디지털 도서관에서 사용자 질의어와 컬렉션 사이의 관련성 분포정보를 이용한 컬렉션 융합

김 현 주† · 김 상 준†† · 배 종 민††† · 강 현 석†††

요 약

본 논문에서는 연합된 디지털 도서관에서 다양한 이질의 정보원으로부터 얻은 검색 결과를 효과적으로 융합하는 알고리즘을 제안한다. 제안된 융합 방법은 검색에 참여한 정보원으로부터 주어진 질의어에 대하여 같은 크기의 검색 문서 모집단을 추출한 후에, 질의어와 추출된 모집단간의 관련성 정도를 계산하고, 계산된 값을 해당 정보원에 대한 관련성 분포 정보로 사용하여 문서를 융합한다. 이때 추출된 문서를 융합하기 위해서, 컬렉션의 관련성 분포 정보, 모집단의 크기 N, 관련 문서의 순서 정보 등을 사용한다. 아울러 실험적으로 개발된 메타 검색기를 통하여 제안된 융합 알고리즘의 성능 평가 결과를 제시한다.

Collection Fusion using Relevance Distribution Information between Queries and Collections in Digital Libraries

Hyun-Ju Kim† · Sang-Jun Kim†† · Jong-Min Bae††† · Hyun-Syug Kang†††

ABSTRACT

This paper proposes an effective fusion algorithm for retrieval results from heterogeneous information sources in federated digital libraries. The algorithm determines the population of documents retrieved from involved information sources for a given query and evaluates the degree of relevance between the query and the population. The evaluated results are used as relevance distribution information for collection fusion. The main informations used for the fusion are relevance distribution among collections, the population size N, and ranking information of relevant documents in their origin. We also present the performance evaluation of the algorithm by developing the prototype of a meta-searcher.

1. 서 론

인터넷상에는 수많은 정보원이 있으며, 지금도 개발되고 있다. Yahoo, AltaVista, Excite 등 뿐만 아니라, 기술보고서, 학위논문 등을 저장한 학술용 디지털 도서관 등, 많은 정보원이 있다. 이러한 정보의 홍수 속에서 사용자들이 원하는 정보를 쉽게 얻을 수 있기가 점점 어려워지고 있다. 많은 정보원은 정보 검색 엔진을 갖추어서 사용자가 질의어 입력을 통해 효율적인 정보 검색이 가능하도록 도와주고 있다. 그러나 사용자는 수많은 디지털 도서관 중에서 어디에 자신이 원하는 정보가 있는지를 미리 알아야 하며, 또한 각 검

※ 이 논문은 1998년도 학술진흥재단의 공모과제 연구비에 의하여 연구되었음.
† 준 회 원 : 경상대학교 대학원 전자계산학과
†† 준 회 원 : 경상대학교 전자계산학과 석사과정
††† 종신회원 : 경상대학교 컴퓨터과학과 교수
논문접수 : 1999년 6월 7일, 심사완료 : 1999년 8월 25일

색 엔진마다 질의어 문서 모델이 다르기 때문에 더 정확한 정보를 빨리 검색하기 위해서 각 검색 엔진들에 대해 상당한 지식을 가지고 있어야 한다.

디지털 도서관 분야에서 이러한 어려움을 극복하고자 하는 노력중의 하나가 연합 검색(Federated search) 혹은 메타 검색(Meta search)이다[2, 3, 4]. 연합 검색 시스템은 사용자에게 분산된 이질의 내용을 가진 수많은 정보원들의 존재를 숨기고, 전체적으로 하나의 정보원만 있다는 관점을 제공한다. 따라서 사용자는 하나의 질의어만을 가지고 질의를 하게 되고, 그것을 연합 검색기가 받아서 사용자 요구를 만족시킬 수 있는 최선의 정보원들을 선택하여, 선택된 정보원에 대한 질의어로 자동으로 번역한 후, 각 정보원으로 질의를 대신하게 된다[8, 9, 11]. 또한 질의에 대한 검색 결과를 통합하고, 이를 순위 매김(ranking)한 후 사용자에게 결과를 보여준다.

그러나 서로 다른 이질의 정보원으로부터 검색 문서를 추출하고, 이들을 통합하여 단일 우선 순위를 가질 수 있도록 각 문서에 대한 순위 매김을 하기란 매우 어렵다. 그 이유는 각 정보원에서 제공되는 문서의 순위 매김 알고리즘은 일반적으로 잘 알려져 있지 않다. 게다가 설령 특정 두 개의 정보원이 같은 순위 매김 알고리즘을 사용한다 하더라도 같은 질의로부터 나온 검색 결과에 대해서도 문서의 순위 매김 값을 상대적으로 비교할 수 없다. 그 이유는 이들 정보원이 가지고 있는 전체 문서의 집합이 틀리기 때문에 같은 순위 매김 알고리즘을 사용한다 하더라도 같은 질의에 대하여 같은 순위 값이 나올 수 없기 때문이다. 이러한 연합 검색 시스템 분야에서는 분산되어 있는 다양한 컬렉션에서 얻어진 검색 결과들을 단일 검색 결과로 통합하는 방법이 부각되었다. 이러한 문제를 해결하기 위해 연구하는 분야를 컬렉션 융합이라고 한다[1, 2, 3, 5, 7].

본 논문에서는 관련성 분포 정보를 이용하여 컬렉션을 융합하는 알고리즘을 제안하고, 이 알고리즘의 성능을 평가하기 위해 실험적으로 구현한 HoleInOne (=*w*HOLE IN*formation* ONE*time*) 메타 검색기를 사용하였다. 제안된 융합 알고리즘에 대하여 성능을 평가하는 항목으로는, 관련있는 문서의 수, 관련없는 문서의 수, 중복된 문서의 수, 빈 URL의 수, 유일하게 관련된 문서의 수, 정확도 등 6가지이다. 그리고 성능 평가를 위해 사용한 질의어는 뉴스 그룹에서 분류한 13개의 주제어를 사용하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 컬렉션 융합에 관한 관련 연구들을 살펴보고, 3장에서는 본 논문에서 제안한 컬렉션 융합 방법을 소개한다. 그리고 4장에서는 3장에서 제안한 컬렉션 융합의 알고리즘을 소개하고, 5장에서는 제안된 알고리즘을 HoleInOne에서 평가한 실험 결과를 살펴본다. 마지막으로 6장에서 결론 및 향후 연구 과제를 살펴본다.

2. 관련 연구

지금까지 컬렉션 융합에 관한 연구는 크게 각 컬렉션에 관한 정보를 분산 환경에서 이용하는 방법과 컬렉션에 관한 정보를 이용하지 않는 방법이 있다.

컬렉션 정보를 이용하는 경우에는 두 가지로 나눌 수 있다. 첫째 색인, IDF(Inverse Document Frequency) 값 등의 컬렉션 정보를 중앙의 한 서버에서 관리하는 방법이다[1, 3]. 이 방법은 중앙의 서버에서 검색에 참여할 정보원의 색인 정보를 모두 관리하여, 질의에 대하여 적합한 정보원을 선택하는 것이 빠르게 결정될 수 있는 장점을 갖는다. 그런데 이러한 중앙 집중식 방법은 중앙의 서버가 연합 검색에 대하여 모든 작업을 수행하여, 이로 인하여 병목 현상을 초래하여 전체 시스템의 성능 저하를 가져오는 문제점이 있다. 또 다른 방법으로는 분산 환경의 모든 서버들이 서로의 컬렉션 정보를 공유하는 방법[4, 5]이다. 이 방법은 검색에 참여하는 정보원의 색인 정보를 분산 환경에서 관리하여, 중앙의 서버에 집중되는 병목 현상을 해소할 수 있는 장점을 갖는다. 그런데 이 방법은 연합 검색기가 검색에 필요한 정보원의 인덱스 정보를 요청할 때, 각 서버에서 자신이 가지고 있는 컬렉션에 관한 모든 정보를 연합 검색기에게 전달해 주어야 하기 때문에 정보 전송에 많은 비용이 요구된다. 따라서 이는 컬렉션이 동적으로 변하는 경우에는 이 방법을 수행하기 매우 어렵다.

다음으로 컬렉션 정보를 이용하지 않는 방법은 학습 질의어를 이용하여 컬렉션의 특성을 파악하고 새로운 질의어에 대해 컬렉션의 관련성 정도를 판단하는 방안이다. 이 방법도 두 가지로 나눌 수 있다. 첫째 지도 학습에 의한 융합 방법으로 Voorhees[2, 3] 연구와 같이 주로 학습자가 질의 결과를 분석하여 컬렉션의 특성을 파악한다. 인터넷과 같이 코퍼스(corpus)가 동적

인 경우에는, 이 방법은 전문가가 학습을 반복하여 시켜야 하므로, 적용하기가 쉽지 않다. 이와는 대조적으로 비지도 학습을 기초로 하는 방법이 있다. 이 방법 [1, 2]은 학습 질의어와 검색된 문서의 유사도를 분석하여 이용한다. 즉, 학습 질의어로 각 컬렉션을 검색하여 얻은 문서들과 학습 질의어와 어느 정도 클러스터링이 잘 되어 있는지를 계산하여 컬렉션의 신뢰도를 측정한다. 이 방법은 기본적으로 어떤 컬렉션에 대하여 질의어를 입력했을 때 검색된 문서들이 실제 질의어와 잘 클러스터링 되어있는 컬렉션은 신뢰할 만 하다는 가정에 기초하여 한다.

본 논문에서 제안한 컬렉션 융합 알고리즘은 질의어와 검색에 참여한 정보원간의 관련성 분포 정보를 이용한다. 이는 사용자로부터 주어진 질의에 적합한 문서를 각 정보원으로부터 임의의 모집단 크기 N 만큼 추출하고, 이를 분석한 후에 정보원의 관련성 분포도를 추론하는 방법이다. 이는 인터넷과 같이 문서들이 동적으로 변화는 환경에서도 쉽게 적용할 수 있으며, 검색 결과에 대한 새로운 융합 정보를 추가할 때에도 쉽게 확장할 수 있다.

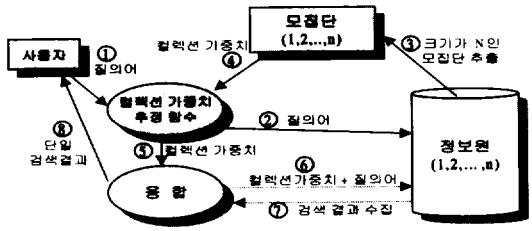
3. 관련성을 이용한 컬렉션 융합 모델

이 장에서는 본 논문에서 제안하는 컬렉션의 관련성 분포 정보를 이용한 컬렉션 융합 모델을 설명하기 위하여 컬렉션 융합의 개괄적인 구조, 주어진 질의에 대하여 컬렉션의 관련성을 추론하는 과정, 그리고 관련성 분포 정보를 이용한 문서의 순위 매김 및 검색 결과의 통합 과정을 논한다.

3.1 컬렉션 융합의 개괄 구조

본 논문에서 제안하는 컬렉션 융합 방법은 질의어와 검색에 참여한 정보원사이의 관련성 분포 정보를 이용한다. (그림 1)은 사용자에 의해 입력된 질의어로부터 각 정보원에서 임의의 모집단을 추출하고, 이들의 관련성 분포 정보를 추정한 후 관련성 분포 정보를 기반으로 각 정보원으로부터 질의어에 적합한 문서를 추출한 후 단일 컬렉션 융합을 수행하는 개괄적인 처리 과정을 보이고 있다.

이들의 처리 순서는 화살표 위에 둥근 번호로 순서를 표시하였다. 먼저 사용자로부터 입력되어진 질의어는 컬렉션의 가중치 추정 함수를 통해 검색에 참여한



(그림 1) 컬렉션 융합의 처리 과정

정보원들과의 관련성 정보를 평가받게 된다. 첫 번째 처리 단계는 질의에 적합한 문서를 각 정보원으로부터 임의의 크기 N개만큼 추출한다. 추출되어진 모든 문서들은 질의어와의 관련성 검사를 수행하여 문서의 관련성 정보를 생성한다. 그리고 이 관련성 정보를 사용하여 해당 정보원에 대한 관련성 분포 정보를 계산한다. 이때 추출되어진 문서와 질의어사이의 관련성 판단은 추출된 문서의 내용 요약 부분에 질의어가 발생한 빈도 수를 근거로 판단하며, 발생 빈도 정도에 따라 관련 문서와 비 관련 문서로 판단한다. 또한 컬렉션에 대한 관련성 분포 정보는 관련 문서의 누적된 순서 정보(order value) 값과 정확도(precision) 값을 곱하여 평가한다.

본 논문에서는 이렇게 평가되어진 관련성 분포 정보 값을 질의에 대해 양질의 문서를 가질 확률과 비례한다고 가정한다. 즉, 질의어에 대해 관련성 분포 정보 값이 크면 클수록 양질의 정보를 가질 확률이 높다고 가정한다.

3.2 정보원의 선택

각 정보원에서 질의어에 의해 추출된 모집단의 문서들은 먼저 질의어와의 관련성 여부를 판단한다. 이는 각 정보원으로부터 추출된 문서내의 내용 요약 부분에 질의어가 발생한 빈도 수에 따라 관련 문서와 비 관련 문서로 판단하고, 이를 통해 질의어와 모집단 사이의 관련성 판단 정보를 추출한다. 본 논문에서는 각 정보원에서 질의어에 적합한 상위 10개의 문서를 대상으로 관련성 분포 정보를 평가하였다. 이때 관련성 판단에 사용된 모집단의 정보원을 C_k라 하면, 이에 대한 관련성 분포 정보 계산은 (그림 2)와 같다. 이는 ProFusion [4] 모델에서 제안한 정보원의 평가 방법을 응용하여 본 논문에서 활용하였다.

$$Ck = \left[\frac{\sum_{i=1}^{10} Ni}{10} * \frac{R}{10} \right]$$

- i : 검색 문서의 위치 정보
- Ck : 정보원 K 에 대한 가중치 값
- Ni : 만약 문서i가 비 관련이면, Ni=0, 아니면 1/i
- R : 각 정보원에서 질의어로 추출된 상위 10개에서 질의어와 관련있는 전체 문서의 수

(그림 2) 컬렉션의 관련성 평가

(그림 2)의 C_k 값은 0과 1사이의 값을 가지며 이는 질의어에 대한 정보원의 관련성 분포 정보 값을 가진다. 이에 대한 평가는 질의어에 의해 추출된 모집단에서 관련 문서들의 누적된 순서 값(order values)과 정확도(precision) 값의 곱으로 계산한다. 먼저 문서들의 누적된 순서 값은 질의어로부터 추출된 모집단에서 관련있다고 판단된 문서의 위치 정보를 누적한 값이다. 이는 관련 문서가 가지고 있는 위치 정보를 정보원에 대한 가중치로 이용하였다. 예를 들어, 10개의 모집단에서 관련 문서가 5개 발생한 두 가지 경우를 살펴본다. 첫 번째는 1~5번째에 관련 문서가 위치한 경우이고, 두 번째는 6~10번째에 관련 문서가 위치한 경우이다. 이때 두 개의 모집단에서 관련 문서의 수는 5개로 동일하다. 만약 관련 문서의 수만으로 정보원과 질의어와의 관련성 분포 정보를 평가하면 동일한 값을 얻는다. 그러나 앞의 두 가지 경우에서 관련 문서들이 위치한 순서는 전혀 다르다. 따라서 이 경우 일반적으로 두 번째보다는 첫 번째의 경우가 좋은 검색 결과라고 할 수 있다. 따라서 이들 두 가지 경우에서 발생하는 문제를 해결하기 위해 누적된 순서 값(order values)을 가중치 값으로 정의하여 이를 해결하였다. 먼저 모집단속에서 관련 문서들은 Ri 값으로 위치 정보를 갖는다. 이 Ri 값을 이용하여 관련 문서의 순서에 대한 의미를 보상했다. 다음은 관련 문서의 위치 정보를 정보원에 대한 가중치 값으로 사용하는 처리 과정을 3단계로 기술하였다.

① Ni의 초기 값을 0으로 한다.

② 먼저 각 문서의 Ri 값을 검사하여 그 값이 1이면 Ni에 "(1/자신의 위치 값)"을 더하고, 아니면 0을 더한다. 그리고 ②번을 모집단의 크기만큼 반복한다.

③ 마지막으로 Ni 값을 모집단의 크기(N)로 나눈다.

두 번째는 관련 문서에 대한 정확도(precision) 값이다. 이는 정보원으로부터 추출되어진 모집단내에서 관련성 검사를 통해 관련있다고 판단된 문서의 전체 수를 모집단의 크기로 나눈 값이다.

(그림 3)는 검색에 참여한 정보원들의 관련성 분포 정보를 상대적인 비율로 계산하는 과정이며, 이는 각 정보원으로부터 검색할 문서의 수를 결정하는 가중치 값으로 사용된다.

$$CollectionRate(CRk) = \frac{Ck}{\sum_{k=1}^{m} Ck}$$

- $\sum_{k=1}^{m} Ck$: 각 정보원 가중치 값의 합계
- Ck : 각 정보원 가중치 값
- m : 검색에 사용된 정보원의 수

각 정보원에서 수집할 최대 문서의 수(Cut-Off Level)

$$CutOffLevel(COLk) = CRk * N$$

- CRk : 각 정보원에 대한 상대 비율값
- N : 검색된 전체 문서의 수(=30)

(그림 3) 정보원에 대한 상대적인 관련성 분포 정보의 계산

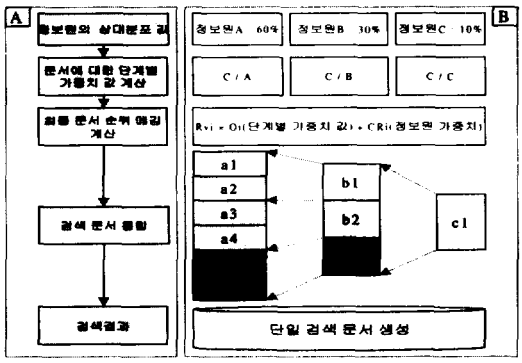
(그림 3)에서 CR_k 값은 질의에 대하여 검색에 참여한 정보원들의 관련성 분포 정보를 상대적인 값으로 계산하는 과정이며, 이는 0에서 100 사이의 값을 가진다. 그리고 CR_k 값에 최종 검색할 전체 문서의 수인 N값을 곱하여 COL_k 값을 계산하며, 이는 주어진 질의어에 대해 정보원으로부터 문서를 수집할 수 있는 최대 수로 사용한다.

3.3 문서의 검색, 순위 매김 및 검색 결과 통합

문서의 순위 매김은 각 정보원으로부터 얻어진 검색 결과를 마치 하나의 정보원으로부터 검색되어진 결과와 같이 단일 집합으로 만들기 위해 검색되어진 모든 문서에 대하여 우선 순위를 부여하는 것을 말한다. 이는 메타 검색기에서 서로 다른 이질의 정보원으로부터 질의에 대하여 검색 결과를 수집하고, 이를 다시 재순위 매김 과정을 통해 모든 문서에 새로운 우선 순위를 부여한다. 그리고 이 문서의 우선 순위를 내림차순으로 모든 문서를 합병하여 사용자에게 단일 검색 결과로서 보여준다.

본 연구에서는 문서의 순서 값, 정보원의 가중치, 문서의 순위 값 등 세 가지 정보만으로 검색되어진 문서

들에 대하여 재 순위 매김을 수행한다. (그림 4)는 두 개의 부분으로 구성되어 있다. 왼쪽의 A 부분은 문서의 순위 매김, 검색 결과 통합 과정 등을 순차적으로 표현하였고, 오른 쪽의 B 부분은 각 과정에서 처리되는 예제를 기술하였다. (그림 4)는 관련성 분포 정보를 이용하여 문서의 재 순위 매김을 처리하는 과정이다.



(그림 4) 관련성 정보를 이용한 문서의 재순위 매김 처리

문서의 재 순위 매김에 대한 첫 번째 처리는 검색에 참여한 정보원들의 상대적인 분포도 값을 계산한다. 이는 검색에 참여한 정보원들이 가지고 있는 분포 값을 합하고, 이를 정보원 자신이 가지고 있는 관련성 분포 값을 분자로 나눗셈을 수행하여 서로에 대한 상대적인 값으로 계산한다. 예를 들어 A, B, C를 검색에 참여한 3개의 정보원의 관련성 분포 정보라고 가정하자, 이때 분포 정보 A의 정보원은 A/(A+B+C)의 값을, 분포 정보 B의 정보원은 B/(A+B+C) 값을, 분포 정보 C의 정보원은 C/(A+B+C) 값의 상대적인 분포도 값을 가진다. 이렇게 추정된 관련성 분포 정보는 본 논문에서 두 가지 형태로 사용된다. 먼저 정보원 선택 단계에서 사용되며, 다음은 문서의 순위 매김 단계에서 사용된다. 정보원 선택 단계에서는 정보원 선택에 대한 가중치로 사용되어 질의에 대하여 해당 정보원으로부터 검색되어질 최대 문서의 수에 영향을 준다. 또한 문서의 순위 매김에서는 문서의 순서 값을 계산할 때 가중치로 결합되어 문서의 순위 매김에 영향을 준다.

두 번째는 문서들간의 간격에 대한 가중치 값 계산이다. 이에 대한 계산은 (그림 5)의 수식으로 계산되며, 해당 정보원에서 문서들간의 순서 값에 대한 간격 가중치 값으로 사용된다. 즉, 검색되어진 문서의 순서 값을 임의의 초기 값으로 설정한다. 그리고 해당 정보원에서 문서의 순서 값을 계산할 때에는 첫 번째 문서에 초기 값을 문서 순서 값으로 할당하고, 다음 문서는 간격 가중치만큼 초기 값에서 감소하여 문서의 순서 값으로 할당한다. 이러한 과정을 반복하여 검색되어진 마지막 문서까지 순서 값을 계산한다.

$$Lwi = \frac{minmum(CRk)}{current(CRk)}$$

- $minmum(CR_k)$: 정보원의 가중치가 가장 적은 값
- $current(CR_k)$: 현재 정보원의 가중치 값

(그림 5) 단계별 가중치 값

(그림 5)에서 L_{wi}값은 0~1 사이의 값을 가지며 각 정보원에서 문서에 대하여 순위 매김을 할 때, 문서의 순서에 대한 간격 가중치 값으로 사용된다. 예를 들면 임의의 정보원에서 가장 첫 번째 문서의 순서 값이 30이고, L_{wi}의 값이 0.3이라면, 두 번째 문서의 순서 값은 순서에 대한 간격 가중치 값인 0.3 만큼을 감소하여 29.7이 된다.

(그림 6)은 질의로부터 검색되어진 문서의 최종 순위 매김에 대한 수식이다.

$$Rvi = Lwi + CRk + \log\left(\frac{N}{\sum_{i=1}^{10} dfki}\right) * tfki$$

- Lwi : 문서의 순서에 대한 가중치 간격 값
- CRk: 각 정보원의 가중치 값
- $tfki$: 정보원 k의 문서 i에서 질의어가 발생한 빈도 수
- $\sum_{i=1}^{10} dfki$: 정보원 k에서 상위 10개에 질의어와 관련 있는 전체 문서의 수

(그림 6) 최종 순위 매김

(그림 6)의 Rvi 값은 정보원으로부터 검색되어진 문서들을 단일 검색 결과로 통합할 때 사용하는 문서에 대한 최종 순위 매김을 처리하는 수식이다. 이때 최종 순위 매김에 사용하는 항목은 (1) 문서의 순서 값, (2) 정보원의 가중치 값, (3) 질의로부터 재평가된 개별 문서의 값 등 3가지이다. 먼저 Lwi는 문서의 간격 가중치 값으로 평가되어진 문서의 순서 값이며, (그림 6)에

서 이에 대한 수식을 기술하고 있다. 다음으로 CRk는 관련성 분포 정보에 의해 평가되어진 정보원의 가중치 값이다. 마지막으로 $\log(\frac{N}{\sum_{i=1}^{m} dfki}) * tfki$는 질의로부터 재평가된 개발 문서의 값이다. 이와 같은 과정을 통해 정보원으로부터 검색되어진 모든 문서는 단일 우선 순위를 가지게 되며, 이를 통해 단일 검색 결과로 만들 수 있다.

4. 제안한 컬렉션 융합 알고리즘

이 장에서는 3장에서 제안한 컬렉션 융합 모델을 구현하기 위한 알고리즘들을 제시한다. 이를 위해 제안된 융합 모델에서 사용한 메타데이터 집합, 사용자 질의에 대하여 양질의 문서가 어느 정보원에 있는지 판단하는 정보원 선택 알고리즘, 검색에 참여한 정보원으로부터 수집되어진 문서들을 통합하는 검색 결과 통합 과정을 논한다.

4.1 메타데이터

본 논문에서 사용한 메타데이터는 총 9개이며 이들은 <표 1>과 같다. 이들은 주로 질의에 적합한 정보원의 선택과 질의에 의해 검색되어진 문서들의 순위 매김 정보로 사용하였다.

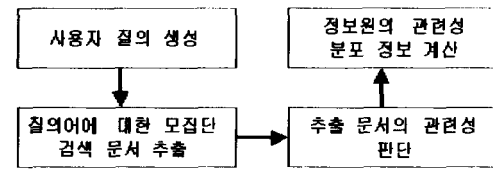
〈표 1〉 HoleInOne 메타데이타

항 목	설 명
DocNo	검색되어진 문서의 순서
Raw scores	정보원에서 제공하는 문서에 대한 ranking 점수
Re-ranking scores	Fusion을 통한 문서에 대한 재 ranking 점수
URL	검색 문서의 주소
Title	검색 문서의 Title 부분
Content	검색 문서의 Abstracts 부분
InformationTypes	정보원의 종류
TitleCount	Title에서 질의어가 발생한 빈도 수
ContentCount	Abstract에서 질의어가 발생한 빈도 수

<표 1>에서 사용한 메타데이터의 기능을 살펴보면, 먼저 **"DocNo"**메타데이터는 검색되어진 문서의 순서 정보를 가지고 있다. 이는 질의에 의해 검색되어진 문서의 위치 정보를 가지고 있으며, 최종 순위 매김에서 문서의 순서 정보를 계산할 때 이를 사용된다. 두 번째는 **"Raw scores"**메타데이터로, 검색되어진 문서의 정보원에서 제공하는 문서의 순위 값을 가지게 된다. 세 번째는 **"Re-ranking scores"** 메타데이터로, 이는 본 논문에서 제안한 질의어 관련성 정보를 기반으로 검색 문서들을 최종 평가하여 이들 문서에 대한 해당 값을 가진다. 네 번째는 **"URL"** 메타데이터로, 검색되어진 문서의 인터넷 위치 정보를 가진다. 이는 질의에 의해 검색되어진 문서가 현재 사용할 수 있는지를 판단할 때 사용된다. 만약 문서가 다른 곳으로 이전되었거나, 인터넷에서 문서가 삭제되었을 경우에는 빈 URL로 판명되어 사용자의 검색 결과 목록에서 삭제된다. 다섯 번째는 **"Title"과 "Content"** 메타데이터이다. 이는 질의에 대해 정보원으로부터 검색되어진 문서에서 문서의 "<title>..</title>" 태그와 <abstract>... </abstract> 태그 안에 있는 내용 데이터를 가진다. 이는 검색 문서와 질의어 사이의 관련성을 판단할 때 사용된다. 여섯 번째는 **"InformationTypes"** 메타데이터이다. 이는 검색되어진 문서가 어느 정보원으로부터 추출되었는지 알 수 있다. 마지막으로 **"TitleCount"**와 **"ContentCount"**는 각 "Title"과 "Content" 메타데이터 안에서 질의어가 일치된 빈도 수이다. 이는 해당 문서가 질의어와의 관련성 분포 정보를 계산할 때 사용된다.

4.2 정보원 선택

정보원의 선택 과정은 사용자가 입력한 질의어에 대하여 가장 적합한 문서를 가지고 있는 정보원을 선택해 준다. 이때 가장 양질의 정보원을 선택하기 위해 (그림 7)과 같은 과정을 수행한다.



(그림 7) 정보원 관련성 검사

(그림 7)에서 질의에 가장 적합한 정보원을 선택하는 과정은 크게 2단계로 수행한다. 첫 번째는 문서의 관련성 판단이다. (그림 7)에서 첫 번째부터 세 번째까지의 과정이 이에 속한다. 사용자로부터 질의어가 주어지면 질의어에 대한 양질의 정보원을 판단해야 하는데, 여기에서는 각 검색에 참여하고 있는 정보원으로

부터 크기 N만큼의 모집단 문서를 먼저 추출한 후에 추출되어진 문서가 질의어와 관련성이 있는지를 판단한다. 그리고 두 번째는 문서들의 관련성 정보를 기반으로 해당 정보원에 대한 관련성 판단 정보를 추론한다

.

4.2.1 모집단에 대한 관련성 검사

이 절에서는 질의로부터 추출된 모집단의 문서에 대하여 질의어와의 관련성 검사를 수행하는 알고리즘을 제시한다. 아래 (그림 8)은 질의어의 의해 추출된 문서에 대하여 관련성을 판단하는 알고리즘이다.

```
1: QueryToRelevanceCheck(String query, String content)
2:     while( SearchToString(query, content) ≠ Φ ) do
                        // 문서내의 질의어 빈도 수 검사
3:         RelevanceValue++;
4:     end_while
5:     return RelevanceValue;
                        // 문서의 관련성 검사 결과
6: End QueryToRelevanceCheck
```

(그림 8) 관련성 판단 알고리즘

(그림 8)의 알고리즘은 질의에 의해 추출된 문서에서 4.2절에서 정의한 "Title"과 "Content" 메타데이터의 내용을 입력으로 받는다. 그리고 질의어와 추출된 문서간의 관련성 판단은 질의어가 "Title"과 "Content" 메타데이터 안에서 일치되어진 횟수에 따라 관련성 유무가 판명된다. (그림 8)의 알고리즘에서는 질의어(query)와 내용 요약(content)등이 입력 변수로 Query oRelevanceCheck 함수에 전달되며, 내용 요약 전체를 검색하여 질의어가 일치하는 횟수를 RelevanceValue에 저장하여 QueryToRelevanceCheck 함수를 호출한 곳으로 이를 되돌려 준다. 이때 질의어 일치 횟수는 0 혹은 그 이외의 값으로 되돌려지며, 0일 경우에는 질의어와 관련없는 문서로, 그 이외에는 질의어와 관련있는 문서로 결정된다.

4.2.2 관련성 분포 정보 평가

이 절에서는 질의에 의해 추출되어진 모집단에 대하여 4.2.1절에서 얻어진 관련 문서 정보를 이용하여 해당 정보원에 대한 관련성 분포 정보를 추측한다. 이 관련성 분포 정보는 모집단에서 관련있다고 판단된 문서의 수에 의해 평가되며, 이때 평가에 사용되는 항목은 관련 문서의 순서 정보, 관련 문서의 개수와 모집단의 크기 등이다. (그림 9)은 정보원에 대한 관련성 분포 정보를 평가하는 알고리즘이다.

```
1:  CollectionToWeight(int CollectionToNum, int SelectedDocNum)
2:    for all i ∈ CollectionToNum do          // 정보원의 수
3:        initial RelevanceDocNum, DocPositionValue;
4:        for all j ∈ SelectedDocNum do       // 상위 10개 문서
5:            CheckValue = QueryToRelevanceCheck(String query, String content);
6:              if(CheckValue != 0) then
7:                  RelevanceDocNum++;        // 관련문서 count
8:                  DocPositionValue = DocPositionValue + (1/index);
9:              end_if                        // 위치 정보에 대한 보상 값
10:       end_for
11:       PrecisionValue = (RelevanceDocNum / SelectedDocNum);
12:       PositionValue = (DocPositionValue / SelectedDocNum)
13:       ColWeight[i] =  PositionValue * PrecisionValue;
14:   end_for
15:   return                                  // 각 정보원에 대한 가중치 계산
16: End CollectionToWeight
```

(그림 9) 관련성 분포 정보 평가 알고리즘

모집단에 대한 관련성 분포 정보는 관련 문서의 순서 정보와 모집단에서 관련되어진 문서 전체의 수 등의 두 가지 정보로 평가된다. 먼저 관련 문서의 순서 정보는 검색에 참여한 정보원으로부터 추출되어진 i번째 문서를 검사하여 관련 문서이면 1/i 값을 누적하고, 아니면 0을 누적한다. 이는 관련 문서로 판단된 문서에 대하여 그 문서가 가지고 있는 위치 정보를 가중치로 사용할 수 있다. 즉 모집단내에서 관련 문서가 첫 번째 나타나는 경우와 마지막에 나타나는 경우에 대하여 서로 다른 가중치를 부여할 수 있다. 다음은 관련되어진 문서 전체의 수로 모집단에 대하여 검색 문서의 정확성을 평가한다. 이는 모집단내에서 질의어와 관련되어진 문서의 전체 수를 추출된 모집단의 크기 N으로 나누어 해당 모집단에 대한 정확도로 평가하였다. (그림 9)의 알고리즘에서 8번째 라인은 관련 문서의 위치 정보에 대한 가중치를 계산하며, 11번째 라인은 모집단에 대한 정확도를 계산한다. 또한 이들 8, 11번째 라인의 계산 결과를 곱하여 해당 정보원에 대한 관련성 분포 정보로 사용하였다.

4.3 검색 문서의 순위 매김 및 통합

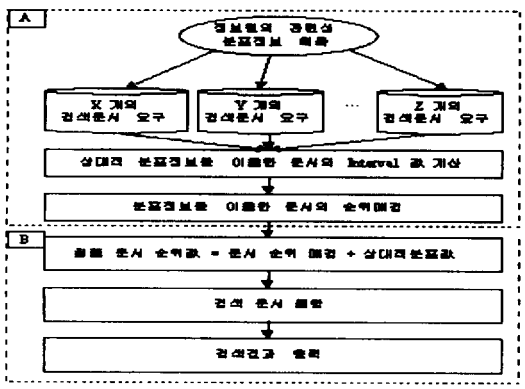
이 절에서는 앞 절에서 얻은 관련성 분포 정보를 사용하여 각 정보원으로부터 질의에 적합한 문서를 수집하고, 수집되어진 문서에 대하여 순위 매김을 수행하여 단일 검색 결과로 통합하는 과정을 기술한다. 이를 위해 관련성 분포 정보를 사용하여 검색에 참여한 정보원으로부터 검색 문서를 수집하고, 수집되어진 각 문서에 대하여 순위 매김을 재평가하는 과정을 논한다. 그리고 각 정보원으로부터 검색되어진 문서들을 사용자에게 단일 검색 결과로 제공하기 위해 문서가 가지고 있는 순위 값에 따라 내림차순으로 통합하는

과정을 보인다. (그림 10)는 검색되어진 문서를 하나의 집합으로 통합하는 과정이다.

4.3.1 검색 문서의 순위 매김

질의에 의해 검색되어진 문서들은 관련성 분포 정보와 이를 응용한 문서의 순서 값을 더하여 순위 매김을 수행한다. 문서의 순위 매김은 크게 (1) 정보원으로부터의 검색 문서 수집, (2) 정보원에서의 간격 값 계산, (3) 관련성 분포 정보를 이용한 순위 매김 등의 3가지 단계를 통해 처리된다. 이는 (그림 10)에서 "A"로 표시되어 있는 부분이다.

먼저 정보원으로부터의 검색 문서 수집은 4.2절에서 계산되어진 관련성 분포 정보를 사용하여 각 정보원으로부터 질의에 대해 검색할 문서의 최대 수를 먼저 계산한다. 이는 검색에 참여한 정보원들의 관련성 분포 정보를 상대적인 비율로 계산한 후에, 메타 검색기에서 최종적으로 검색할 최대 문서의 수를 곱하여 계산한다. 이렇게 계산되어진 값이 해당 정보원으로부터 검색할 문서의 최대 수로 사용된다. 두 번째는 정보원에서의 간격 값 계산이다. 이는 문서에 대한 순서 값을 계산할 때 사용되는 가중치 값이다. 검색에 참여한 정보원들은 관련성 평가를 통해 서로 다른 관련성 분포 정보를 가지게 된다. 이는 질의에 적합한 문서의 개수를 서로 다르게 검색 결과로 돌려준다. 그러나 질의에 의해 검색되어진 모든 문서는 검색 문서의 의미로서는 동등한 입장을 가진다. 따라서 각 정보원에서 검색 문서의 분포도만 다르고 검색 문서의 의미는 동등하다고 가정할 수 있다. 이러한 의미를 부여하기 위해 문서의 순위 매김을 수행할 때, 문서의 간격 값을 가중치로 사용하였다.



(그림 10) 검색 결과 통합 과정

예를 들면 A, B, C 3개의 정보원이 검색에 참여하였고, 이들로부터 관련 문서로 판단된 문서가 각각 6, 3, 1라고 가정하자. 이때 3개의 정보원에 대한 질의어와의 관련성 분포도는 60%, 30%, 10%로 계산된다. 따라서 A, B, C 순서로 양질의 문서를 제공할 수 있다. 그러나 정보원으로부터 검색된 모든 문서는 검색되어진 문서라는 동등한 의미를 가지고 있다. 따라서 사용자에게 검색 문서를 검색 결과로서 되돌려줄 때에는 이 의미를 부여해야 한다. 따라서 예제에서는 6 : 3 : 1의 비율 정보를 문서의 간격 값으로 사용하여 문서의 순위 매김에 가중치로 사용하였다. 즉 A 정보원의 상위 6개 문서는 B 정보원의 상위 3개의 문서와 C 정보원의 상위 1개의 문서와 같은 그룹 내에서 문서의 우선 순위 경쟁을 통하여 이를 내림차순으로 통합하였다. 따라서 관련성 분포 정보가 적게 평가된 정보원에서 검색된 문서들도 문서의 우선 순위를 평가할 때에 다른 정보원과 동등한 입장에서 문서를 평가받을 수 있다. 이에 대한 알고리즘을 (그림 11)과 같이 사용하였다.

```
1: IntervalValue(int ColWeight, int CurrentColWeight, int ColSearchToNum)
2:  move ColWeight[0] to MinValue;
3:     for all i ∈ ColSearchToNum do
4:          if(MinValue < ColWeight[i+1]) then
5:              MinValue = ColWeight[i+1];
6:          end_if  // 정보원 가중값 중에서 최소 값을 찾는 과정
7:      end_for
8:     InterWeight = MinValue / CurrentColWeight; // 문서사이의 간격 값
9:   return InterWeight;
10: End IntervalValue
```

(그림 11) 문서 사이의 간격 값 계산 알고리즘

(그림 11)은 검색에 참여한 정보원의 가중치 값을 상대적인 비율 값으로 변환하여 계산한다. 즉 정보원 중에서 가장 적은 가중치 값을 분자로 하고 자기 자신의 정보원 가중치 값을 분모로 하여 나눈 값을 해당 정보원에서 검색되어진 문서의 간격 값으로 사용한다. (그림 11)에서 3번째부터 7번째 라인에서 검색에 참여한 정보원 중에서 가장 적은 가중치를 가지는 정보원의 가중치 값을 찾는다. 그리고 8번째 라인에서 자신의 정보원 가중치 값으로 나누어서 검색되어진 문서간의 간격 값을 계산한다.

마지막으로 관련성 분포 정보를 이용한 최종 순위 매김이다. 이는 관련성 분포 정보와 문서의 순서 값을 더하여 검색 문서에 대하여 순위 매김을 하였다.

4.3.2 검색 문서의 통합

질의에 의해 검색되어진 문서는 4.3.1절에서 관련성

분포 정보를 이용하여 문서의 순위 매김을 수행하였다. 앞에서 관련성 분포 정보를 이용한 순위 매김에서는 검색에 참여한 모든 정보원이 동등하며, 검색 문서의 분포만 다르다고 가정하여 문서 순위 매김을 수행하였다. 그러나 검색에 참여한 정보원들 사이에서 동일한 우선 순위를 가지는 검색 문서가 서로 비교되는 경우가 있다. 이 경우에는 관련 분포 정보가 클수록 양질의 검색 문서를 가질 확률이 높다라고 가정하였으므로, 이를 위해 관련성 분포 정보를 최종 순위 매김의 가중치로 사용하였다. (그림 12)는 문서에 대한 최종 순위 매김 알고리즘이다.

```
1: DocRankValue(int ColWeight, int ColSearchToNum, int DocMaxNum)
2: for all i ∈ ColSearchToNum do
3:   InterWeight=IntervalValue(ColWeight,CurrentColWeight,ColSearchToNum);
4:   DocNum = InterWeight * DocMaxNum;
5:     for all j ∈ DocNum do
6:       DocRankScore[j,i] = ColWeight[i]-(DocMaxNum-(InterWeight[i]*j));
7:     end for // 문서의 순위 매김
8:   end for
9:  return;
10: End DocRankValue
```

(그림 12) 최종 순위 매김 알고리즘

(그림 12)에서 5번째 라인부터 7번째 라인까지가 검색되어진 모든 문서에 대하여 최종 순위 매김을 수행하고 있다. 이들은 해당 정보원에 대한 가중치 값과 문서에 대한 단계별 가중치를 더하여 최종 순위 매김 값으로 사용한다.

5. 실험 결과

이 장에서는 본 논문에서 제안한 컬렉션 융합 알고리즘을 평가하기 위한 실험 환경, 실험 결과 등을 비교 분석하였다. 본 논문에서는 제안한 융합 알고리즘을 평가하기 위해 실험적인 HoleInOne 메타 검색기를 구현하여 실험하였으며, 현재 일반적으로 많이 사용되고 있는 5개 정보원을 선정하여 이들과 실험 결과를 비교 분석하였다.

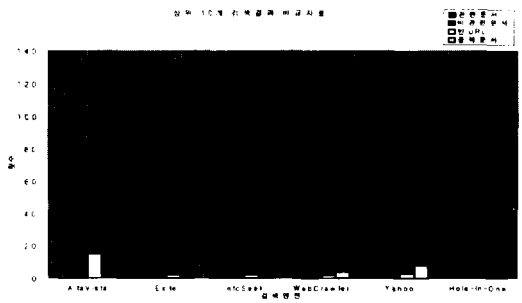
5.1 융합 알고리즘에 대한 성능 분석

먼저 각 정보원에 입력될 질의어는 뉴스 그룹에서 사용하고 있는 13개의 단일 주제어를 선택하였다. 이는 다양한 기능을 가진 질의어를 사용할 때는 불리언 모델에 바탕을 둔 질의어 처리와 순위 매김(ranking) 모델에 바탕을 둔 질의어 처리에 대하여 성능의 결과가 각기 달리 나올 수 있다. 따라서 이는 제안된 알고리즘의 성능을 분석하는데 변수가 될 수 있기 때문에, 질의어 처리 모델과는 독립적인 실험 결과를 얻기 위하여 질의어를 단일 단어로 한정하였다.

이에 대한 질의어로는 (1) Engineering, (2) Computer, (3) Travel, (4) Medical, (5) Finance, (6) Religion, (7) Government, (8) Animals, (9) History, (10) Recreation, (11) Art, (12) Music, (13) Food 등이다. 이들 질의어로 검색되어진 문서를 평가한 항목은 (1) 관련 문서의 수, (2) 비 관련 문서의 수, (3) 빈 URL, (4) 관련 문서 가운데 한번이상 검색된 문서의 수, (5) 관련 문서 가운데 한번만 검색된 문서의 수, (6) 정확도 등의 항목에 대하여 평가하였다.

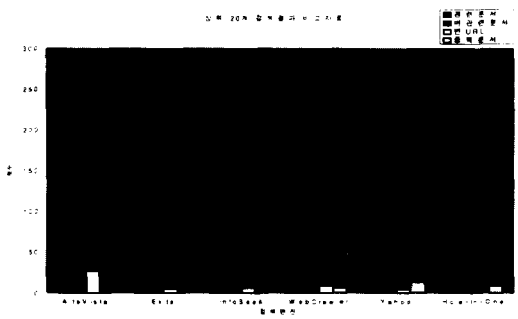
이들 중에서 (1), (5) 항목은 정보원에 대하여 검색 결과의 정확성을 판단할 수 있는 항목이며, (2), (3), (4) 항목은 검색 결과의 효율성을 판단할 수 있는 항목이다.

다음 (그림 14)는 질의어에 의해 검색되어진 문서 가운데 상위 10개의 문서를 앞의 평가항목 중에서 (1)부터 (4)까지를 분석하여 막대 그래프로 나타내었다. 이때 실험의 결과는 본 논문에서 제안한 메타 검색기가 다른 정보원에 비해 우수한 검색 효율성을 보였다.



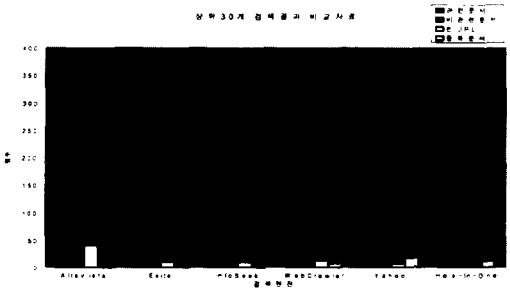
(그림 14) 상위 10개 문서의 검색 결과

다음 (그림 15)은 질의어에 의해 검색되어진 문서 가운데 상위 20개의 문서를 앞의 평가항목 중에서 (1)부터 (4)까지를 분석하여 막대 그래프로 나타내었다. 이 실험 결과는 (그림 14)에 나타나는 결과를 포함하고 있으며 (그림 14)에서 분석한 실험 결과와 유사하게 나타났다.



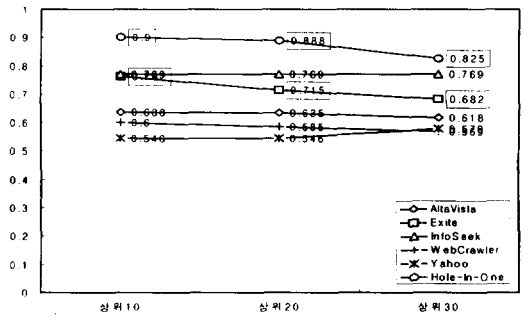
(그림 15) 상위 20개 문서의 검색 결과

다음 (그림 16)는 질의어에 의해 검색되어진 문서 가운데 상위 30개의 문서를 앞의 평가항목 중에서 (1)부터 (4)까지를 분석하여 막대 그래프로 나타내었다. 이 실험 결과는 (그림 15)에서 나타나는 결과를 포함하고 있으며 앞의 두 가지 경우와 유사한 실험 결과를 얻었다.



(그림 16) 상위 30개 문서의 검색 결과

(그림 17)은 실험에 참여한 6개 정보원의 검색 문서 결과에 대한 정확성을 평가한 자료이다. 세로는 0부터 1사이의 값을 가지며 정확도에 대한 값을 0.1의 간격으로 표시하였으며, 가로는 정확도를 평가하기 위한 단위 집합으로 상위 10개, 상위 20개, 상위 30개 등의 검색 문서 집합을 표시하였다. 그리고 여기에서 사용한 정확도는 13개의 질의어를 전체 집합으로 가정하고 상위 10, 20, 30개의 검색 문서 집합을 대상으로 평가하였다. 즉 상위 10개의 검색 문서에 대한 정확성은 13개의 주제어와 각 주제어로부터 검색되어질 상위 10개의 문서 수를 곱하여 전체 문서 집합을 130으로 하였다. 여기에 평가 항목 가운데 (5)를 분자로 하여 계산한 값을 상위 10개의 검색 문서에 대한 정확도 값으로 하였다. 나머지 상위 20, 30개의 검색 문서 집합도 동일한 방법으로 계산하였는데, 그 결과는 (그림 17)과 같다. 그림에서 보는바와 같이 제안된 랭킹 알고리즘은 기존의 검색 엔진보다 정확성, 효율성 면에서 우수성을 보였다.



(그림 17) 검색 문서에 대한 정확도

6. 결론 및 향후 연구 과제

본 논문에서는 인터넷에서 사용자가 정보를 검색할 때 질의에 대하여 검색 결과의 정확성과 효율성을 개선한 컬렉션 융합 알고리즘을 제안하고, 실험적인 메타 검색기를 구현하여 성능 평가를 하였다. 제안된 컬렉션 융합 방법은 분산되어 있는 다양한 정보원들을 검색에 참여시켜서 이들을 통해 질의에 적합한 양질의 문서를 찾을 수 있었으며, 5장의 실험 결과를 통해 확인하였다.

앞으로는 제시된 융합 모델은 정보원에 대한 정보를 필요로 한다. 정보원에 대한 양질의 정보를 얻기 위해서는 질의에 적합한 정보원을 선택할 수 있도록 표준화된 메타데이터 개발이 필요하고, 정보원에 대한 정보 수집 방법과 융합 클러스터링 기법의 개발 등에 대한 연구가 필요하다. 또한 질의어 처리 기능의 확장이 필요하다. 즉, 불리언 모델에 바탕을 둔 질의어 처리 기능과 순위 매김(ranking) 모델에 바탕을 둔 질의어 처리 기능 등의 연구이다. 이러한 정보는 본 논문에서 제시된 알고리즘의 성능을 크게 개선시킬 수 있다.

참 고 문 헌

[1] J. P. Callan, Z. Lu, and W. B. Croft, "Searching Distributed Collections with Inference Networks," In Proceedings of the Eighteenth Annual Inter-

national ACM SIbv4GIR Conference on Research and Development in Information Retrieval, Seattle, WA. pp.21-28. 1995.

[2] E. M. Voorhees, N. K. Gupat, and B. Johnson-Laird., "The Collection Fusion Problem," In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3), Gaithersburg, MD, pp. National Institue of Standards and Technology, Special Publication pp.500-225, 1994.

[3] E. M. Voorhees, N. Gupta, and B. Johnson-Laird., "Learning Collection Fusion Strategies," SIGIR '95, pp172-179, 1995.

[4] S. Gauch, G. Wang, and M. Gomez, "ProFusion: Intelligent Fusion from Multiple, Distributed Search Engines, WebNet '96," The First World Conference of the Web Society, San Francisco, October, 1996.

[5] C. Baumgarten, "A Probablistic Model for Distributed Information Retrieval," SIGIR '97, pp.258-267, 1997.

[6] B. T. Bartell, G. W. Cottrell, and R. K Belew, "Automatic Combination of Multiple Ranked Retrieval," SIGIR '94, pp.173-181, 1994.

[7] J. Xu and J. P. Callan. "Effective Retrieval with Distributed Collections." In Proceedings of the Twenty-first Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, August, 1998.

[8] B. T. Bartell. G. W. Cottrell, and R. K. Belew, "Automatic Combination of Multiple Ranked Retrieval Systems." In Croft, W. B. and van Rijsbergen, C., editors, SIGIR 94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Dublin. Springer-Verlag. 1994.

[9] D. Harman, W. McCoy, R. Toense, and G. Candela. "Prototyping a Distributed Information Retrieval System Using Statistical Ranking," Information Processing and Managements, 27(5):449-460, 1991.

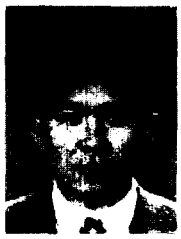
[10] A. Moffat and J. Zobel, "Information Retrieval Systems for Large Document Collection," TREC- 3, 1994

[11] C. Baumgarten, "A Probablistic Model for Distributed Information Retrieval," SIGIR '97, pp.258-267, 1997.

[12] 금기문, 남세진, 신동욱, 김태균, "문서 클러스터링 정보를 이용한 컬렉션 융합", 한국정보과학회 추계학술 논문발표집. pp.147-149, 1998. 10.

[13] 김현주, 김상준, 배종민, "관련성 분포 정보를 이용한 컬렉션 융합", 한국정보처리학회 춘계학술 논문발표집. pp.907-910, 1999. 4.

김 현 주



E-mail : hjk@base.gsnu.ac.kr

1988년 경상대학교 전산통계학과 졸업(학사)

1990년 숭실대학교 전자계산학과 공학석사

1999년 경상대학교 전자계산학과 박사과정 수료

1994년~1997년 제일정밀공업(주) 연구원

관심분야 : 정보검색, 디지털 도서관, 컴파일러

김 상 준

E-mail : twist@sys.gsnu.ac.kr

1999년 경상대학교 컴퓨터과학과 졸업(학사)

1999년~현재 경상대학교 전자계산학과 석사과정 재학중

관심분야 : 정보검색, 디지털 도서관, RDF, SGML/XML

배 종 민

E-mail : jmbae@nongae.gsnu.ac.kr

1980년 서울대학교 사범대학 수학과 졸업(학사)

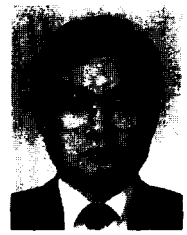
1983년 서울대학교 계산통계학과 이학석사(전산학)

1995년 서울대학교 계산통계학과 이학박사(전산학)

1982년~1984년 한국전자통신연구원 연구원

1984년~현재 경상대학교 컴퓨터과학과 교수

관심분야 : 병렬 프로그래밍 언어, 디지털 도서관, 정보



강 현 석

E-mail : hskang@nongae.gsnu.ac.kr

1981년 동국대

학교 전자계산학과 졸업 (학사)

1983년 서울대학교 계산통계학과 이학석사(전산학)

1989년 서울대학교 계산통계학과 이학박사(전산학)

1981년~1984년 한국전자통신연구원 연구원

1984년~1993년 전북대학교 전자계산학과 부교수

1993년~현재 경상대학교 컴퓨터과학과 교수

관심분야 : 객체지향 데이타베이스, 컴퓨터 그래픽스, 멀티미디어