문서 클러스터링에 의한 효율적인 병렬 정보검색 시스템

(An Efficient Parallel Information Retrieval System using Document Clustering)

강 유 경 † 류 광 렬 †† 정 상 화 ††
(Yu Gyung Kang)(Kwang Ryel Ryu)(Sang-Hwa Chung)

요 약 본 논문은 고품질의 정보를 신속하게 제공할 수 있으면서 가격대 성능비가 우수한 병렬 정보 검색 시스템을 제시하고 있다. 본 검색 시스템은 문서 라이브러리를 여러 개의 클러스터로 세분화하고 검색 시 클러스터 단위로 프로세서에 할당함으로써 작업 단위를 적절한 규모로 하였을 뿐만 아니라, 문서의 점수 계산 시 프로세서 간 통신이 전혀 필요치 않게 하였다. 검색은 1차로 클러스터 레벨에서 관련 클러스터들을 찾는 것으로 시작하여 2차로 관련 클러스터 내에서 실제 문서를 찾는 방식으로 이루어진다. 이러한 계층적인 검색 구조로 인하여 1차 검색 후 여과가 가능하므로 전체적인 검색의 부하를 줄일 수 있다. 또한 문서의 클러스터가 가능한 한 유사한 문서군이 되도록 함으로써 불필요한 클러스터가 검색될 가능성을 최소화하여 성능을 높였다. 본 검색 시스템은 분산메모리 MIMD 구조의 다중 트랜스퓨터 시스템에서 구현되었으며, 실험 결과 무작위적으로 클러스터링한 경우에 비해 유사 문서군으로 클러스터링한 접근 방법이 우수함을 확인하였다.

Abstract This paper presents a fast and cost-effective parallel information retrieval system. In this system, the document library is divided into many clusters so that retrieval tasks are assigned to each processor on cluster-by-cluster basis. To minimize the number of clusters relevant to a user's query, each cluster is made to consist of similar documents. The size of the clusters are determined in such a way that the balancing of loads among the working processors is easily achievable. The retrieval is done in two stages. In the first stage, all the clusters relevant to the user's query are retrieved and those with low scores are screened out to lessen the load of the second stage. The qualified clusters are then distributed to the processors. In the second stage, relevant documents are retrieved from the assigned clusters and those documents are scored. There is no need for inter-processor communication during document scoring because all the necessary information is self-contained in the respective cluster. The system has been implemented on a multi-transputer system which is a distributed memory MIMD machine. Experimental results show that our similarity-based document clustering scheme gives better performance than a random clustering scheme.

1. 서 론

현재 문서 검색 기술의 주류는 통계적 접근방식을 기본으로 한 것으로 사용자가 지정한 키워드들을 포함한 문서들을 문서 라이브러리로부터 모두 찾은 뒤, 키워드의 중요도 및 등장빈도에 따라 순위를 부여하여 사용자에게 되돌려 주는 방식을 취하고 있다. 지정된 단어를 포함한 문서를 신속히 찾기 위한 수단으로는 색인어 역파일(inverted index file)이 널리 사용되고 있다. 색인어 역파일이란 문서 라이브러리에 들어 있는 모든 문서에 등장하는 모든 단어들에 대해 각 단어별로 그것이 등장하는 문서들의 리스트를 그 단어의 문서별 가중치와 함께 가지고 있는 파일이다(그림 2 참조). 색인어 역

· 본 연구는 한국과학재단의 핵심전문연구과제(과제번호 971-0901-013-2) 지원사업에 의해 수행되었음
† 비 회 원 : 통계청 전산사무관
summaul@nso.go.kr
†† 종신회원 : 부산대학교 컴퓨터공학과 교수
krryu@hyowon.cc.pusan.ac.kr
shchung@hyowon.cc.pusan.ac.kr
논문접수 : 1999년 3월 17일
심사완료 : 2000년 8월 14일

파일을 만들 때 단어들을 사전적 순서대로 정렬해 둠으로써, 입력되는 질의어와 관련이 있는 문서들 및 그 문서에서의 가중치 정보를 신속하게 찾을 수 있다. 해당 문서들을 찾은 뒤에는 해당 질의어의 중요도와 문서 내 가중치 등을 고려하여 문서들의 점수를 계산한 후 순위를 부여한다. 이때 사용자가 원하는 고품질의 정보를 제공하기 위해 P-norm 모델[4,16]이나 본문 검색과 같은 신뢰성이 높은 점수계산 모델을 도입할 경우 파생되는 문제는 이를 위한 계산과정의 복잡성 또한 높아져서 계산시간이 늘어난다는 것이다. 이러한 상황에서 순위 부여에 소요되는 계산시간을 단축하기 위하여 효율적인 병렬처리 방안을 강구하는 것이 바람직하다.

본 논문에서는 문서 라이브러리를 가능한 한 유사한 문서군으로 이루어진 클러스터들로 세분화하고, 색인어 역파일을 클러스터 레벨 및 문서 레벨의 두 단계로 계층화하여 접근하는 새로운 병렬처리 방안을 제시하고 있다. 문서 라이브러리를 여러 개의 문서군으로 클러스터링하는 것은 각 프로세서에 클러스터 단위로 작업을 할당할 수 있게 하여 병렬처리를 용이하게 한다. 그리고 이들 클러스터들을 대상으로 하여 만든 클러스터 레벨 색인어 역 파일과 각 클러스터마다 그 클러스터 내에 속하는 문서들을 대상으로 만든 문서 레벨 색인어 역파일의 2단계 구조로 인하여 계층적인 검색이 가능하다. 1차 검색은 클러스터 레벨 색인어 역 파일을 이용하여 질의어와 관련된 클러스터들을 검색하는 것을 말하고 2차 검색은 문서 레벨 색인어 역 파일을 이용하여 질의어와 관련된 문서들을 검색하는 것을 말한다. 이러한 계층적 검색 구조 하에서 문서라이브러리를 가능한 한 유사한 문서군으로 세분화하게 되면 질의 관련 클러스터의 수가 줄게 되어 1차 검색의 부담을 줄일 수 있다. 또한 1차 검색 후 여러 정보들을 이용하여 클러스터들의 순위를 부여한 후 여과를 한다면 2차 검색의 부담 또한 줄일 수 있다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 본 논문과 관련되는 기존의 연구에 대하여 서술한다. 3장에서는 본 연구에서 제안하는 문서 라이브러리의 클러스터링을 통한 정보검색의 병렬화 방안을 설명한다. 4장에서는 문서 클러스터링에 사용될 수 있는 구체적 기법들을 소개하고 정보검색 병렬화에 응용하기 위한 효과적인 적용 방안에 대해 토의한다. 그리고 5장에서는 본 연구의 정보검색 병렬화 방안을 분산 메모리 MIMD 구조의 다중 트랜스퓨터 시스템에서 구현하여 실험한 결과를 제시하고 이를 바탕으로 6장에서 결론을 맺는다.

2. 관련 연구

2.1 병렬 정보검색

기존의 병렬 정보 검색에 관한 연구로는 Stanfill의 MPP 모델[17]이 있다. Stanfill의 MPP모델은 Connection Machine 상에서 구현된 것으로, 색인어 역파일의 분산 저장 시 한 문서에 포함된 색인어들 및 그와 관련된 역파일 정보가 하나의 프로세서에 모여 있도록 하였다. 분산 저장을 고르게 하기 위해서는 그림 1에 보인 바와 같이 무작위적으로 부여한 문서 ID를 각 프로세서에 interleaving 방식으로 배정하였고, 배정된 ID의 문서에 포함된 색인어들과 관련 정보를 해당 프로세서에 저장하였다. 이 방식에서는 각 색인어의 역파일 정보가 실제로 어떤 프로세서에 저장되어 있는지를 indexing해 주는 별도의 데이타 맵을 필요로 한다.

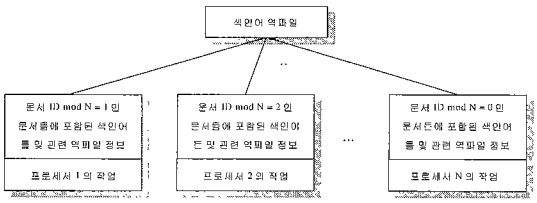

그림 1 Stanfill 방식의 병렬 정보검색

Stanfill 방식은 그림 1에서 보는 것 같이 문서 ID에 따라 작업 프로세서가 결정되어 한 문서에 속하는 모든 정보는 같은 프로세서로 보내지기 때문에 문서의 점수 계산 시 프로세서간 통신이 불필요하다는 장점을 지닌다. 그러나 Stanfill 방식에서는 계층적인 검색을 할 수 없어서 여과가 불가능하고 이로 인하여 질의어와 관련된 문서가 아주 많은 경우에는 검색의 부담이 크다. 또한 새로운 문서가 문서 라이브러리에 추가될 때마다 거쳐야 하는 색인과 관련한 최적화 과정에서 색인어 역파일의 전체 저장 구조를 다시 만들어야 하는 단점이 있다.

병렬 정보검색을 위해 Sharma[15]는 각 프로세서를 하이퍼큐브로 연결하고 각 프로세서마다 전용 하드디스크를 장착한 시스템을 사용하였다. 검색에 사용할 문서는 클러스터링되어 모든 디스크에 균등하게 배분되어 있고, 호스트컴퓨터로부터 사용자 질의를 받으면 각 프로세서는 질의와 관련있는 문서 클러스터를 호스트로 전송하게 된다. 이 시스템은 검색결과로 클러스터 전체를 넘겨주므로 정확도가 다른 검색 시스템에 비하여 좋지 않으며 또한 해당 문서 클러스터가 저장된 하드디스

크를 보유한 프로세서만 검색에 참여하므로 병렬처리의 효율이 클러스터의 지역성(locality)에 의해 좌우되는 단점이 있다.

Cahoon[1]은 대용량 문서 라이브러리에 대한 다중 사용자의 동시적, 분산된 질의 검색을 제공하는 분산 정보검색 시스템을 제시하였다. 이는 워크스테이션 네트워크 상에 구현된 것으로 검색을 수행하는 Inquery 서버와 검색 시스템의 관리자인 중앙 서버로 구성되어 작업량을 분산시킴으로 인해 검색 시스템의 성능을 향상시킬 수 있으나, 질의어가 크고 검색 요구가 계속적으로 증가하는 경우 네트워크 및 Inquery 서버의 병목현상으로 성능 향상의 한계를 지니고 있다.

[13,20]에서는 정확한 정보제공을 위해 본문검색을 실시하고 신속한 정보검색 서비스를 제공하기 위하여 다수 개의 질의를 병렬로 처리하며, 각 질의 또한 병렬로 처리하는 병렬 정보 검색 모델을 제시하고 있다. 질의를 병렬로 처리하기 위해서는 각 질의별로 검색할 프로세서 클러스터를 할당하는 것이 필요한데, [13]에서는 프로세서 클러스터의 할당을 효율적으로 수행하는 알고리즘을 제시하고 있다. 또한 본문검색은 상당한 시간을 요하는 작업이므로 문서들을 프로세서에 할당하는 방법에 따라 검색의 성능이 달라지게 되는데 [20]에서는 다양한 문서 할당 방법에 따른 검색 성능을 비교하고 있다. 이들 연구는 본문검색을 전제로 한 것으로 실제 현재 많이 사용되고 있는 대규모 색인어 역파일을 기반으로 한 검색 시스템을 병렬화하는 방안을 제시한 것은 아니다.

2.2 문서 클러스터링

[11]에서는 SONIA시스템을 제시하고 있다. SONIA 시스템은 웹 검색 엔진이 돌려주는 URL주소리스트를 입력으로 받아 관련되는 문서들을 읽어온다. 이를 대상으로 다단계에 걸친 속성선별작업을 한 후 문서들을 클러스터링한 결과를 사용자에게 되돌려준다. 이때 해석상의 편의를 위해 각 클러스터를 설명해주는 중심 키워드들의 리스트 또한 첨부한다.

[8]에서는 클래스를 알지 못하는 문서들을 이용하여 문서 분류기(classifier)의 성능을 향상시키는 방법을 보여주고 있다. 이는 클래스를 알고 있는 문서들의 비용이 비싸다는 점에 착안한 것으로 그 원리는 다음과 같다. 먼저 적은 수의 레이블된 문서들을 대상으로 분류기를 만든 후 이를 이용하여 클래스를 알지 못하는 문서들의 클래스를 정한다. 그 후 모든 문서들을 대상으로 다시 새로운 분류기를 만들고 이러한 단계를 수렴할 때까지 계속해서 한다.

그러나, 이들을 포함하여 문서 클러스터링에 관한 종래 연구들을 보면, 본 논문에서와 같이 병렬처리의 효율 향상에 그 목적을 두고 있는 것은 찾아보기 어렵다.

3. 문서 클러스터링을 이용한 병렬 정보검색

3.1 정보검색의 병렬화 문제

본 연구의 병렬 정보검색 시스템의 구현 대상 컴퓨터는 여러 개의 로컬 하드디스크를 가진 분산메모리 MIMD 구조의 다중 트랜스퓨터 시스템이다(3.3절 참조). 이러한 시스템에서 병렬 검색이 효율적으로 이루어지기 위해서는 검색 대상 자료가 여러 하드디스크에 분산 저장됨으로써 디스크 작업이 병렬로 이루어질 수 있어야 하고, 디스크 작업 이후 여러 관련 문서의 점수 계산 과정 또한 모든 작업 프로세서에서 고르게 이루어지도록 부하가 균등화되어야 한다.

디스크 작업의 병렬화를 위한 한 가지 방안으로는 그림 2와 같이 색인어 역파일을 단순히 분할하여 여러 로컬 디스크에 분산 저장시키는 것을 생각할 수 있다. 이렇게 하면 사용자의 질의 입력 시 질의어들과 관련된 색인어 역파일 정보를 읽어 들이기 위한 디스크 작업은 적절히 병렬로 이루어질 것이다. 그러나, 이 방식은 문서의 점수 계산 과정에서 부하의 배분이 까다로울 뿐 아니라 프로세서간 과다 통신이 발생하는 문제를 안고 있다. 예로써, t_1과 t_k 및 t_{k+1}이 질의어로 포함된 질의를 처리하는 과정을 생각해 보자. 디스크 작업이 완료되고 나면 각 작업 프로세서에 적절히 작업을 할당하여 관련 문서들의 점수를 계산해야 하는데, 예를 들어 문서 d_7의 점수 계산을 위해서는 먼저 d_7내에 등장하는 질의어 t_1, t_k, t_{k+1}의 가중치 정보인 w_1^7, w_k^7, w_{1k+1}^7를 모두 하나의 작업 프로세서로 취합하여야 한다. 이러한 취합 과정은 관련 문서 모두에 대해 필요하므로 상당한 통신이 발생

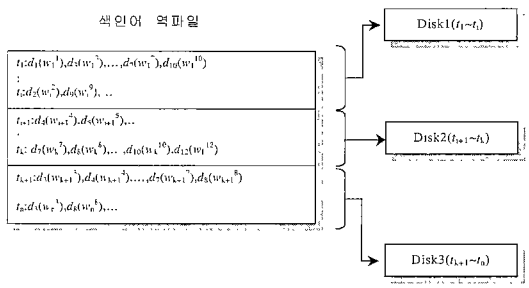

그림 2 색인어 역파일의 단순 분할 (로컬 디스크가 3개인 경우)

하게 된다. 뿐만 아니라, 문서별 점수 계산 작업을 어떻게 각 프로세서에 나누어 할당할 것인가 하는 것도 까다로운 문제가 된다. 다음 절에서는 이러한 문제점을 극복할 수 있도록 본 연구에서 제안하는 병렬화 방안을 설명한다.

3.2 문서 클러스터링과 계층적 색인어 역파일

본 연구에서 제안하는 방법의 핵심은 문서 라이브러리의 클러스터링에 있다. 문서 라이브러리를 다수의 작은 클러스터로 나누는 목적 중 하나는 트랜스퓨터 기반의 분산메모리 MIMD 구조에 적합하도록 작업 할당 단위를 소규모화 하는데 있다. 그림 3과 같이 문서 라이브러리를 작은 규모의 클러스터들로 나눈 다음 각 클러스터별로 그 내부의 문서만을 대상으로 하는 독립적 색인어 역파일(문서 레벨 색인어 역파일)을 만들어 두면 검색 작업이 각 클러스터별로 독립적으로 이루어질 수 있게 된다. 각 클러스터별로 만들어진 문서 레벨 색인어 역파일들은 클러스터별로 로컬 하드디스크에 분산 저장되어, 검색 시 관련 역파일들이 여러 디스크로부터 병렬로 읽혀질 수 있게 한다.

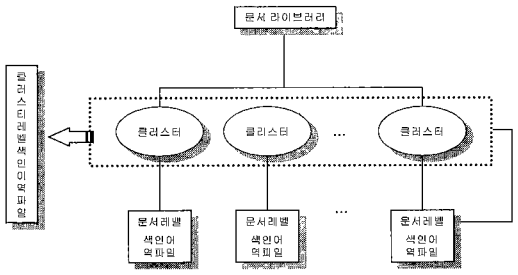

그림 3 문서 클러스터링에 의한 계층적 색인어 역파일

질의가 들어오면 마스터(master) 프로세서는 그 질의에 포함된 질의어들이 나타나는 문서들이 어떤 클러스터들에 속해있는지 알 수 있어야 한다. 이를 위해 본 연구에서는 그림 4에 나타낸 바와 같은 구조의 클러스터 레벨의 색인어 역파일을 별도로 만들어 계층적 색인어 역파일을 운영하는 방안을 제안하고 있다. 클러스터 레벨 색인어 역파일은 각 클러스터별로 그 클러스터에 속한 문서들을 모두 병합함으로써 클러스터 하나를 마치 하나의 문서인 것처럼 간주하여 만든 색인어 역파일이다. 일반적으로 문서 레벨의 색인어 역파일은 문서 라이브러리의 규모가 클 경우 그 크기가 대단히 커지지만, 클러스터 레벨 색인어 역파일은 클러스터의 수가 문서의 수에 비하면 훨씬 적으므로 그 크기가 크지 않다. 따라서, 클러스터 레벨 색인어 역파일은 입력 질의를 처리하는 마스터 프로세서의 메모리에 저장해 둔다. 질의 입력 시 이 파일을 이용한 간단한 1차 검색을 통해 마스터 프로세서는 질의어들과 관련된 클러스터 ID들을 파악하고, 클러스터 단위로 각 프로세서에 작업을 할당한다. 즉, 클러스터가 작업의 할당 단위이며 부하 균등화의 단위가 되는 것이다. 부하 균등화를 위해서는 각 클러스터별로 질의어와 관련된 문서의 수를 고려하여 각 작업 프로세서에서 처리해야 할 문서의 수가 비슷하도록 클러스터들을 할당한다.

단어 ID	클러스터 ID	관련 문서 수	클러스터 내 단어 빈도수
t_j	C_i	D_{ij}	tf_{ij}

그림 4 클러스터 레벨 색인어 역파일 구조

그림 4에서 클러스터 내 단어 빈도수는 관련 문서 수와 함께 클러스터의 점수를 계산하는데 사용된다. 클러스터의 점수는 클러스터 내의 총 문서 수 대비 관련 문서 수의 비율이 높으면서 클러스터 내 단어 빈도수가 큰 클러스터일수록 높다. 이 점수가 일정 임계치 이하인 클러스터들을 여과해 버릴 경우 이후 각 작업 프로세서에서 수행될 문서 레벨의 2차 검색의 부담을 크게 줄일 수 있다는 이점도 있다. 앞서도 설명하였지만 이러한 조기 여과는 앞의 2.1절에서 설명한 Stanfill 방식에서는 불가능한 것이다.

2차 검색은 클러스터 ID들을 할당받은 각 작업 프로세서들이 해당 클러스터들의 문서 레벨 색인어 역파일들을 로컬 하드디스크로부터 병렬로 읽어 들이는 작업으로 시작한다. 문서 레벨 색인어 역파일로부터 질의어들이 등장하는 문서들을 모두 찾은 뒤 문서별로 각 질의어의 가중치 정보를 이용하여 점수를 계산한다. 이 방법에서는 해당 문서와 관련된 질의어들에 대한 정보가 모두 그 프로세서에 할당된 클러스터 내에 있기 때문에 문서의 점수 계산 시 프로세서들 간 통신을 필요로 하지 않는다.

3.3 병렬 정보검색 시스템의 구조

본 연구에서 사용한 병렬 정보검색 시스템은 그림 5에서 제시된 바와 같이 호스트 컴퓨터와 병렬 컴퓨터로 구성된다[19]. 호스트 컴퓨터는 사용자 질의로부터 2차 검색 대상 클러스터를 선정하는 1차 검색을 수행하고 그 결과를 병렬 컴퓨터로 공급하며, 병렬 컴퓨터로부터 추출된 검색 결과를 사용자에게 제공한다. 병렬 컴퓨터는 다수의 서로 다른 질의에 대해 병렬로 클러스터별 2

차 검색을 수행하고 P-norm 모델을 이용하여 문서들의 순위를 계산한다.

본 연구에서는 이러한 병렬 정보검색 시스템을 구현하기 위해 1개의 루트 프로세서와 16개의 검색 프로세서, 그리고 4개의 하드디스크로 구성된 분산 메모리 MIMD 구조의 다중 트랜스퓨터 시스템을 사용하였다. 각 프로세서는 25MHz에서 동작하는 IMS T805 32bit 트랜스퓨터로 12개는 지역 하드디스크를 보유하고 있지 않고 4개는 각각 1GB의 지역 하드디스크를 보유한다. T805는 다른 프로세서와의 통신을 지원하기 위해 4개의 양방향 고속 통신 링크를 가지고 있으며, 인접 프로세서와 20Mbits/sec로 백그라운드 통신이 가능하다. 상호 연결 네트워크는 2차원 mesh with wrap-around를 사용한다.

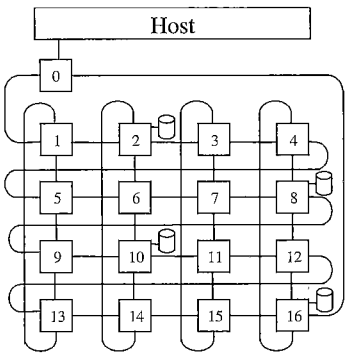

그림 5 다중 트랜스퓨터 기반의 병렬 정보검색 시스템

4. 클러스터링 기법의 적용

문서 라이브러리를 클러스터링하는 방법[2,7]은 크게 두 가지로 생각할 수 있다. 첫 번째 방법은 단순히 무작위적으로 문서들을 나누어 적정 크기의 클러스터를 만들어 가는 방법이다. 이 방법에서 클러스터들의 크기를 서로 비슷하게 하면 2차 검색을 위한 작업 할당 시 프로세서 간 부하 균등화를 쉽게 이룰 수 있다는 점에서는 유리하다. 그러나 무작위로 클러스터링을 할 경우 임의의 입력 질의어가 나타나는 문서들이 거의 모든 클러스터에 흩어져 포함되어 있을 가능성이 크게 되고, 이로 인하여 질의어가 등장하는 관련 클러스터들을 찾는 1차 검색의 결과로 대부분의 클러스터가 선택되는 문제가 있다. 즉, 부하의 균등화는 쉽게 가능하더라도 전체 부하가 불필요하게 커지는 것이다.

다른 한 방법은 기계학습 기법을 활용하여 각 클러스터가 유사한 문서들끼리의 집합이 되도록 클러스터링을 하는 것이다. 이렇게 하면 임의의 질의어와 관련된 문서들이 비교적 적은 수의 클러스터에 집중되어 있을 가능성이 높아지므로 1차 검색 결과로 선택되는 클러스터의 수가 줄어들고 따라서 2차 검색의 총 부하를 줄일 수 있게 된다. 아래에서는 먼저 여러 가지의 관련 기계학습 기법들을 소개한 다음, 이들을 문서 클러스터링에 의한 정보검색 병렬화 작업에 효과적으로 적용하기 위한 방안을 구체적으로 설명한다.

4.1 기계학습 기법

본 연구의 문서 클러스터링 문제에 적용 가능한 기계학습 기법은 크게 비감독 학습(unsupervised learning)과 감독 학습(supervised learning) 기법으로 나눌 수 있다. 감독 학습 기법[6,9]은 표본 문서의 집합을 대상으로 미리 유사한 것들끼리 분류를 해 둔 분류 예로부터 분류 규칙을 학습한 후 이 규칙을 적용하여 임의의 문서들을 분류하는 방법이다. 비감독 학습에 의한 클러스터링[11,14]은 문서들 사이의 유사도를 기준으로 그들 사이에 존재하는 원천적인 분류 구조를 찾아내는 방법으로, 미리 분류된 표본 문서 집합을 필요로 하지 않는다.

학습기법을 적용하기 위해서는 각 문서를 속성(attribute) 벡터로 표현하여야 한다. 일반적으로 임의의 문서 d_i의 속성 벡터 표현은 다음과 같이 된다.

$$d_i = (f_{1i}, f_{2i}, \ldots f_{ki}, \ldots f_{ni})$$

여기서 이 벡터의 차원에 해당하는 n은 문서 라이브러리에 등장하는 단어의 총 수이고, f_{ki}는 k 번째 단어가 문서 d_i에 나타나는 횟수이다. 문서 라이브러리 규모가 클 경우 n은 매우 커지며, n에 비해 한 문서에 등장하는 단어는 상대적으로 얼마 되지 않으므로 이 벡터의 대다수 엔트리는 0으로 되어 있다. 흔히 문서간의 유사도를 측정하는 척도로는 문서 벡터 사이의 유클리드 거리(Euclidean distance)를 사용한다.

■ 비감독 학습기법의 적용

본 연구에서 적용한 방법은 generalized Lloyd algorithm[10]을 기반으로 한 것이다. 이 방법에서 클러스터의 개수 M은 사람이 임의로 결정해 주어야 한다. 처음에 임의의 M 개 문서를 뽑아 클러스터의 중심(centroid)으로 삼은 다음, 나머지 모든 문서들에 대하여 최근접이웃(NN: Nearest Neighbor) 알고리즘을 적용하여 각 문서가 가장 가까운 클러스터로 할당되도록 한다. 그 다음에는 기 형성된 각 클러스터별로 할당된 문서 벡터들의 평균값을 구해 각 클러스터의 새 중심으로 삼고 이를 기준으로 다시 전체 문서의 클러스터 할당 작업을 반복한다. 이렇게 클러스터 할당에 이은 중심 갱신

과정을 계속 반복하여 중심의 변화가 거의 없어지면 중단한다. 이 알고리즘은 문서 라이브러리의 규모가 클 경우 계산 시간이 너무 길어진다. 따라서, 실제로 본 연구에서는 먼저 무작위적으로 추출된 일부 문서를 대상으로 이 알고리즘을 적용하여 클러스터들을 생성시킨 후, 나머지 전체 문서를 다시 NN알고리즘으로 가장 가까운 클러스터에 추가로 편입시키는 방법을 사용하였다. 이 방법의 가장 큰 장점은 사람이 개입하여 미리 표본 문서를 수작업으로 분류해 두어야 할 필요가 없다는 것이다.

■ **감독 학습기법의 적용**

감독 학습기법은 비감독 학습기법과는 달리 미리 분류된 표본 문서들을 필요로 한다. 이 방법은 문서 라이브러리의 규모가 커질수록 문서의 종류가 다양해지고 학습용 데이타의 규모도 커져야 하므로 수작업 분류 비용이 크게 증가하는 부담이 있다. 본 연구에서는 비감독 학습과 비교하여 클러스터링의 효과가 어떤지를 보기 위해 감독학습 기법도 적용해 보았다. 표본 문서 집합을 만드는 과정은 계층적으로 진행되었다. 즉 문서들을 처음부터 N개의 클래스로 분류한 것이 아니라 상위 개념인 몇 개의 클래스로 분류하고 이들 클래스들을 좀 더 상세히 분류해 나가는 방식으로 표본 문서 집합을 만들었다.

이렇게 미리 분류된 표본문서들로부터 분류규칙들을 유도해 내는 방안으로 본 연구에서는 결정트리(decision tree) 학습과 기억 기반(memory-based) 학습 기법을 적용하였다. 결정트리 학습 기법은 주어진 표본 데이타들을 정해진 클래스로 분류해 줄 수 있는 결정트리를 유도해 준다. 결정트리는 주어진 데이타의 속성값들을 소정의 순서에 따라 확인하여 클래스를 결정해 주는 트리이다. 본 시스템에서는 결정 트리 학습 도구로 Quinlan의 C4.5[9]를 사용하였다. C4.5로 대표되는 결정트리 학습기법은 현재 학습과 관련한 연구 및 응용분야에서 가장 널리 활용되고 있는 기법으로 대개의 실험적 연구에서 다른 기법들과의 비교 대상 혹은 비교 기준으로서의 역할을 하고 있다.

기억 기반 학습은 표본 데이타들을 메모리에 저장해 두고 유사도 측정 함수를 만들어 둠으로써, 새로운 데이타가 주어지면 그와 유사한 표본을 찾아서 그 표본의 클래스로 분류하도록 하는 방법이다. 이 방법은 앞에서 설명한 문서 벡터와 같이 수치 데이타에 적용하기 유리한 기법으로서 문서 분류 작업에 많이 활용되고 있다. 기억 기반 학습에서는 분류 과정에 표본 데이타들이 직접적으로 관여하므로, 표본 데이타들에 의해 분류의 정확도가 많은 영향을 받는다. 따라서 표본 데이타의 선정 과정에 세심한 주의가 필요하며 분류에의 기여도가 떨어지는 속성이나 노이즈 속성들을 제거하는 속성 선별 작업등을 거치는 것이 필요하다. 본 연구에서는 기억 기반 학습 기법으로 성능이 우수한 Relief-f 알고리즘[6]을 구현하여 실험하였다. Relief-f 알고리즘의 특징은 학습 과정에서 전체 표본 문서 대신 무작위로 선택한 일부 표본 문서들을 사용하여 속성들의 가중치를 조정하며, 보통의 NN(1-NN) 알고리즘이 아닌 k-NN을 사용한다는 점이다. k-NN은 가장 가까운 k 개의 이웃을 찾은 뒤 그 중 가장 많은 것들이 속한 클래스로 분류하는 알고리즘이다.

4.2 속성 선별 작업

일반적으로 학습기법에서 표본 집합의 수에 비해 속성의 수가 월등히 많은 경우에는 학습에 소요되는 시간 측면 뿐 아니라 학습의 정확도 면에서 좋지 못한 결과가 나올 가능성이 크다. 특히, 노이즈로 인해 잘못된 값을 가진 속성이 있다든지 분류를 하는데 오히려 방해가 되는 속성들이 포함되어 있을 경우 매우 부정적인 영향을 받게 된다. 문서의 분류 혹은 클러스터링을 위해 각 문서를 4.1절에서 설명한 바와 같이 단어들을 속성으로 한 벡터로 표현할 경우 속성의 수는 표본 집합의 수에 비해 엄청나게 많다. 따라서, 클러스터링에 앞서서 문서의 구분을 위해 중요한 단어 즉 속성들을 미리 선별하는 속성 선별 작업[3,5,18]이 선행되어야 좋은 결과를 얻을 수 있다.

본 연구에서는 학습에 앞서서 다음과 같은 과정을 거쳐 단어들을 선별하였다.

1. 형태소 분석기를 이용하여 조사나 어미 등에 해당하는 단어들을 제거한다.
2. 한자나 숫자, 기호 등에 해당하는 문자들을 제거한다.
3. 별 의미 없는 단어들을 제거대상리스트에 두어 제거한다.
4. 이상의 제거 과정 이후 남은 단어들 중에서 클러스터링에 보다 중요한 기여를 한다고 판단되는 단어들을 다음의 알고리즘에 의하여 추출한다.

1. 용어 정의

N : 전체 문서 수
k : 클러스터 수
m : 단어 수
$|C_i|$: i_{th} 클러스터에 속하는 문서의 수
n_{ij} : i_{th} 클러스터 내에서 j_{th} 단어가 등장하는 문서의 수
$n_{i'j}$: i_{th} 클러스터를 제외한 모든 클러스터에서 j_{th} 단어가 등장하는 문서의 수
$V_i(j)$: V_i 벡터의 j_{th} 요소

2. 알고리즘

for i = 1 to k

do

C_i에 속하는 문서 벡터들의 평균을 구하여 중심(centroid) 벡터 Cen_i를 만든다.

벡터 V_i를 구한다.

$$V_i = (\frac{n_{i1}}{|C_i|}, \ldots, \frac{n_{ij}}{|C_i|}, \ldots, \frac{n_{im}}{|C_i|})$$

벡터 V_i'를 구한다.

C_i의 각 단어의 가중치는 다음과 같이 결정된다.

for j = 1 to m

$$V_i' = (\frac{n_{i1}}{N-|C_i|}, \ldots, \frac{n_{ij}}{N-|C_i|}, \ldots, \frac{n_{im}}{N-|C_i|})$$

$$w_{ij} = Cen_i(j) \times V_i(j) \times (1 - V_i'(j))$$

C_i에서 가중치가 높은 상위 x개의 단어를 선정한다.

done

위의 알고리즘에서는 세 가지 요소가 고려되고 있다. 먼저 클러스터 내에 속하는 문서 벡터를 평균한 중심 벡터를 이용하는데 이는 그 클러스터 내에서 단어들의 평균 빈도수를 고려하기 위한 것으로 평균 빈도수가 높은 단어일수록 중요하다고 판단한다. 두 번째 요소는 한 클러스터 내에 각 단어가 등장하는 문서의 수를 이용하는데 이는 많은 문서에 등장하는 단어일수록 가중치를 높게 주기 위한 것이다. 마지막으로 다른 클러스터 내에는 그 단어가 나타나지 않는 문서가 많을수록 좋다고 판단하여 그런 성질을 지닌 단어를 중요하게 취급한다.

속성 선별 작업이 끝나면 이 과정을 거쳐 선택된 단어들만을 이용하여 다시 문서 벡터를 표현하고 앞의 4.1절에서 소개한 학습 기법들을 적용하여 클러스터링을 수행한다.

4.3 클러스터 크기의 재조정

앞의 3.2절에서 이미 설명하였듯이 본 연구의 병렬 정보검색 시스템에서는 클러스터가 작업의 할당 단위이며 부하 균등화의 단위가 된다. 따라서, 만약 클러스터들 사이의 크기 차이가 현저하게 된다면 문서 검색 시 프로세서 간 부하의 균형을 이루기가 어려워질 것이다. 표 1은 4.1절에서 소개한 여러 방법들로 클러스터링을 했을 때 만들어지는 클러스터들의 크기를 분석한 것이다. 표 1에 보인 바와 같이 무작위적으로 클러스터링한 경우는 표준편차가 18로서 클러스터의 크기들이 거의 고른 편이다. 그러나 나머지 유사 문서끼리 클러스터링한 방법들의 결과를 보면 가장 큰 클러스터는 몇 백 개 정도의 문서들로 이루어져 있는데 비해 가장 작은 클러스터는 두세 개의 문서들로 이루어져 있는 등 클러스터들 간의 크기 편차가 현저하다.

표 1 각 학습기법을 적용한 결과의 클러스터들 크기

	C4.5	Relief-f	무작위	비감독
클러스터 개수	121	124	125	140
표준편차	73.1	133.7	18.0	58.8
최소크기	2	6	36	2
최대크기	432	1304	125	294

이러한 클러스터의 크기 차이는 문서 검색 시 프로세서 간 부하 불균형 문제를 초래하므로 검색 성능 저하의 원인이 된다. 따라서 본 연구에서는 일차로 클러스터링한 결과를 분석한 후 클러스터들의 세분화 혹은 병합을 통해 그 크기를 재조정함으로써 이 문제를 해결하였다. 적정 크기 이상의 클러스터들은 무작위적으로 문서들을 나누어 원하는 크기의 클러스터들로 세분화하였다. 클러스터의 병합은 크기가 지나치게 작은 클러스터를 찾은 뒤 그것을 그와 가장 유사한 클러스터에 흡수시키는 방식으로 하였다. 감독 학습기법으로 클러스터링한 경우에는 문서들을 수작업으로 분류할 때 형성한 계층 구조(4.1절 참조)에 근거하여 서로 유사한(즉, 계층 구조에서 서로 가까이 위치한) 클러스터들끼리 병합이 이루어지도록 하였다. 비감독 학습기법으로 클러스터링한 경우에는 NN 알고리즘에 의하여 클러스터 중심들 사이의 유사성을 측정하여 가장 가까운 클러스터를 선정한 후 이 클러스터에 병합시켰다. 표 2는 이러한 기준에 의하여 재조정한 후의 클러스터 크기를 분석한 것이다.

표 2 클러스터 크기 재조정 후의 분석

	C4.5	Relief-f	비감독
클러스터 개수	117	118	117
표준편차	24.22	32.46	26.8
최소크기	31	23	30
최대크기	140	166	164

5. 실험 결과

실험 대상 문서로서는 부산일보사에서 제공한 95년도 신문기사들의 전자문서 모음집에서 발췌한 9,593개의 문서를 사용하였다. 감독 학습을 위해서는 전체 문서 중에서 2,160개의 문서를 선택하여 수작업으로 125개의 클래스로 분류시켜서 학습용 표본 문서 집합으로 삼았다. 이 2,160개의 표본 문서 집합에 등장하는 총 단어 수는 42,557개이나, 4.2절에서 제시한 속성 선별 알고리즘을 적용하여 약 2,000개로 단어 수(즉, 속성 수)를 줄인 후 학습시켰다.

본 연구에서 제안한 문서 클러스터링의 효과를 확인하기 위하여, 4.1절에서 소개한 비감독 학습기법, 감독 학습기법 중 C4.5와 Relief-f, 그리고 무작위적인 클러스터링 기법까지 모두 적용해서 비교 실험해 보았다. 그리고, 최근 문서 검색의 한 주류는 시소러스나 관련성 피드백(relevance feedback)[12]을 통하여 질의어를 확장시켜 검색하는 것이므로 본 실험에서도 질의어 수를 1에서 11개까지 증가시켜 가며 실험하였다. 질의어 확장은 1개의 질의어로 검색했을 때 찾은 문서들 중 가장 높은 점수를 가지는 상위 3개의 문서에 등장하는 단어들을 추가하는 방법을 사용했다.

5.1 클러스터 크기의 영향

그림 6은 대표적으로 C4.5을 이용해 클러스터링한 경우, 클러스터 크기의 재조정 전과 후의 검색 시간을 무작위적으로 클러스터링한 경우와 비교하고 있다. 클러스터 크기 재조정을 하지 않은 경우는 부하균등화가 잘 이루어지지 않아서 질의어 수가 늘어남에 따라 무작위적으로 클러스터링한 경우보다 검색 시간이 더 길어지는 것을 볼 수 있다. 이는 무작위적인 방법의 경우 질의어의 수에 관계없이 항상 많은 클러스터들이 검색 대상이 되므로 질의어 수가 늘어나더라도 검색 대상이 되는 클러스터의 수가 상대적으로 크게 늘어나지 않으며 또한 클러스터들 사이의 크기에 편차가 거의 없어서 프로세서간 부하 균등화의 측면에서 C4.5보다 더 유리하기 때문이다. 그러나 클러스터 크기를 재조정하여 클러스터 크기 편차를 무작위적으로 클러스터링한 수준으로 조정한 경우에는 질의어 수에 상관없이 C4.5 방식이 검색 시간 면에서 더 우수하였다. 이는 프로세서 간 부하균등화 측면에서 유사한 조건으로 출발할 경우, 무작위적으로 클러스터링하는 것보다 기계 학습기법을 활용하여 클러스터링하는 것이 보다 유사한 문서군들의 집합이 되게 하므로, 주어진 질의와 관련하여 2차 검색 대상이 되는 클러스터의 수가 줄어들어서 전체적으로 검색 부하가 경감되기 때문이다.

5.2 클러스터링 기법별 성능 비교

문서 클러스터링에 적용한 기계 학습기법들 중 비감독 학습기법은 표본 문서 집합을 필요로 하지 않으며 속성 선별 작업을 거치지 않아도 비교적 우수한 성능을 보이는 장점을 가지고 있음이 확인되었다. 학습하는데 걸리는 시간면에서도 비감독 학습 기법이 가장 빨랐으며, relief-f, C4.5의 순이었다. C4.5나 relief-f는 속성 선별 작업 없이 학습을 하게 되면 아주 많은 시간이 걸렸으나 속성 선별 작업을 통하여 학습 시간은 상당히 단축되었다.

그림 7은 세 가지의 학습기법에 의한 클러스터링을 이용한 검색 결과를 무작위적 클러스터링을 이용한 경우와 비교한 것이다. 세 방법 모두 무작위적 방법보다 우수하며 검색 시간에 있어서 거의 비슷한 성능을 보이고 있다. 단, 질의어 수가 많은 경우 C4.5를 이용한 방법이 다른 두 방법에 비해 약간 더 빠르게 나타난다. 어떤 방법이든 질의어 수가 늘어남에 따라 검색 시간 면에서 무작위적 방법과의 차이가 줄어드는 것을 볼 수 있다. 이는 무작위적 방법의 경우 질의어 수에 관계없이 검색 대상이 되는 클러스터의 수에 별 차이가 없으나 (항상 많음) 학습 기법을 이용한 경우에는 질의어 수가 늘어남에 따라 2차의 문서 검색 대상이 되는 클러스터들 수도 늘어나기 때문이다.

그림 8은 기존의 병렬 정보 검색 알고리즘인 Stanfill 방법과 본 연구의 클러스터링기법 중 대표로 C4.5를 이용한 경우의 검색시간을 비교하고 있다. 그림 8에서 질의어 수에 상관없이 본 연구의 방식이 Stanfill 방식보다 검색시간 면에서 우수함을 알 수 있다. 이는 계층적 접근 방식의 특징인 여과 효과에 기인하는 것으로 1차 검색 후 그다지 점수가 좋지 않은 클러스터들을 2차 검

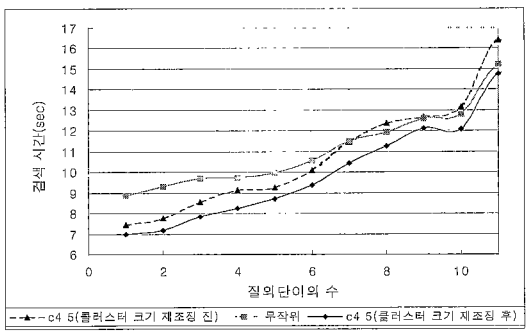

그림 6 클러스터 크기 재조정 전후의 검색 시간

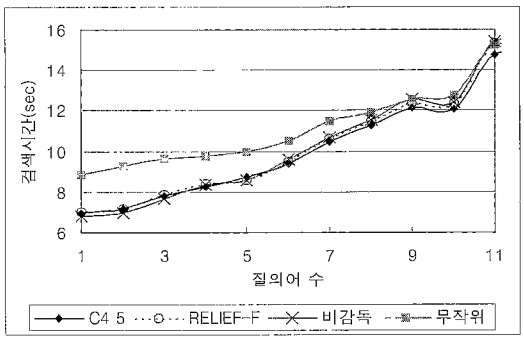

그림 7 여러 학습기법 적용 결과의 검색 시간 비교

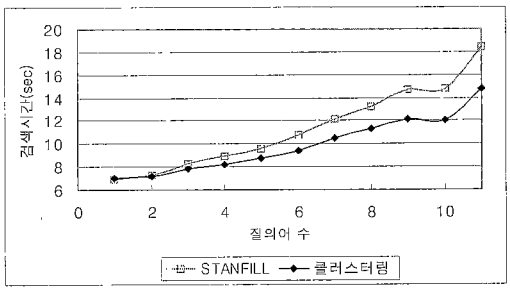

그림 8 Stanfill 방법과의 비교

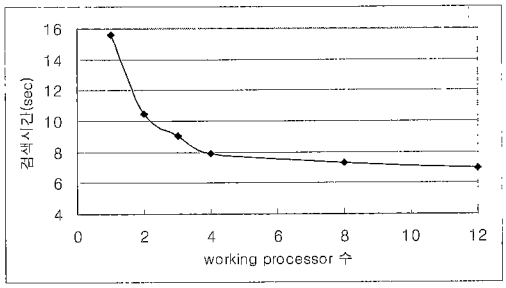

그림 9 프로세서 수에 따른 검색 시간

색 대상에서 제외시켜 2차 검색의 부담을 줄일 수 있기 때문이다. 그러나 이러한 여과 과정으로 인해 실제로 질의어에 부합하는 좋은 문서이더라도 평가 값이 낮은 클러스터에 속하는 경우에는 검색되지 못하는 경우가 생겨서 검색의 질을 저하시킬 수 있다. 여과의 효능을 평가하기 위하여 질의어를 포함하는 모든 문서들을 검색하는 Stanfill 방법에 대해 검색되는 상위 50개 문서의 일치도를 측정해 보았다. 측정 결과 학습기법을 이용한 경우에는 평균 90%이상의 문서가 일치하였고 무작위적으로 클러스터링한 경우는 평균 80%정도의 문서만 일치하였다. 이 결과를 통하여 우리는 학습기법을 활용한 클러스터링 및 계층적 색인어 역파일을 두는 접근법에서 여과가 검색의 질에 큰 영향을 주지 않으면서 검색 성능을 향상시킴을 알 수 있다. 최근에는 여러 통계적 기법을 이용하여 문서 라이브러리에서 아주 많은 정보들을 추출해 내고 있다. 이러한 정보들을 활용하여 학습기법을 적용한다면 더욱 더 유사도가 잘 반영된 클러스터링이 가능하게 되고 이로 인해 여과의 효과 또한 훨씬 높일 수 있을 것이다.

5.3 프로세서 수에 따른 성능 향상

본 연구에서 사용하는 16개의 검색 프로세서 중 지역 디스크를 가지고 있는 4개의 프로세서를 제외한 12개의 작업 프로세서를 대상으로 그 수를 증가시켜 나갈 때의 검색 속도 향상 효과를 관찰하였다. 그림 9는 질의단어의 수가 1일때의 검색 결과이다.

작업 프로세서 수 증가에 따라 성능은 향상되고 있으나 4를 기점으로 하여 그 향상 정도가 둔화되는 것을볼 수 있다. 작업 프로세서 수가 1에서 4까지 증가하는 동안에 특히 성능 향상이 큰 이유는, 로컬 디스크가 4개까지 늘어나는 동안은 검색 대상 클러스터의 문서 레벨 색인어 역 파일을 읽어와 작업 프로세서들에게 나누어 주는 과정의 병렬도가 그만큼 같이 증가하는 효과로 이어지기 때문이다. 프로세서 수 4 이후 향상 정도가 둔화되는 것은 디스크의 수가 4에 묶여 있고, 작업을 많은 수의 프로세서에 할당하는 과정에서의 네트워크 통신 비용이 증가하기 때문이다.

6. 결론 및 향후 과제

본 논문에서는 병렬 정보검색 시스템을 구현하는데 있어서 문서를 클러스터링하고 계층적인 색인어 역파일을 이용하는 방안을 제시하였다. 문서의 클러스터링은 다중 트랜스퓨터 시스템과 같은 분산메모리 MIMD 구조의 병렬 컴퓨터 상에서 검색과정의 병렬화 및 부하균등화가 용이하도록 해준다. 특히, 학습기법들을 이용하여 문서 라이브러리를 클러스터링할 경우 무작위적으로 클러스터링하는 경우에 비해 보다 유사한 문서군들로 클러스터가 형성됨으로 인해 질의 관련 클러스터의 수를 줄임으로써 전체 검색의 부하를 경감시킬 수 있음을 확인하였다.

클러스터 레벨과 문서 레벨의 계층적인 색인어 역파일을 통하여 이루어지는 검색은 1차 검색 후 여과가 가능하여 2차 검색의 부담을 줄임으로써 전체적인 검색 소요 시간을 단축시킬 수 있게 해 준다. Stanfill 방법과의 비교 실험 결과, 문서 클러스터링에 따른 계층적인 접근법은 질의어 수가 늘어남에 따라 검색 시간이 훨씬 더 짧아지는 것을 볼 수 있었다. 이는 문서 검색의 추세가 관련성 피드백을 보다 적극적으로 활용하는 방향으로 나아가고 있다는 관점에서 볼 때 본 연구의 접근법이 상당한 이점을 지닌 것임을 보여주는 것이라 하겠다. 또한 프로세서 수에 따른 속도 개선 실험을 통하여 본 연구의 병렬 정보 검색 모델이 대규모 시스템으로 확장 가능함을 입증하였다.

본 논문에서 제시한 계층적 검색 구조는 여과 과정이 가지는 위험성, 즉 좋은 문서라 할지라도 평가값이 낮은 클러스터에 속하는 문서의 경우 검색 대상에서 제외되는 문제점을 지니고 있다. 이와 관련해서는 벤치마크 데

이타를 이용하여 정확률과 재현률(precision & recall)을 동시에 고려한 재평가 작업이 뒤따라야 할 것이다. 현재 본 연구진은 본 논문의 클러스터링 기법을 확장하여 PC 클러스터 환경에서 동작하는 효율적인 병렬 정보검색 시스템을 개발하기 위한 연구를 진행 중에 있다. 향후 통신 부담을 최소화하여 분산 환경에서의 정보검색을 효율적으로 하는 방안에 관한 연구도 필요할 것으로 생각된다.

참 고 문 헌

[1] Cahoon, B. and McKinly, K. S., "Performance Evaluation of a Distributed Architecture for Information Retrieval," *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[2] Cohen, W. W. and Singer, Y., "Context-Sensitive Learning Methods for Text Categorization," *Proceedings of The Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 307-315, Zurich, Switzerland, 1996.

[3] Dietterich, T. G., Machine Learning Research: Four Current Directions, *Artificial Intelligence,* pp 1- 64, 1997.

[4] Fox, E. A., "Extending the Boolean and Vector Space Models of Information Retrieval with P-norm Queries and Multiple Concept Tyes," *Ph D. dissertation*, Cornell University, 1983.

[5] Kira, K. and Rendell, L. A., "A practical approach to feature selection," *Proceedings of The Ninth International Conference on Machine Learning,* pp. 249-256 San Francisco, CA. Morgan Kauffman, 1992.

[6] Kononenko, I., "Estimating attributes: Analysis and extensions of relief," *Proceedings of The 1994 European Conference on Machine Learning,* pp.171-182 Amsterdam. Springer Verlag, 1994.

[7] Liere, R. and Tadepalli, P., "Active Learning with Committees for Text Categorization," *Proceedings of The Fourteenth National Conference on Artificial Intelligence*, pp. 591-597, Providence, Rhode Island, 1997.

[8] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T., "Learning to Classify Text from Labeled and Unlabeled Documents," *Proceedings of The Fifteenth National Conference on Artificial Intelligence*, pp.792-799, Madison, Wisconsin, 1998.

[9] Quinlan, J.R., C4.5 Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, California, 1993.

[10] Rabiner, L. R., and Juang, B. H., Fundamentals of Speech Recognition, Prentice- Hall, Englewood Cliffs, New Jersey, 1993.

[11] Sahami, M., Yusufali, S., and Baldonado, M. Q. W., "Real-time Full-text Clustering of Networked Documents," *Proceedings of The Fourteenth National Conference on Artificial Intelligence*, pp.845, Providence, Rhode Island, 1997.

[12] Salton, G. and Buckley, C., "Improving Retrieval Performance by Relevance Feedback," *Journal of the American society for Information Science*, pp. 88-297, 1990.

[13] Sang-Hwa Chung, Soo-Cheol Oh, Kwang Ryel Ryu, Soo-Hee Park, Parallel Information Retrieval on a Distributed Memory Multiprocessor system, Proceedings of the Third International Conference on Algorithms and Architectures for Parallel Proceeding(ICA3PP-97), pp. 163-176, Melbourne, Australia, 1997.

[14] Schutze, H., & Silverstein, C. "Projections for Efficient Document Clustering," *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in information Retrieval,* pp. 74-81, 1997.

[15] Sharma, R., "A Generic Machine for Parallel Information Retrieval," *Information Proceeding and Management*, Vol. 25, No. 3, pp. 223-235, 1989.

[16] Smith, M.E., "Aspects of the p-norm model of information retrieval : syntactic query generation, efficiency, and theoretical properties," *Ph.D. thesis*, Cornell University, 1990.

[17] Stanfill, C. and Thau, R., "Information Retrieval on the Connection Machine : 1 to 8192 Gigabytes," *Information Processing & Management*, pp.285-310, 1991.

[18] Wettschereck, D. and Aha, D. W., "Weighting Features," First International Conference on Case-Based Reasoning, 1995.

[19] 박 수희, 정 상화, 류 광렬, 병렬 정보검색 시스템에서의 부하평준화 기법, 한국정보과학회 '97 가을학술발표논문집(IV), Vol. 24, No. 2, pp. 385-387, 1997.

[20] 박태완, 류광렬, 정상화, 동적문서할당 기법을 적용한 병렬 정보검색, 정보과학회논문지 C 제4권 제2호, pp. 219-227, 1998.

강 유 경

1996년 부산대학교 컴퓨터공학과 학사. 1999년 부산대학교 컴퓨터공학과 석사. 1999년 ~ 현재 통계청 전산사무관. 관심분야는 인공지능, 기계학습, 정보검색, 데이타마이닝 등임.

류 광 렬

1979년 서울대학교 전자공학과 학사. 1981년 서울대학교 전자공학과 석사. 1983년 3월 ~ 1984년 8월 충북대학교 컴퓨터공학과 전임강사. 1992년 University of Michigan 컴퓨터 공학 박사. 1992년 3월 ~ 1993년 2월 Scientific Research Lab., Ford Motor Company, 선임연구원. 1993년 3월 ~ 현재 부산대학교 컴퓨터공학과 부교수. 관심분야는 인공지능, 스케줄링, 기계학습, 정보검색, 데이터마이닝 등임.

정 상 화

1985년 서울대학교 전기공학 학사. 1988년 Iowa State University 컴퓨터공학 석사. 1993년 University of Southern California 컴퓨터공학 박사. 1993년 ~ 1994년 University of Central Florida 전기 및 컴퓨터공학과 조교수. 1994년 ~ 현재 부산대학교 컴퓨터공학과 부교수. 관심분야는 클러스터 시스템, 병렬처리, VOD, 정보검색.