지고 30개의 자연어 질의어로 벡터모델과 신경망 모델을 이용하여 실험하였다. 구현된 시스템은 검색된 문헌의 타이틀을 클릭하면 직접 원문에 접근이 가능하도록 하였다. 검색결과는 서지사항인 타이틀, 초록을 제시한다.



<그림 1> 검색시스템 인터페이스

3.2 성능평가

유사도 측정에 의한 검색에서는 불리안 검색과는 달리 질문과 문헌과의 유사도를 나타내는 유사계수를 산출하여 유사계수의 값이 큰 문헌부터 작은 문헌 순으로 출력하게 된다. 이러한 검색시스템에서는 흔히 유사계수의 값에 의해 결정되는 문헌의 순위가 검색기준치가 된다. 실제 검색에서 재현율 값이 0.1, 0.2와 같이 표준적인 값이 아닐수 있기 때문에 일반적으로 보간법을 이용한다. 보간법은 재현율의 값을 먼저 표준 재현율로 바꾸고 이에 대한 정확률의 값을 산출하는 방법으로 정확률의 산출방법은 크게 선형보간법과 비관적 보간법으로 나뉜다. 본 실험에서는 '비관적' 보간법에 의해 정확률을 산출하였다.

4. 실험결과 및 분석

4.1 질의처리와 가중치 퓨전

정보검색에서 간과하기 쉬운 것이 질의처리이다. 그러나 질의처리를 어떻게 하느냐에 따라 문서벡터와 질의벡터간의 매칭에 많은 영향을 미치기 때문에 곧 검색성능에 영향을 미치게 된다. 본 논문에서는 자연어 질의를 Linux용 HAM5.0을 이용하여 입력된 질의를 분석하여 질의를 처리한다. KTSET 테스트 컬렉션에서 제공되는 불리언 질의에서 연산자를 제외한 키워드를 추출하여 단순히 공백으로 각 키워드를 분리하여 검색한 결과보다 키워드를 형태소 분석기로 처리하여 질의하는 것이 벡터모델과 신경망 모델 모두에서 더 높은 성능향상을 보였다. 일례로 단순한 TFIDF 조합인 ntn의 경우 키워드를 형태소로 분석한 것이 키워드를 공백으로 분리한 경우보다 벡터공간 모델에서 11-포인트 평균 정확률이 18.6%(0.4614), 신경망 모델에서 19.1%(0.463)가 향상되었다. 그리고 키워드 질의를 형태소로 분석한 경우가 자연어 질의를 형태소로 분석하는 것보다 더 좋은 성능을 보였다. 가중치 퓨전은 벡터모델은 <수식11>, 신경망 모델은 <수식21>을 각 가중치 알고리즘과 결합하였다. 퓨전연산은 단순히 가중치를 결합하는 것이 아니라 자연어 질의어가 검색시스템에 입력되면 실시간으로 형태소분석을 통해 불용어제거와 복합명사를 분해하여 분석된 질의어 중에서 결합에 좋은 질의어를 선정하기 위해 문헌분리가와 엔트로피를 이용한다. 문헌분리가와 엔트로피를 색인어 가중치로 이용하고자 할 때는 여기에 문헌내 용어빈도를 곱하여 색인어 가중치로 이용한다. 그러나 기존의 연구결과 이론적으로는 매우 훌륭하나 실제로 실험한 결과들은 좋은 결과를 얻지 못했다. 본 논문에서는 가중치를 결합할 때 직접적으로 이들을 색인어 가중치로 이용하지 않고 음수값을 가지면 가중치를 결합시키지 않는 결합제약조건으로만 문헌분리가와 엔트로피를

<표 3> 가중치 퓨전을 통한 공기유사도로 질의확장한 11-포인트 평균정확률

() : 질의확장없이 단일가중치의 평균정확률과 비교한 증감율
{} : 질의확장없이 가중치를 결합한 평균정확률과 비교한 증감율
[] : 단일가중치로 질의확장한 평균정확률과 비교한 증감율
vsm : 벡터공간 모델, nn : 신경망 모델

유사도 / 가중치	cosign		dice		jaccard		nmi		acp		wsim	
	vsm	nn	vsm	nn	vsm	nn	vsm	nn	vsm	nn	vsm	nn
ntn	0.4686	0.4808	0.4685	0.4838	0.4517	0.4753	0.4716	0.4831	0.4669	0.4828	0.454	0.4757
	(5.6)	(7.8)	(5.5)	(8.5)	(1.8)	(6.5)	(6.2)	(8.3)	(5.2)	(8.2)	(2.3)	(6.6)
	{4.4}	{0.7}	{4.4}	{1.3}	{0.6}	{-0.4}	{5.1}	{1.2}	{4.0}	{1.1}	{1.2}	{-0.4}
	[1.6]	[4.2]	[1.8]	[5.1]	[1.0]	[6.2]	[0.8]	[3.3]	[1.3]	[4.8]	[1.8]	[6.7]
ltn	0.4895	0.4981	0.4883	0.4972	0.484	0.4907	0.4759	0.4948	0.4573	0.4846	0.484	0.4914
	(2.5)	(4.4)	(2.3)	(4.3)	(1.4)	(2.9)	(-0.3)	(3.8)	(-4.2)	(1.6)	(1.4)	(3.0)
	{1.5}	{1.7}	{1.2}	{1.5}	{0.3}	{0.1}	{-1.3}	{1.0}	{-5.2}	{-1.1}	{0.3}	{0.3}
	[3.2]	[5.1]	[1.6]	[3.5]	[0.7]	[2.1]	[0.7]	[4.7]	[0.9]	[6.9]	[0.6]	[2.1]
avtf	0.5013	0.5037	0.5028	0.5041	0.4975	0.4965	0.5006	0.4996	0.4879	0.4944	0.494	0.4959
	(15.5)	(16.0)	(15.9)	(16.1)	(14.7)	(14.4)	(15.4)	(15.1)	(12.4)	(13.9)	(13.9)	(14.2)
	{0.6}	{1.6}	{0.9}	{1.7}	{-0.1}	{0.1}	{0.5}	{0.8}	{-2.0}	{-0.3}	{-0.8}	{0.0}
	[18.9]	[19.5]	[18.1]	[18.4]	[15.0]	[14.8]	[11.1]	[10.8]	[17.0]	[18.6]	[19.8]	[20.2]
atn	0.4773	0.5044	0.4793	0.4991	0.4791	0.4943	0.4715	0.4899	0.4554	0.4881	0.4797	0.4948
	(5.6)	(11.2)	(6.0)	(10.1)	(6.0)	(9.0)	(4.3)	(8.1)	(0.8)	(7.7)	(6.1)	(9.1)
	{0.1}	{1.8}	{0.6}	{0.7}	{0.5}	{-0.2}	{-1.1}	{-1.1}	{-4.4}	{-1.5}	{0.7}	{-0.1}
	[6.9]	[12.9]	[5.3]	[9.7]	[6.7]	[10.1]	[3.9]	[8.0]	[6.7]	[14.3]	[7.7]	[11.1]
ltf	0.5059	0.5036	0.5055	0.5043	0.4998	0.4979	0.5043	0.5006	0.495	0.4946	0.4978	0.4959
	(2.7)	(2.4)	(2.6)	(2.6)	(1.5)	(1.3)	(2.4)	(1.8)	(0.5)	(0.6)	(1.1)	(0.9)
	{1.1}	{1.5}	{1.0}	{1.7}	{-0.1}	{0.4}	{0.8}	{0.9}	{-1.1}	{-0.3}	{-0.5}	{-0.0}
	[0.8]	[0.4]	[0.4]	[0.1]	[1.3]	[1.0]	[1.3]	[0.6]	[1.0]	[1.0]	[0.4]	[0.1]
lnc	0.5052	0.5043	0.5066	0.5041	0.4977	0.4978	0.5037	0.5005	0.4974	0.4945	0.5017	0.496
	(8.8)	(8.8)	(9.1)	(8.8)	(7.2)	(7.4)	(8.5)	(8.0)	(7.1)	(6.7)	(8.1)	(7.0)
	{1.0}	{1.7}	{1.3}	{1.6}	{-0.5}	{0.3}	{0.7}	{0.9}	{-0.5}	{-0.3}	{0.3}	{-0.0}
	[6.5]	[6.3]	[7.6]	[7.1]	[9.0]	[9.0]	[3.7]	[3.0]	[2.4]	[1.8]	[5.6]	[4.4]
ntc	0.5066	0.5039	0.5081	0.5048	0.4988	0.4981	0.5071	0.5009	0.5038	0.4936	0.5043	0.4955
	(11.1)	(10.5)	(11.4)	(10.7)	(9.4)	(9.2)	(11.2)	(9.8)	(10.5)	(8.2)	(10.6)	(8.6)
	{1.3}	{1.5}	{1.6}	{1.7}	{-0.2}	{0.3}	{1.4}	{0.9}	{0.8}	{-0.6}	{0.9}	{-0.2}
	[4.3]	[3.7]	[7.0]	[6.3]	[8.8]	[8.7]	[2.8]	[1.5]	[1.2]	[-0.8]	[6.6]	[4.7]
anc	0.5041	0.5038	0.5037	0.5041	0.4985	0.4979	0.503	0.4998	0.495	0.4941	0.4993	0.4959
	(11.4)	(11.5)	(11.3)	(11.6)	(10.1)	(10.2)	(11.1)	(10.6)	(9.3)	(9.4)	(10.3)	(9.8)
	{1.1}	{1.6}	{1.0}	{1.7}	{-0.1}	{0.4}	{0.8}	{0.8}	{-0.8}	{-0.4}	{0.1}	{0.0}
	[7.4]	[7.3]	[10.3]	[10.4]	[11.5]	[11.4]	[7.6]	[6.9]	[6.9]	[6.7]	[10.1]	[9.3]
atc	0.5059	0.5039	0.5063	0.5047	0.4995	0.4978	0.5041	0.5006	0.4962	0.4944	0.4982	0.4958
	(11.8)	(11.3)	(11.9)	(11.5)	(10.4)	(9.9)	(11.4)	(10.6)	(9.7)	(9.2)	(10.1)	(9.5)
	{1.2}	{1.5}	{1.3}	{1.7}	{-0.1}	{0.3}	{0.9}	{0.8}	{-0.7}	{-0.4}	{-0.3}	{-0.1}
	[6.8]	[6.4]	[10.1]	[9.7]	[10.5]	[10.1]	[2.6]	[1.9]	[1.3]	[1.0]	[8.0]	[7.4]
dnb	0.4856	0.499	0.487	0.501	0.4788	0.4921	0.4899	0.4962	0.4827	0.4903	0.4815	0.4928
	(6.0)	(9.1)	(6.3)	(9.6)	(4.5)	(7.6)	(6.9)	(8.5)	(5.3)	(7.2)	(5.1)	(7.8)
	{1.1}	{1.2}	{1.4}	{1.6}	{-0.3}	{-0.2}	{2.0}	{0.6}	{0.5}	{-0.5}	{0.3}	{-0.0}
	[4.7]	[7.6]	[5.8]	[8.8]	[5.6]	[8.6]	[3.2]	[4.5]	[6.6]	[8.3]	[4.4]	[6.9]
dtus	0.5073	0.5039	0.5039	0.5035	0.4974	0.4974	0.4997	0.4999	0.4964	0.4939	0.4973	0.496
	(5.0)	(3.9)	(4.3)	(3.8)	(2.9)	(2.5)	(3.4)	(3.0)	(2.7)	(1.8)	(2.9)	(2.2)
	{1.4}	{1.6}	{0.7}	{1.5}	{-0.6}	{0.3}	{-0.2}	{0.8}	{-0.8}	{-0.4}	{-0.6}	{-0.0}
	[4.1]	[3.4]	[2.4]	[2.3]	[2.9]	[2.9]	[1.8]	[1.9]	[3.8]	[3.3]	[2.5]	[2.2]
btu	0.502	0.5034	0.501	0.5027	0.4943	0.4972	0.4958	0.4997	0.4885	0.4945	0.4955	0.4951
	(17.1)	(17.5)	(16.9)	(17.3)	(15.3)	(16.1)	(15.7)	(16.6)	(13.9)	(15.4)	(15.6)	(15.6)
	{1.3}	{1.7}	{1.1}	{1.5}	{-0.2}	{0.4}	{0.1}	{0.9}	{-1.4}	{-0.1}	{0.0}	{0.0}
	[15.9]	[16.2]	[18.0]	[18.4]	[17.1]	[17.8]	[19.9]	[20.8]	[17.7]	[19.1]	[14.0]	[13.9]

이용하였다. 이것은 유전자 알고리즘에서 우성인자가 열성인자보다 더 많은 자손을 낳도록 하는 것과 유사한 기법이다. 실험결과 문헌분리가와 엔트로피를 결합제약조건으로 이용할때가 이용하지 않은 경우보다 더 좋은 성능을 보였다. 가중치 퓨전 실험결과, 코사인 정규화 기법을 이용한 atc의 경우 단일가중치 알고리즘의 평균정확률과 비교하면, 벡터모델에서 11-포인트가 10.5%(0.4998), 3-포인트가 8.7%(0.4930), 순위 30위에서 평균정확률이 0.3932로 성능향상이 있었으며, 신경망 모델에서 11-포인트가 9.6%(0.4964), 3-포인트가 10.0%(0.4938), 순위 30위에서 평균정확률이 0.3895로 성능이 향상되었다.

4.2 질의확장

<표3>에서 알수 있듯이 검색모델보다는 어떤 가중치 알고리즘과 결합하느냐에 따라 성능향상이 차이를 보이고 있다. 문서용어 가중치로 코사인 정규화한 ntc, atc, anc 가중치 알고리즘과 결합하여 공기유사도에 기반한 질의확장이 다른 가중치 알고리즘 보다 평균정확률의 향상이 더 높게 나타났다. 특히, 전반적으로 단순 TFIDF(ntn, ltn) 및 출현빈도를 최대출현빈도로 정규화한(atn) 가중치 알고리즘 보다 코사인 정규화(ntc, atc, anc)와 문서길이를 정규화(btu, dtus)한 가중치 알고리즘과 결합시켰을 때 11-포인트 평균정확률의 증가율이 높게 나타나는 실험결과를 얻었다. 가중치 퓨전기법을 통한 질의확장에서 좋은 성능을 보이고 있는 atc의 경우, 공기유사도(cosign)에서 단일가중치의 평균정확률보다 11.8%(0.5059), 질의확장을 하지 않고 가중치 퓨전만 한 것에 비해 1.2%, 가중치 퓨전 없이 단일가중치로 질의확장한 경우보다 6.8%의 성능향상을 보였다. 문서길이 정규화 기법를 이용한 dtus 가중치 알고리즘이 벡터모델의 경우, 질의확장을 하여 11-포인트가 0.5073으로 가장 좋은 성능을 보였으며, avtf(0.5013), ltf(0.5059), lnc(0.5052), ntc(0.5066), anc(0.5041), atc(0.5059), btu(0.502) 가중치 알고리즘이 좋은 성능을 보였고, 신경망 모델에서도 벡터모델과 비슷한 성능향상을 보여주었다. 그러나 <표3>에서 알수 있듯이 가중치 퓨전을 통한 공기기반 질의확장과 질의확장 없이 가중치 퓨전만을 이용한 경우와 비교해보면 커다란 성능향상은 없었으며, 오히려 11-포인트 평균정확률이 감소하는 경우도 많았다. 이것은 질의확장을 하지 않고도 단지 가중치만을 결합하여도 충분히 좋은 성능을 보일 수 있다는 것을 의미한다.

5. 결 론

현재의 검색시스템의 대부분은 이미 발표된 연구 문헌에 나타난 유용한 아이디어를 이용함으로써 상당한 개선이 가능하다. 최근에 기존 연구결과들을 분석하여 이러한 방법들을 결합하기 위한 연구들이 수행되었으며, 본 논문에서도 가중치, 문헌분리가, 엔트로피, 공기유사도의 다중증거 결합을 통해 질의확장을 하는 방법을 제시하였다. 가중치를 2-3개 결합할때는 시간복잡도가 크게 차이가 나지 않기 때문에 제안하는 방법은 기존 검색시스템에 쉽게 수용될 수 있다. 다양한 공기기반 유사도를 이용하여 질의확장을 한 결과 벡터모델과 신경망 모델에서 코사인 공기유사도를 이용하여 질의확장한 경우가 다른 공기유사도 보다 좋은 성능을 보였으며, 향후 이들 유사도 모델들을 결합하여 검색 효율성을 향상시키는 연구와 지식기반과 탐색결과에 근거한 질의확장을 결합하는 것 또한 연구과제가 될 것이다.

참 고 문 헌

정영미. 1993. 정보검색론. 서울:구미무역출판부.

이기호. 1999. 적합성 피드백 방법을 이용한 검색 효율의 향상. 충남대학교 컴퓨터공학과, 박사학위논문.

Belkin, N.J., Kantor, P., Fox, E.A., and Shaw, J.A., 1995. Combining the evidence of multiple query representations for information retrieval, Information Processing and Management, 31(3), pp. 431-448.

Rijsbergen, C.J.V., 1977. A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval, Journal of Documentation, Vol. 33, pp. 106-119.

Salton, G., and Buckley, C., 1988. Term weighting approaches in automatic text retrieval, Information Processing and Management, 24(5), pp. 513-523.