HoleInOne 메타검색 시스템의 설계 및 구현

김현주[†] · 배종민[††]

요 약

본 논문에서는 제안된 관련성 분포 정보(Relevance Distribution Information: RDI)를 이용하여 메타검색 시스템을 제안한다. 이는 먼저 주어진 질의에 대하여 검색에 참여한 정보원(source)을 평가하고 질의에 가장 적합한 정보원을 선택한다. 그리고 정보원의 평가 결과에 따라 해당 정보원으로부터 검색 문서를 차별적으로 수집하고, 검색된 문서들은 정보원의 평가 값인 *RDI*를 기반으로 최종 검색 문서의 순위 매김을 수행한다. 이렇게 순위 매김 된 검색 문서는 단일 우선 순위를 가지는 검색 문서의 집합으로 수집하여 사용자에게 단일 검색 결과를 제공한다. 이를 위해 본 논문에서 질의와 정보원 사이에 대한 RDI를 표현할 수 있는 평가요소들을 설계하고, 이들 평가 요소를 기반으로 RDI를 추출하는 방법을 제안하였다. 그리고 질의에 대하여 가장 좋은 정보원들을 분류할 수 있는 체계를 개발하여 사용자의 질의에 대하여 최선의 정보원들을 선택할 수 있는 알고리즘을 제시하였다. 마지막으로 선택된 정보원으로부터 질의에 적합한 문서를 검색한 후에 이들을 순위 매김하고 수집하는 *HoleInOne*(*wHOLE INformation ONEtime*) 메타검색 시스템을 제시한다.

Design and Implementation of *HoleInOne* Metasearch System

Hyun-Ju Kim[†] and Jong-Min Bae[††]

ABSTRACT

The Meta Search system proposed in this paper is operated based on relevance distribution Information(RDI). It first evaluates the sources applicable to the search, and then selects the most appropriate source. According to the evaluation of the sources, it discreetly collects the documents from the concerned sources and classifies them into a useful order based on the RDI, which is an evaluation score of the sources. The documents are classified into order and presented to the user as a single search result. For this purpose, this study presents evaluation factor models to present the RDI between the query and source, and proposes a method for drawing out the RDI based on the evaluation factors. The system for selecting the most appropriate sources according to the query has been developed based on an algorithm that selects the best source. Finally, after searching the documents suitable for query from extracted sources, we present a Meta Search system, HoleInOne, that ranks and merges them.

Key words: 메타검색, 통합검색, 정보검색, 메타 데이터, 디지털 도서관, Collection Selection, Collection Fusion

1. 서 론

최근 Yahoo[19], InfoSeek[20] 등과 같은 정보검색 기들은 폭발적으로 늘어나는 정보를 자신의 컴퓨터에 저장하는 중앙 집중식 정보 관리법에 한계를 느끼고 있으며, 그 결과 InfoSeek[20]에서는 통합검색기인 InfoSeek Patent를 개발하여 실험적으로 운영하고 있다. 메타검색기는 분산, 병렬검색기에 속한다. 이에, 한글검색엔진을 효과적으로 지원하고, 나아가서 지능을 갖춘 강력한 통합 정보검색기를 개발하는 것은 정보검색 분야의 학문과 기술의 발전은 물론이고, Yahoo[19]와 같은 정보검색 서비스 사업에 경쟁력을 갖추는 것은 산업의 발전에도 중요한 과제라 할 수 있다.

접수일 : 2002년 11월 6일, 완료일 : 2002년 12월 10일
[†] 정회원, 진주산업대학교 컴퓨터공학과 전임강사
[††] 정회원, 경상대학교 컴퓨터과학과 교수

메타검색기의 구성 요소들에 대한 부분적인 연구는 데이터베이스 분야와 정보검색 분야에서 오래 전부터 있어 왔으나, 최근 인터넷과 디지털 라이브러리 개발이 성숙되면서 웹 상의 많은 검색엔진에 대하여 사용자에게 하나의 통합된 검색엔진이라는 관점을 제공하는 노력이 이루어졌다. 그 결과 메타검색기 혹은 통합검색기라는 이름으로 디지털 라이브러리 개발사업의 주요과제 중의 하나가 되었다. 그 중에서 Harvest는 인터넷상의 정보를 모으고 검색하는 도구로서, Harvest gather는 다수의 정보원에서 색인정보를 모으고, broker는 질의 인터페이스를 제공하고 정보를 검색한다[1,2]. 이는 메타검색기 연구의 출발점으로 좋은 참고 시스템이다. 지금까지 개발된 메타검색기로는 SavvySearch[18], ProFusion[1,17], Infoseek Patent, Inference Find, STARTS[2,3] 등이 있으며, NCSTRL의 기반이 되는 Dienst도 동질의 정보원에 대한 메타검색기라 할 수 있다. 이들은 제각기 제한된 범위 내에서 특징을 가지고 있으나, 메타검색의 핵심적인 주요 연구 주제에 대하여 만족스럽게 해결한 시스템은 없으며, 모두가 현재로서는 실험적인 시스템이라 할 수 있다[1,12,14-16].

이러한 메타 검색 분야에서 사용자들의 질의에 대하여 효율적인 검색 결과를 얻기 위해서 주로 연구되고 있는 분야는 크게 세 가지로 구분할 수 있다. 첫 번째는 질의에 대해 가장 좋은 정보원을 선택하는 문제이다. 이는 메타검색 시스템이 검색에 참여시키고 있는 수많은 이질의 정보원 중에서 사용자의 질의어를 만족시킬 수 있는 가장 좋은 정보원들을 자동으로 결정하는 방법에 대한 것이다. 두 번째는 질의어 자동번역 문제이다. 메타검색 시스템에서 질의는 가장 적합한 정보원을 선택한 후에 해당 정보원에서 자동적으로 질의를 수행하게 된다. 그러나 검색에 참여한 이질의 정보원은 서로 다른 질의 문법을 가지고 있어서 메타검색 시스템에서 생성된 질의를 직접 인식하지 못한다. 따라서 이들을 자동 번역하는 질의어 번역기가 필요하다. 마지막으로는 검색 문서의 수집 및 순위 매김을 처리하는 문제이다. 메타검색 시스템은 입력된 질의에 대하여 분산된 이질의 정보원으로부터 검색 결과를 수집한다. 그리고 문서에 대하여 순위 매김을 수행하고, 이를 단일 검색 결과로 생성한 후에, 사용자에게 검색 결과로 회신한다. 이러한 메타검색 시스템의 세 가지 연구 분야는 메타검색 시스템의 검색 능력에 많은 영향을 미치며, 또한 검색을 수행할 때 상호 연관되어 동작한다[12,13,15,16].

따라서 본 논문에서는 메타검색 시스템에서 핵심적인 정보로 사용될 관련성 분포정보(*RDI* : Relevance Distribution Information)를 추정하는 새로운 모델을 제안하고, 이를 기반으로 정보원 선택(collection selection), 검색결과 통합(collection fusion) 등의 알고리즘을 제시한다. 그리고 제안된 알고리즘 기반으로 구현한 *HoleInOne* 메타검색 시스템으로 실험 결과를 분석한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구를 살펴본다. 그리고 3장에서는 본 논문에서 제안한 *RDI*를 기반으로 설계한 *HoleInOne* 메타 검색 시스템의 알고리즘을 살펴본다. 그리고 4장에서는 제안된 메타검색 시스템의 개괄구조와 구현된 객체 및 메소드에 대한 기능들을 살펴보고, 5장에서는 제안된 메타검색 시스템의 실험결과를 소개한다. 마지막으로 6장에서는 결론 및 향후연구 방향을 소개한다.

2. 관련 연구

이 장에서는 기존 메타검색 시스템에 대해 살펴본다. 첫 번째는 Voorhees[2,3]외 2명이 제안한 메타검색 모델을 실험한 SMART 시스템이고, 두 번째는 Callan 외 3명이 제안한 모델을 평가하기 위해 개발한 INQUERY[1,13] 시스템에 대하여 기술한다. 마지막으로 미국 캔자스 대학 Susan Gauch외 2명이 개발한 ProFusion[6,17] 시스템 등이다.

2.1 SMART 검색시스템[2,3]

SMART 시스템은 벡터공간 모델을 기반으로 하며, 기본적으로 색인, 검색, 평가 등에 대한 기능을 제공해 준다. Voorhees[2,3]외 2명이 제안한 메타검색 통합모델은 질의와 검색에 참여한 정보원과의 유사도 값을 추정하며, SMART 시스템을 기반으로 제안한 통합검색 모델의 성능을 평가하였다[2,3]. 이때 정보원에 대한 유사도 값을 추정하는 방법으로는 문서의 관련성 분포 정보와 질의 클러스트링 정보를 이용하였다.

먼저, 문서의 관련성 분포 정보를 이용하는 방법은 질의들을 학습시켜 각 정보원에 대해 질의의 유사도 값을 평가하고, 저장한다. 만약 새로운 질의가 주어지면 질의와 유사한 k개의 학습된 질의를 추출하여 이들이 가지고 있는 유사도 값들의 평균값을 새로운 질의

에 대한 정보원의 유사도 값으로 추정하는 방법이다. 다음으로는 질의들의 클러스트링 정보를 이용하여 정보원을 추정하는 방법이다. 이는 앞의 방법과 동일하게 미리 질의들을 학습시켜 질의와 정보원 사이의 유사도 값을 평가한다. 이렇게 학습된 질의들은 공통된 검색 문서의 빈도 수에 따라 질의들을 클러스트링 하며, 이들은 각각의 유사도 값들을 평균값으로 해당 정보원에 대한 유사도 값으로 추정하고 이를 중심 값이라 한다. 만약 새로운 질의가 입력되면 먼저 유사한 학습 질의를 찾고, 이 질의가 속해 있는 클러스트링의 중심 값을 새로운 질의에 대한 정보원의 유사도 값으로 평가하는 방법이다.

2.2 INQUERY 검색시스템[1,13]

INQUERY 시스템은 Callan[1,13]외 3명이 제안한 메타검색 통합모델을 실험·평가하기 위해 구현되었다. 이는 CORI net (***CO***llection ***R***etrieval ***I***nference ***net***work) 검색 모델이라고도 하며, 문서, 정보원과 질의 사이의 관련성을 *df*와 *icf*를 기반으로 유사도를 평가한다. 또한 질의와 정보원내의 문서 사이에 대한 관련성을 문서 네트워크와 질의 네트워크로 분류하여 관련성 정보를 표현한 모델이다. CORI net 모델에서는 주어진 질의에 대하여 가장 적합한 정보원을 선택하기 위해 term과 *df*를 기반으로 정보원의 유사도 값을 평가한다.

2.3 ProFusion 검색시스템[6,17]

ProFusion 메타검색 시스템은 미국 캔자스 대학의 Susan Gauch외 2명이 제안한 모델을 평가하기 위해 구현된 검색시스템이다[6,17]. 이 메타검색 시스템은 9개의 일반 검색 엔진을 대상으로 질의를 수행하고 이들로부터 검색 결과를 수집하여 통합검색 결과를 사용자에게 인터넷 주소로 보여준다. ProFusion 메타 검색 시스템에서는 사용자의 질의에 대하여 9개의 정보원을 선택하는 방법으로는 (1) 최상의 3개 정보원을 선택하는 방법, (2) 가장 빠른 검색 결과를 보여주는 3개의 정보원을 선택하는 방법, (3) 9개의 정보원 모두다 사용하는 방법, (4) 사용자가 정보원을 선택하여 사용하는 방법 등 4가지 기능을 제공하며, 본 논문에서는 첫 번째 방법에 대해서만 다룬다. ProFusion 메타검색 시스템은 질의에 가장 적합한 3개 정보원을 선택하기 위해, 미리 질의 후보들에 대하여 정보원을 평가하고 이를 기반으로 정보원을 선택할 수 있는 신뢰도(*CF*: Confidence Factor) 정보를 생성한다. 이를 데이터베이스 정보로 구축하여 새로운 질의가 발생될 때 이를 사용한다.

최상의 검색 엔진을 선택하기 위해 뉴스 그룹에서 사용하는 도메인 네임으로부터 13개의 카테고리를 선정하여 이를 질의에 대한 분류로 사용하였다. 그리고 이들 뉴스 그룹으로부터 4,000 개의 유일한 Term 후보들을 추출한 후에 이들 Term이 카테고리내의 문서에 포함되어 있는 문서의 발생 빈도 수에 대한 정보를 지식 데이터베이스로 구축한다. 이러한 지식 데이터베이스 정보는 새로운 질의가 발생될 때 정보원 선택에 대한 *CF* 값으로 사용된다. 이렇게 평가된 값을 기반으로 ProFusion 메타 검색기에서는 주어진 질의에 대해 최상의 3개 컬렉션을 선택한다.

3. HoleInOne 메타검색 시스템의 설계

이 장에서는 본 논문에서 제안한 관련성 분포 정보(RDI) 기반의 메타검색 시스템에 대해 기술한다. 먼저, 3.1절에서는 제안된 메타검색 시스템의 개괄 처리과정을 살펴보고, 3.2절에서는 메타검색 시스템을 설계할 때 고려한 기본적인 사항들을 살펴본다. 그리고 3.3절에서는 본 논문에서 제안한 RDI의 생성과정에 대해 살펴보고, 3.4절과 3.5절에서는 RDI 평가 값을 적용한 정보원 선택, 검색결과 통합 알고리즘에 대해 각각 기술한다.

3.1 제안된 메타검색 시스템의 개괄구조

먼저 본 논문에서 제안한 메타검색 시스템은 RDI를 기반으로 설계하였다. RDI는 사용자가 검색을 하기 위해 질의하는 검색어와 정보를 제공해주는 정보원 사이의 유사성을 나타낸 통계적 추정 값이다. 이에 대한 자세한 사항은 3.3절에서 기술하였으며, 이를 적용하여 설계한 HoleInOne 메타검색 시스템의 동작원리는 그림 1과 같다.

제안된 메타검색 시스템의 처리과정은 화살표로 표시하였으며, 전달되는 값 또는 매개 변수들은 화살표 위의 주석으로 표시하였다. 제안된 메타검색 시스템은 크게 3가지의 주요 기능으로 구분할 수 있다. 이들은 RDI 평가, 정보원선택, 검색결과 통합 등이다. 첫 번째로, RDI 평가는 사용자의 검색어와 정보원사이의 유사

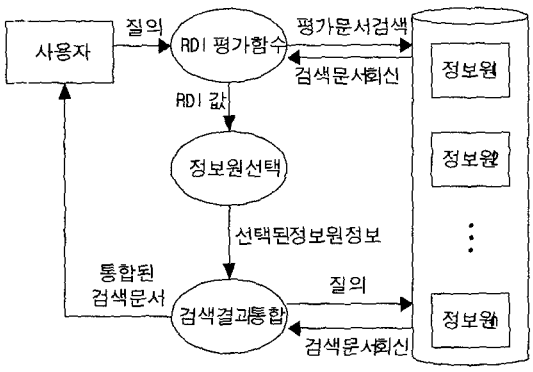


그림 1. HoleInOne 시스템의 처리과정

도를 추정하는 과정이다. 먼저 본 논문에서의 RDI 평가는 실시간으로 이루어진다. 사용자가 질의를 실행하면, 주어진 검색어로 정보원의 RDI를 평가를 위한 모집단 검색문서를 수집하고, 수집된 검색문서들로부터 본 논문에서 제안된 메타데이터를 추출하여 RDI를 평가하였다. 이때 RDI 평가를 위해 정의된 메타데이터는 3.3절에서 소개한다. 두 번째는 정보원 선택이다. 정보원에 대한 RDI 평가 추정치는 질의에 대한 정보원의 관련성 정도를 나타내는 값이다. 따라서 이에 대한 RDI 추정치가 크면 클수록 좋은 검색결과를 얻는다고 가정할 수 있다. 본 논문에서는 정보원의 추정치 값을 상대 비율로 변환하여 정보원 선택 기준으로 사용하였으며, 또한 각 정보원으로부터 검색될 문서의 크기도 결정하였다. 마지막으로 검색결과 통합이다. 이는 다양한 정보원으로부터 검색문서를 단일 우선 순위를 가지는 검색 집합으로 생성하는 것이다. 그런데 서로 다른 정보원으로부터 검색된 문서들은 우선 순위를 결정하기 위해 상대적으로 비교할 수 없다. 그 이유는 검색에 참여한 정보원이 서로 다르며, 이질적인 검색결과 생성알고리즘을 사용하고 있기 때문이다. 따라서 본 논문에서는 RDI 추정치를 각 정보원의 상대 값으로 변환하여, 이를 기반으로 문서의 간격 값을 제안하였다. 이는 해당 정보원이 생성한 문서의 우선 순위는 그대로 유지하면서 RDI의 값에 따라 다른 정보원에서 검색된 문서들과 비교되어 단일 우선 순위를 가지는 새로운 검색결과 집합을 생성한다. 이에 대한 자세한 알고리즘은 3.5절에서 소개한다.

3.2 제안된 메타검색 시스템의 기본설계 정책

일반적으로 메타검색에서 이질의 정보원으로부터 수집된 검색문서를 단일 검색결과 집합으로 통합하기 위해서는 다음의 두 가지 알고리즘에 대한 문제가 해결되어야 한다[2,3].

○첫 번째는 질의와 검색된 문서 사이의 유사도(similarity)를 추정할 수 있는 알고리즘의 개발

○두 번째는 검색된 문서들의 순위 매김 시 유사도에 따른 문서 가중치 부여 알고리즘의 개발

위의 두 가지 조건에 대해 본 논문에서는 첫 번째로 3.2절에서 알고리즘 1에 해당하는 유사도 추정 알고리즘을 제안하였으며, 이를 정보원에 대한 RDI 추정치라고 하였다. 그러나 본 논문에서는 제안된 RDI 추정치는 개별 문서에 대한 유사도를 추정하지 않는다. 단지 검색에 참여한 정보원을 추정하기 위해 모집단의 검색문서를 사용한다. 이를 통해 평가된 RDI 추정치를 문서의 우선 순위를 결정할 때 사용함으로써 검색문서에 대한 추정 값으로 대신하였다. 두 번째로 문서 가중치 부여 알고리즘에 대해서는 알고리즘 3에서 기술하였으며, 이를 문서 간격 값 평가 알고리즘이라고 하였다. 이와 같이 본 논문에서 제안하는 RDI 기반 메타검색 시스템은 아래 두 가지 조건을 만족하는 특징을 가진다.

○첫 번째, 만약 제안된 메타검색 시스템에 하나의 정보원만 검색에 참여하면, 서로 동일한 검색 결과를 생성한다.

○두 번째, 메타검색에 참여하여 개별 정보원들이 생성하는 검색문서의 우선 순위는 그대로 유지된다.

첫 번째 단일 정보원이 제공하는 검색 결과의 정보를 메타검색 시스템에서도 그대로 사용할 수 있도록 설계하였다. 이는 본 논문에서 제안하는 메타검색 시스템은 검색에 참여하는 정보원의 메타정보, 문서의 순위 매김 방법 등을 전혀 인식하지 못한다. 따라서 검색에 참여하는 정보원의 문서 수집 및 순위 매김 방법을 가능한 그대로 적용하는 방법을 선택하였다. 두 번째는 이질의 정보원으로부터 생성된 검색결과를 메타검색 시스템에서 통합할 때에는 첫 번째 조건을 만족하면서 문서의 우선 순위를 재평가하도록 설계하였다.

3.3 제안된 *RDI* 알고리즘

본 논문에서 제안한 HoleInOne 메타검색 시스템은

RDI를 기반으로 설계하였다. 여기에서 사용한 RDI는 메타검색 시스템에서 질의와 정보원사이의 유사도 추정 값을 말한다. 이는 사용자가 정보검색을 위해 질의할 때, 다양한 정보 제공자로부터 검색 결과를 수집하기 위해 제한된 정보원을 평가하는 기준으로 사용된다. 즉 인터넷의 다양한 정보 제공자중에서 사용자의 질의에 가장 적합한 정보 제공자는 어느 것인지를 판단하는 기준으로 사용된다. 다음의 그림 2는 RDI를 생성하는 과정이다.

본 논문에서 제안한 RDI 생성과정은 평가문서 수집, 문서의 재평가 및 관련성 검사, 관련문서의 위치정보 값, 정확도 값 등의 4단계로 해당정보원의 RDI 값을 평가한다. 다음 그림 2는 RDI 값을 생성하는 과정이다.

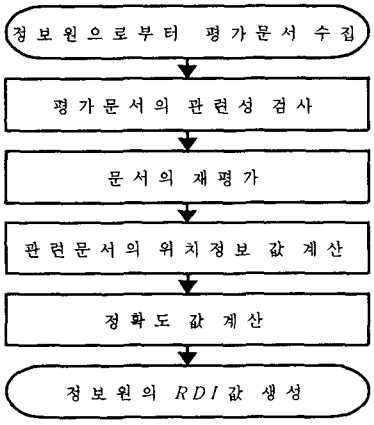


그림 2. *RDI* 생성과정

3.3.1 RDI 평가를 위한 메타데이터

메타검색 시스템에서 검색에 참여한 정보원으로부터 *tf, df, N* 등의 정보를 수집하기란 거의 불가능하다. 또한 대부분의 정보원은 서로 다른 문서 평가모델을 사용하고 있으며, 평가모델에 대한 상세한 메타정보는 전혀 공개하지 않는다, 혹 같은 평가모델을 사용하였다하더라도 서로 다른 정보원의 검색결과를 직접적으로 비교하기가 어렵다. 이는 서로 다른 정보원이 가지고 있는 문서의 집합이 동등하다고 할 수 없기 때문이다[6]. 이를 위해 본 논문에서는 검색문서 재평가를 위해 표 1과 같이 메타데이터를 정의하여 사용하였으며, 메타검색에 참여한 정보원으로부터 검색문서를 수집하여, 이들을 기반으로 해당 정보원의 *RDI*를 평가하였다.

표 1. *RDI* 평가 메타데이터

메타데이터	의미
DocumentRanking	· 검색문서의 순위
Re-ranking Scores	· 재평가된 문서의 값
TitleContent	· 검색문서의 Title 내용
AbstractContent	· 검색문서의 Abstract 내용
TitleKeywordCount	· Title내의 키워드 빈도 수
AbstractKeywordCount	· Abstract내의 키워드 빈도 수
DocumentRelCount	· 관련문서의 수
DocumentTCount	· 평가문서의 수

본 논문에서 정의한 메타데이터 TitleKeywordCount, *AbstractKeywordCount*는 *tf*로, *DocumentRelCount*는 *df*로, *DocumentTCount*는 *N* 문서평가 정보로 사용하였다.

3.3.2 검색문서 재평가 알고리즘

그림 1에서 첫 번째 단계로, 문서의 재평가 과정이다. 본 논문에서는 재평가된 문서 값을 $DocWeight_{ik}$ 메타데이터로 정의하였으며, 이는 정보원 k에서 i번째 문서의 재평가 값이라는 의미이다. 이를 위해 본 논문에서는 정보검색 분야에서 문서를 평가할 때 사용하는 모델과 본 논문에서 정의한 메타데이터를 응용하여 검색문서를 재평가하였다. 이때 평가된 값은 $0 \leq DocWeight_{ik} \leq 1$ 사이의 값을 가지며, 본 논문에서 사용한 검색문서 재평가 모델은 $tf_{ijk} \times \log(N/df_{jk})$를 사용하였다. 이때 *tf, df, N*의 값은 표1에서 정의한 메타데이터를 사용하여 평가하였으며, 이에 대한 검색문서에 대한 재평가 알고리즘은 다음과 같다.

알고리즘 1. 검색문서 재평가 알고리즘

```
1: DocumentValueEst(String Query, int N)
2:  While( N <= 0 ) Do
3:    findCollectionSearchDocument(String Query);
4:    getCollectionMetaData( int TitleKeywordCount,
       AbstractKeywordCount, DocumentRelCount, DocumentTCount);
5:  tf_ik = (TitleKeywordCount_ik + AbstractKeywordCount_ik);
6:    DocWeight_ik = tf_ik * log(DocumentRelCount_k / DocumentTCount_k)
8:    N--;
9:  EndWhile
10:  RETURN DocWeight_ik
11: End DocumentValueEst;
```

3.3.3 관련성 검사 알고리즘

이 절은 그림 1에서의 두 번째 단계이며 이는 평가

문서의 관련성 검사이다. 본 논문에서는 평가문서를 통해 생성된 관련성 정보를 $DocRel_{ik}$ 메타데이터로 정의하였으며, 이는 정보원 k에서 i번째 문서의 관련성 평가 정보 값을 의미한다. 본 논문에서는 관련 문서이면 1로, 그렇지 않으면 0으로 평가정보를 표현하였다. 이러한 $DocRel_{ik}$ 값들은 질의와 문서의 관련성 정도를 나타내는 값이며, 본 논문에서는 빈 URL, 중복문서, 첫 번째 단계의 검색문서 재평가 값 등의 3가지 정보를 기반으로 관련성 유무를 추정하였다. 이에 대한 처리 과정은 알고리즘 2와 같다[16].

알고리즘 2. 관련성 판단 알고리즘

```
1: DocRelCheck(String Query, String content, int N)
2:   Wile( N <= 0 ) Do
3:     If NotEmptyURL(content) Then
4:        NonRelevanceValue++;
5:        ElseIf DuplicateDocument(content) Then
6:        RelevanceValue++;
7:        ElseIf (DocumentWeight(content) ≥ α) Then
8:        RelevanceValue++;
9:        Else
10:       NonRelevanceValue++;
11:       EndIf
12:    N--;
13:  EndWhile
14:  RETURN RelevanceValue;
15: End DocRelCheck;
```

알고리즘 2의 검색문서의 관련성 검사과정은 표 1의 메타데이터를 기반으로 수행된다. 첫 번째 빈 URL은 정보원으로부터 검색된 문서 중 현재 사용할 수 없는 URL을 말하며, 두 번째 중복된 문서는 검색문서 중 동일한 URL을 가지고 있는 것으로 간주하여 처리하였다. 마지막으로 검색문서 평가 값은 첫 번째 단계에서 문서 재평가 값을 사용하였으며, 그 평가 값이 α 이하이면 관련이 없는 문서로 처리하였다.

3.3.4 관련문서의 위치정보 알고리즘

이 절은 그림 1에서 세 번째 단계로써, 검색문서 중에서 질의와 관련있다고 판단된 문서들의 순서정보를 이용하는 관련문서의 위치정보 평가 값이다. 이에 대한 평가정보를 본 논문에서는 $DocPos_{ik}$라 정의하였으며, 이는 정보원 k에서 i번째 문서의 위치정보 평가 값이라는 의미이다. 또한 본 논문에서는 검색문서가 관련문서이면 1로, 그렇지 않으면 0의 집합으로 위치정보를 표현하였다. 이러한 $DocPos_{ik}$ 평가 값은 관련 문서들이 나타나는 위치에 따라 차별적으로 정보원을 평가할 수 있는 기능을 제공해준다. 예를 들어, 서로 다른 두 정보원에서 평가 대상 10개 문서의 위치정보 집합이 다음과 같이 $DocPos_{k1}$, $DocPos_{k2}$ 이라고 가정하자.

$DocPos_{k1} :\rightarrow \{ 1, 1, 1, 1, 1, 0, 0, 0, 0, 0 \}$
$DocPos_{k2} :\rightarrow \{ 0, 0, 0, 0, 0, 1, 1, 1, 1, 1 \}$

이들 *k1*, *k2* 정보원의 $DocPos$ 값은 표면적으로 동등하게 5개의 문서가 관련된 문서로 평가되었다. 그러나 *k2*의 정보원과 *k1*의 정보원 사이에는 검색결과의 차이점이 있다. 그래서 일반적으로 *k2*보다는 *k1*이 더 좋은 검색결과를 생성했다고 할 수 있다. 이와 같은 점을 해결하기 위해 본 논문에서는 $DocPos$ 값을 평가할 수 있도록 다음과 같이 정의하였다.

$$DocPos_k = \frac{\sum_{i=1}^{n} \frac{DocRel_{ik}}{i}}{N} \quad (1)$$

위의 수식에 의해 값을 평가하면 $DocPos_{k1}$은 $\frac{(1+0.5+0.33+0.25+0.2)}{10} = 0.408$이고, $DocPos_{k2}$는 $\frac{(0.17+0.14+0.13+0.11+0.1)}{10} = 0.065$이다. 따라서 *k2*의 정보원보다는 *k1*의 정보원을 더 좋은 검색결과를 제공하는 것으로 평가할 수 있었다.

3.3.5 정확도평가 알고리즘

이 절은 그림 1에서 네 번째 단계로써, 검색문서의 정확도 값 평가이다. 이에 대한 평가정보를 본 논문에서는 $CoPre_k$로 정의하였으며, 이는 정보원 k에 대한 정확도 평가 값이라는 의미이다. $CoPre_k$ 평가 값은 검색된 문서 중에서 질의와 관련된 문서의 비율을 의미한다. 따라서 $CoPre_k$값이 크면 클수록 사용자의 질의에 좋은 검색결과를 제공한다할 수 있다. 본 논문에서는 이를 위해 다음과 같은 평가 식으로 $CoPre_k$값을 추정하였다.

$$CoPre_k = \frac{\sum_{i=1}^{n} DocRel_{ik}}{N} \quad (2)$$

마지막으로 해당 정보원의 RDI을 생성하는 과정이다. 이를 위해 본 논문에서는 RDI_k 메타데이터를 정의

하였으며, 추정된 평가 값은 주어진 질의와 정보원 k와의 관련성 정도를 표현하는 의미이다. 이러한 RDI_k 값의 평가는 첫 번째 단계에서부터 네 번째 단계까지 생성된 값들의 곱으로 해당정보원의 RDI을 추정하였으며, 이때 평가된 수식은 다음과 같다.

$$RDI_k = \left[\sum_{i=1}^{n} DocWeight_{ik} \times DocPos_k \times CoPre_k \right] \quad (3)$$

이때 추정된 RDI_k 값이 크면 클수록 질의에 대해 좋은 검색결과를 제공한다고 본 논문에서는 추정하였다.

3.4 *RDI* 기반 정보원 선택알고리즘

이 절에서는 3.3절에서 평가된 정보원의 RDI 값을 기반으로 정보원 선택 알고리즘을 소개한다. 이에 대한 처리과정은 그림 3과 같다.

먼저 3.3절로부터 평가된 각 정보원의 RDI 값을 입력으로 받는다. 이들은 질의된 검색어와의 유사도에 대한 추정 값으로 클수록 좋은 정보원이다라는 의미를 포함하고 있다. 본 논문에서는 개별적으로 평가된 정보원의 RDI 값을 직접적으로 메타검색에 사용하지 않고 검색에 참여한 모든 정보원을 대상으로 상대적인 값으로 변환한 후에 이를 오름차순으로 정렬하여 정보원을 선택하였다. 다음의 수식 4는 정보원의 RDI 평가 값을 상대적인 값으로 변환한다.

$$ColW_{Rate}k = \frac{RDI_k}{\sum_{k=1}^{m} RDI_{mk}} \quad (4)$$

수식 4에서 사용한 RDI_k 는 수식 3에서 평가된 정보원k에 대한 RDI 평가 값이다. 이 RDI_k 값을 사용하여 정보원에 대한 상대적 관련성 분포정보를 계산하였다. 즉 질의에 대하여 평가된 개별 정보원의 가중치 값에 검색에 참여한 모든 정보원에 대한 가중치 값을 합산한 값 $\sum_{k=1}^{m} RDI_{mk}$ 으로 나누어 개별 정보원이 차지하는 상대적인 비율을 구하였다. 이를 통해 질의에 대하여 해당 정보원이 차지하는 상대적인 비율을 계산하였다.

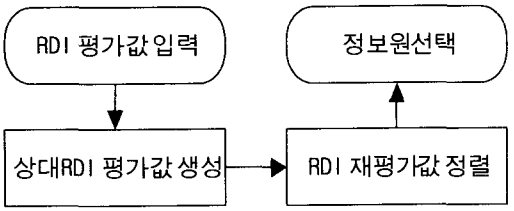


그림 3. 정보원선택 처리과정

이렇게 평가된 상대적인 $ColW_{Rate}k$ 추정치는 본 논문에서 정보원 선택, 문서수집의 크기 등을 판단하는 기준으로 응용하였다. 먼저, 첫 번째로 정보원 선택에 응용한 알고리즘이다. 이는 정보원에 대한 RDI 평가 값이 질의의 검색어와 정보원사이의 유사도를 나타내는 수치이다. 따라서 RDI의 값을 내림차순으로 정렬하여 가장 적합한 정보원을 판단하는 기준으로 사용하였다. 두 번째로는 정보원으로부터의 검색할 문서 집합의 크기를 결정하는 알고리즘에 응용하였다. 일반적으로 정보원에서는 검색 키워드가 발생되면 수만 혹은 수백 만개의 검색 문서를 사용자에게 검색 결과로써 회신한다. 그러나 대부분의 사용자는 검색결과로 회신된 문서 가운데 일부분만으로 자신이 찾고자하는 정보를 확인한다. 만약에 사용자가 선택한 일부의 문서 중에서 찾고자 하는 정보를 얻지 못하면 나머지 수많은 검색문서들을 무시하고 새로운 질의 검색어로 다시 검색을 시도한다. 따라서 본 논문에서는 3.2절에서 평가된 정보원에 대한 RDI를 기반으로 정보원으로부터 검색할 문서의 크기를 상대적인 RDI 값의 비율만큼 결정하였다. 또한 본 논문에서 제안한 HoleInOne 메타검색 시스템에서는 실험적으로 최종 검색결과 문서 집합의 크기를 50개로 설정하여 이를 실험하였다. 이때 각 정보원으로부터 검색될 문서집합의 크기는 다음의 수식 5로 결정된다.

$$ColSeDocNO_k = ColW_{Rate}k \times TotalDocNum \quad (5)$$

예를 들어 A, B, C 3개의 정보원이 검색에 참여하고, 이때 임의의 질의에 대하여 3.2절에 제시된 RDI 추정 값으로 각각 0.5, 0.3, 0.2를 얻었다고 가정한다. 이를 통해 본 논문에서 제안한 메타검색 시스템에서는 2가지 정보를 판단하게 된다. 첫 번째는 검색에 참여한 정보원들은 A, B, C 각각의 RDI 추정치에 대한 내림차순으로 A, B, C 순서로 질의에 대해 좋은 검색결과를 생성한다고 결정하게 된다. 두 번째는 검색에 참여한 정보원들에 대해 각각 A = (0.5×50), B = (0.3×50), C = (0.2×50) 개의 검색문서 크기 결정, 즉 25, 15, 10개의 문서를 차등적으로 정보원으로부터 수집하도록 검색문서의 크기를 판단한다. 또한 이질의 정보원으로부

터 검색된 문서들을 하나의 검색 결과 집합으로 통합할 때에는 관련성 분포정보의 상대적 비율을 기반으로 동일한 비율 내에 포함된 검색문서는 문서의 순위 매김 방법이 동등하다는 것으로 가정하였다.

3.5 RDI 기반 검색결과 통합알고리즘

이 절에서는 3.3절에서 추정한 정보원의 RDI를 기반으로 한 검색결과 통합 알고리즘을 소개한다. 이는 RDI를 기반으로 각 정보원으로부터 질의에 적합한 문서를 수집하고, 수집된 문서에 대하여 순위 매김을 수행하여 단일 검색 결과로 통합하여 사용자에게 검색결과로 회신한다. 본 논문에서 제안한 검색결과 통합 알고리즘은 각 정보원으로부터의 검색문서와 RDI 평가값을 입력받아, 단일 검색결과 집합을 출력한다. 이 알고리즘은 (1) 정보원의 문서간격 값 평가, (2) 문서의 순위 매김 및 (3) 검색 문서의 통합 정렬 등의 세 가지 주요 알고리즘으로 구성되어있다. 다음의 그림 4는 이에 대한 처리과정이다.

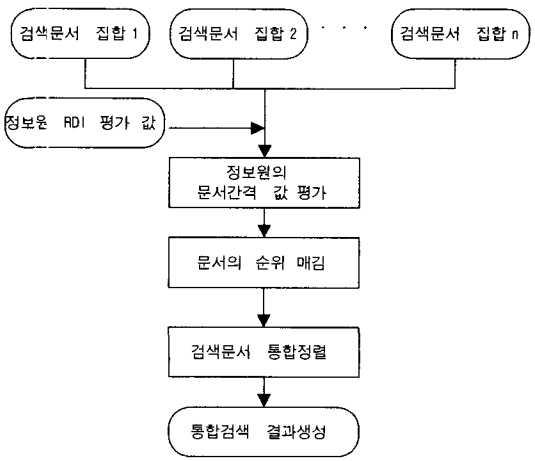


그림 4. 검색결과 통합 처리과정

3.5.1 문서 간격 값 평가 알고리즘

문서 간격 값 평가는 본 논문에서 제안한 검색문서의 순위 매김을 하는 방법이다. 일반적으로 문서의 순위 매김은 문서를 개별적으로 평가하고 이를 점수로 환산하여 내림차순 혹은 오름차순으로 정렬하는 것을 말한다. 본 논문에서는 이질의 정보원을 이용하는 메타검색 방법을 사용하였다. 이는 다음과 같은 몇 가지 단점을 가지고 있다. 첫째 개별 정보원의 문서 평가 방법을 알지 못한다. 둘째 개별 정보원을 상대적으로 비교할 수 없다. 따라서 본 논문에서 제안한 문서의 순위 매김 방법은 정보원으로부터 수집한 검색결과를 상대적으로 비교할 수 있으며, 개별정보원이 생성한 검색결과를 그대로 사용할 수 있도록 알고리즘을 설계하였다. 다음의 알고리즘 3은 문서 간격 값 평가 알고리즘이다.

알고리즘 3. 문서 간격 값 평가 알고리즘

```
1: IntervalValue(int ColWt[], int TotNum)
2:   Find(MinVal In ColWt);
3:   For All i ∈ TotNum Do
4:    InterWt[i]=MinVal/CoWt[i];
5:   EndFor
6:  RETURN InterWt[];
7: END IntervalValue
```

이는 문서검색에 참여한 정보원의 가중치 값을 상대적인 비율 값으로 변환하여 문서의 순위 매김에 사용하는 방법이다. 즉 정보원 중에서 가장 적은 가중치 값을 분자로 하고, 평가하고자하는 정보원의 가중치 값을 분모로 하여 나눈 값을 해당 정보원에서 검색된 문서의 간격 값으로 사용한다. 알고리즘 4에서 2번째 라인에서는 검색에 참여한 정보원 중에서 가장 적은 가중치를 가지는 정보원의 가중치 값을 찾는다. 그리고 3번째부터 5번째 라인에서 개별 정보원의 문서 간격 값을 계산한다.

3.5.2 문서순위 매김 알고리즘

본 논문에서 제안된 문서 순위 매김 알고리즘은 개별 정보원은 RDI의 평가 값에 따라 차등적으로 평가되며, 이들로부터 수집된 검색 문서는 양질의 검색문서가 나타날 확률만 다르다고 가정하여 문서 순위 매김을 수행하였다. 이때 양질의 검색문서가 나타날 확률은 정보원 평가 값인 RDI 추정 값을 응용하였다. 그러나 검색에 참여한 정보원들 사이에서 동일한 우선 순위를 가지는 검색 문서가 서로 비교되는 경우가 있다. 이 경우에는 RDI 추정 값이 클수록 양질의 검색 문서를 가질 확률이 높다라고 가정하였으므로, 이를 최종 순위 매김의 가중치로 사용하였다. 다음의 알고리즘 4는 문서순위 매김 알고리즘이다.

알고리즘 4에서 5번째 라인부터 7번째 라인까지가 검색된 모든 문서에 대하여 최종 순위 매김을 수행하

알고리즘 4. 문서순위 매김 알고리즘

```
1: DocRankValue(int ColWt[], int TotNum)
2: Set 100 To DocMaxScore;
3: For All i ∈ TotNum Do
4:  Call IntervalValue(int ColWt[], int TotNum)
5:  DocNum=InterWt*100;
6:  FOR j = 0 to (DocNum - 1) DO
7:   DocRank[j,i]=DWij+(DocMaxScore-(InterWt[i]*j));
8:  EndFor
9: EndFor
10: RETURN DocRank[][];
11: End DocRankValue;
```

고 있다. 이들은 해당 정보원에 대한 가중치 값과 문서에 대한 단계별 가중치를 더하여 최종 순위 매김 값으로 사용한다. 마지막으로 관련성 분포 정보를 기반으로 평가된 문서의 평가 값에 따라 내림차순으로 단일 검색 결과의 집합으로 생성되며, 사용자에게 질의에 대한 결과로써 제공한다. 예를 들어, 검색에 참여한 정보원 A, B, C의 RDI 평가 값이 0.2, 0.3, 0.5이고, 전체 검색문서의 집합을 50이라고 가정하자. 이에 따라 3.4절에서 제안된 정보원 선택알고리즘에 의해 각 정보원 A, B, C로부터 검색할 문서집합의 크기를 결정한다. 이때 각 정보원으로부터 검색할 문서집합의 크기는 A: 10 = (0.2×50), B: 15 = (0.3×50), C: 25 = (0.5×50)로 결정된다. 두 번째로 3.5절에서 제안한 검색결과 통합 알고리즘에 의해 문서 순위 매김에 사용될 문서 간격 값을 평가한다. 이에 대한 문서 간격 값은 알고리즘 3에 의해 평가되며, A:1=(0.2÷0.2), B:0.67=(0.2÷0.3), C:0.4=(0.2÷0.5)가 된다. 따라서 이를 기반으로 정보원 C에 대한 25개의 검색문서 순위 매김 값은 다음과 같다.

$$Doc_{value}C_1: 50.5=(0.5+(50-(0.4\times0)) \quad (6)$$

위의 계산과정에서 $Doc_{value}C_1$는 정보원 C의 첫 번째 문서를 의미하며, 50.5는 최종 평가된 첫 번째 검색문서의 순위 매김 값이다. 다음으로 0.5는 정보원C의 간격 값으로 이는 다른 정보원과 동일한 위치에 존재하는 검색문서를 비교할 때 우선 순위를 가질 수 있도록 하는 의미를 포함하고 있다. 다음의 50은 메타검색에서 최종 전체 검색집합의 크기를 의미하며, 다음의 0.4는 정보원 C의 문서 간격 값이다. 마지막으로 0은 현재 평가하고 있는 문서의 순서를 의미한다. 이는 두 번째 문서인 경우에는 1로 되며, 검색문서의 순서에 따라 1씩 증가한다. 따라서 정보원 C의 나머지 문서에 대한 순위 매김은 다음과 같이 계산된다.

$$Doc_{value}C_1: 50.5=(0.5+(50-(0.4\times0))$$

$$Doc_{value}C_2: 50.1=(0.5+(50-(0.4\times1)$$

$$\cdots$$

$$Doc_{value}C_{25}: 40.9=(0.5+(50-(0.4\times24)$$

즉 정보원 C의 문서 순서 간격 값은 0.4씩 감소하면서 문서에 대한 순위 매김 정보를 생성하게된다. 이는 정보원 C가 검색에 참여한 다른 정보원보다 검색문서의 순위 매김 정보가 분포도가 높다는 것을 의미하며, 이를 통해 질의와 관련성이 높다는 의미를 구현하였다.

4. HoleInOne 메타검색 시스템의 구현

이 장에서는 본 논문에서 제안한 관련성 분포 정보(RDI) 기반 HoleInOne 메타검색 시스템에 대해 살펴본다. 먼저 4.1절에서는 HoleInOne 통합 검색 시스템의 전체적인 개요에 대해 살펴보며, 4.2절에서는 HoleInOne 메타검색 시스템에서 RDI 평가 처리과정을 살펴본다. 이들 평가 요소는 질의어와 정보원 사이의 관련 분포 정보를 추출하는데 사용되며, 또한 질의에 대하여 가장 적합한 정보원의 선택과 검색 문서의 순위 매김 및 통합 등에 사용된다. 다음으로 4.3절에서는 질의에 대해 가장 적합한 정보원을 선택하는 정보원 선택 부 시스템에 대하여 살펴보고, 4.4절에서는 이 질의 정보원으로부터 검색된 문서를 단일 검색 결과로 생성하는 컬렉션 융합 부 시스템에 대해 살펴본다.

4.1 HoleInOne 시스템의 개괄구조

이 절에서는 본 논문에서 제안된 컬렉션 융합 모델을 평가하기 위해 구현한 HoleInOne 통합 검색 시스템의 전체적인 구조에 대해 살펴본다. 본 논문에서 구현한 HoleInOne 통합 검색 시스템은 사용자 인터페이스 부 시스템, 질의어 처리 부 시스템, 컬렉션 평가 부 시스템, 컬렉션 융합 부 시스템 등 4 개의 부 시스템으로 구성되어 있다. 그림 5는 실험을 위해 구현한 HoleInOne 통합 검색 시스템의 전체 구조이다.

사용자 인터페이스 부시스템은 검색에 참여한 이질의 다양한 정보원들을 하나의 모습으로 보여주며, 사용자들은 질의 및 검색 결과 등에 관한 정보의 입력

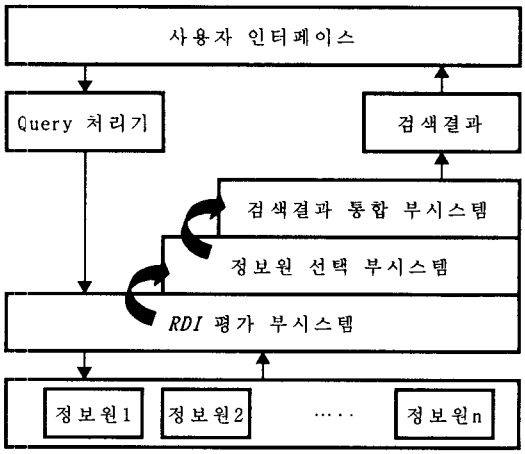


그림 5. HoleInOne 시스템의 개괄 구조

및 수집을 할 수 있다. 이를 위해 본 논문에서는 HTML, 자바 서블릿 등을 사용하여 구현하였다. RDI 평가 부시스템은 사용자로부터 질의어가 주어질 때 가장 양질의 문서를 가지고 있는 정보원을 선택하는 기능을 제공한다. 본 논문에서의 정보원 평가는 검색 문서의 관련성 평가와 정보원 평가 등으로 검색에 참여한 정보원을 평가한다. 검색결과 통합 부시스템은 이 질의 정보원으로부터 검색된 문서들을 하나의 검색 결과 집합으로 통합하여 사용자에게 검색 결과로써 되돌려주는 기능을 제공한다. 이는 검색된 문서의 순위 매김 및 통합 과정으로 검색 결과를 생성한다.

4.2 구현된 클래스와 메소드

이 절에서는 본 논문에서 구현한 HoleInOne 메타검색 시스템을 위해 설계된 클래스와 메소드의 계층적 구조와 그 기능에 대해 소개한다. 본 논문에서 메타검색 시스템을 구현하면서 정의된 클래스와 메소드의 상관관계는 그림 6과 같다.

본 논문에서 HoleInOne 메타검색 시스템을 구현하기 위해 3개의 클래스와 10개의 메소드를 정의하였다. 이에 대한 자세한 기능은 4.2.1절과 4.2.2절에서 소개한다. 그림 6에서 클래스는 사각형으로, 메소드는 둥근 사각형으로 표시하였다. 이들의 계층적 관계는 클래스 혹은 메소드에서 호출되어지는 관계를 그림으로 표현하였다.

4.2.1 구현한 클래스

이 절에서는 메타검색 시스템을 구현하면서 정의한

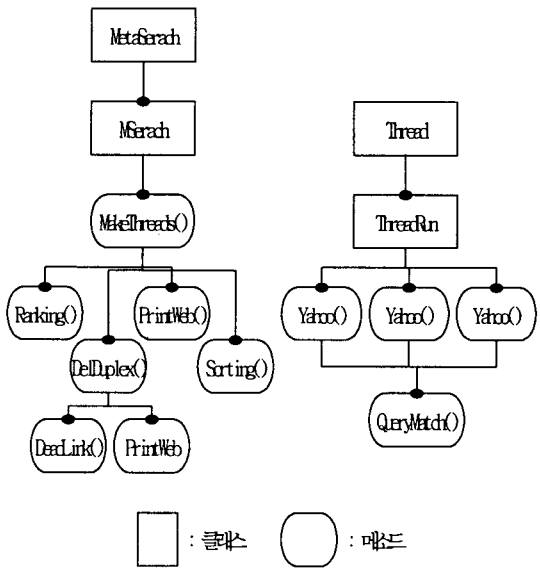


그림 6. HoleInOne 시스템의 클래스와 메소드

클래스와 그 기능에 대해 소개한다. 이에 대한 클래스들은 다음과 같다.

■ public class MetaSearch extends HttpServlet

HTML FORM 태그의 action 속성과 연결될 명칭이며, 본 논문에서 구현한 메타검색 시스템의 최상위 클래스이다.

■ class MSearch

자바 서블릿의 main() 메소드를 포함하고 있는 클래스이며, 메타검색 시스템의 직접적인 동작에 대한 메소드들을 가지고 있다. 이에 대한 메소드로 MakeThreads(), Ranking(), DelDuplex(), Sotring(), PrintWeb(), DeadLink() 등이 있다.

■ class ThreadRun extends Thread

MakeThreads() 메소드에서 쓰레드 객체를 생성할 때 객체에 대한 타입으로 정의하였다. 이들 타입은 메타검색에 참여하는 정보원에 대한 검색호출과 검색된 문서를 3.3.1절에 정의된 메타데이터에 대해 정보를 분석할 수 있게 정의되어 있다. 본 논문의 메타검색 시스템은 이를 기반으로 RDI를 평가하고 메타검색을 수행한다.

4.2.2 구현한 메소드

이 절에서는 메타검색 시스템을 구현하면서 정의한 메소드와 그 기능에 대해 소개한다. 이에 대한 메소드들은 다음과 같다.

■ public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException

HTML FORM에서 method 속성에서 "get"방식으로 입력 데이터를 전송할 때 서블릿 실행 프로그램에서 전송된 데이터를 읽기 위한 메소드이다.

■ public void MakeThreads()

메타검색에 참여하는 이질의 정보원을 쓰레드로 생성하여 정보원을 동작하는 메소드이다. 본 논문에서는 3개의 yahoo, altavista, excite 검색엔진을 대상으로 구현하였다.

■ public void Ranking()

메타검색에 참여하는 이질의 정보원으로부터 검색된 문서에 대해 본 논문에서 제안한 평가 방법에 따라 문서를 재평가하여 문서에 대한 순위 매김을 수행하는 메소드이다.

■ public int DelDuplex()

메타검색에 참여하는 이질의 정보원으로부터 검색된 문서에 대해 본 논문에서 제안한 평가 방법에 따라 문서를 재평가하여 문서에 대한 순위 매김을 수행하는 메소드이다.

■ public void Sorting(int count)

이질의 정보원으로부터 검색된 문서들은 Ranking() 메소드에 의해 순위 매김된다. 이를 단일 검색 결과로 생성하기 위해 내림차순으로 검색문서 집합을 정렬하는 메소드이다.

■ public void PrintWeb(int TotalCount)

Sorting() 메소드에 의해 정렬된 검색문서를 사용자에게 최종회신하는 메소드이다. 입력 매개변수는 정렬된 문서의 전체 수이며, 이를 기반으로 사용자의 웹 브라우저에 검색결과를 회신한다.

■ public boolean DeadLink(String target)

이질의 정보원으로부터 검색된 문서들에 대해 현재 사용할 수 있는지의 유·무를 평가하는 메소드이다. 입력 매개변수는 검색문서의 URL이며, 리턴 값은 불리언 값으로 1이면 사용 가능한 URL이고, 0이면 현재 사용할 수 없는 URL이란 의미를 가진다.

■ public void makeURLnStream(String query)

메타검색에 사용되는 정보원에 대해 동시에 질의를 발생하도록 하는 메소드이다. 입력매개변수는 사용자가 질의한 검색어 값을 가지며, 리턴 값은 각 정보원에 대한 메소드 호출을 가진다.

■ public void Yahoo()

사용자가 입력한 검색어를 yahoo 정보원에게 질의하고, 검색문서 목록과 3.3.1절에 정의한 메타데이터에 대한 메타정보를 분석하는 메소드이다.

■ public void Alta()

사용자가 입력한 검색어를 altavista 정보원에게 질의하고, 검색문서 목록과 3.3.1절에 정의한 메타데이터에 대한 메타정보를 분석하는 메소드이다.

■ public void Excite()

사용자가 입력한 검색어를 excite 정보원에게 질의하고, 검색문서 목록과 3.3.1절에 정의한 메타데이터에 대한 메타정보를 분석하는 메소드이다.

■ public int QueryMatch(String query, String content)

사용자가 입력한 질의에 대해 제목이나 내용에서 매치되는 개수를 카운터하는 메소드이다.

4.3 HoleInOne 시스템의 구현

이 절에서는 HoleInOne 메타검색 시스템의 구현 화면에 대해서 살펴본다. 먼저 이 절의 구성은 HoleInOne 메타검색 시스템의 질의 화면과 HoleInOne 메타검색 시스템의 검색 결과 화면에 대한 내용이다. 다음의 그림 7은 HoleInOne 메타검색 시스템의 질의 화면이다.

그림 7은 본 논문에서 제안한 RDI 기반 메타검색 시스템 모델의 성능을 분석하기 위해 최소의 기능으로

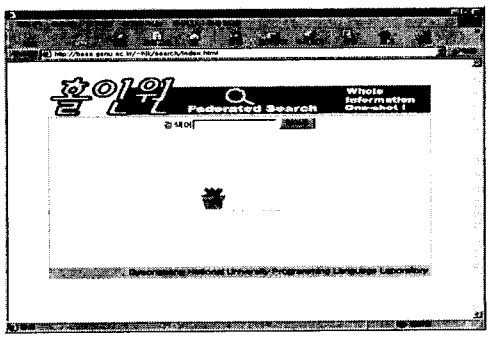


그림 7. 질의 화면

구현한 질의 화면이다. 질의 화면에 입력된 검색어는 HoleInOne 시스템의 검색에 참여한 이질의 정보원에서 사용하는 질의 문법으로 자동 번역되어 HoleInOne 시스템에서 대신 질의를 한다. 이는 사용자들에 다양한 정보원들의 모습을 숨기고 하나의 인터페이스만을 제공함으로써 인터넷상에 존재하는 다양한 정보원의 질의 문법 이해에 대한 사용자의 부담을 해결해 준다.

다음의 그림 8은 그림 7로부터 입력된 검색 질의에 대해 HoleInOne 시스템에서 검색 결과를 생성해주는 검색 결과 화면이다.

그림 8은 HoleInOne 시스템에 참여한 여러 개의 이질 정보원으로부터 검색 결과를 수집한 후에 본 논문에서 제안하는 메타검색 방법으로 단일 검색 결과를 생성한 화면이다.

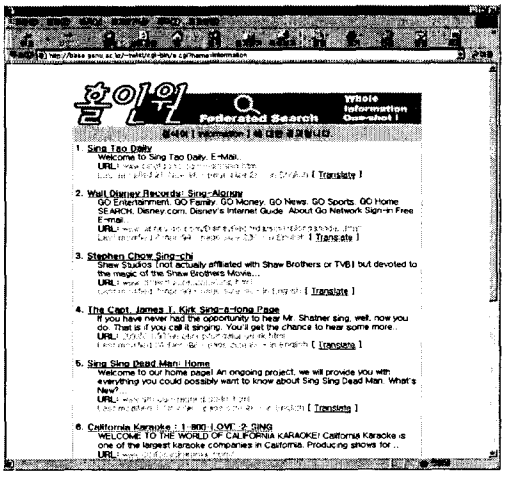


그림 8. 검색 결과 화면

5. 실험 결과

이 절에서는 본 논문에서 제안한 HoleInOne 메타검색 시스템을 평가하고, 그 실험결과를 소개한다. 먼저 제안된 메타검색 시스템을 평가하는 방법으로 일반검색 시스템 InforSeek[20], Excite[22], Altavista[21], Yahoo[19]와 메타검색 시스템인 SavvySearch[18], ProFusion[6,17] 등을 사용하여 상대적으로 검색결과를 분석하였다. 이때 검색결과를 분석할 때 본 논문에서는 정확도를 검색성능 평가의 측정 기준으로 사용하였다. 정확도란 검색된 문서 중에서 유일하게 검색 키워드와 관련된 문서의 수를 의미한다. 따라서 정확도가 높으면 높을수록 사용자에게 양질의 검색결과를 제공한다고 가정할 수 있다.

먼저 검색 정확도를 평가하기 위해 사용된 검색 키워드는 ProFusion[6,17] 메타검색 시스템에서 사용한 질의를 본 논문에서도 검색 키워드로 사용하였다. 이에 대한 검색 키워드의 순서는 다음과 같다. (1) Science and engineering, (2) Computer Science, (3) Travel, (4) Medical and Biotechnology, (5) Business and Finance, (6) Social and Religion, (7) Society Law and Government, (8) Animals and Environment, (9) History, (10) Recreation and Entertainment, (11) Art, (12) Music, (13) Food 등이다. 이때 키워드에 붙여진 번호는 다음의 실험결과 테이블에서 나타나는 번호를 의미한다. 다음으로는 검색 문서에 대한 정확도로 검색 성능을 평가하였다. 본 논문에서는 검색된 문서 중에서 빈URL(DOC_{empty}), 중복문서(DOC_{dup}), 관련없는 문서(DOC_{irrel})등의 검색문서를 제외한 것을 정확도라 정의하였다. 이에 대한 정확도 평가 수식은 다음과 같다.

$$DocRe_{precision} = (TotalSearchDoc - (DOC_{empty} + DOC_{dup} + DOC_{irrel})) \quad (6)$$

따라서 본 논문에서는 수식 6에서 제시한 것과 같이 검색의 정확도가 높으면 해당 정보원이 질의에 대해 매우 양질의 가지고 있다고 가정하였다. 본 논문에서 평가하는 검색 문서의 정확도는 평가 문서 집합 내에서 유일하게 관련된 문서 수의 비율이다.

본 논문에서는 수식 6과 같이 검색문서의 정확도에 대하여 실험하고 이를 분석하였으며, 이에 대한 실험결과를 상위 30개 검색문서에 대한 분석자료를 테이블로 표현하였다. 표 2는 상위 10개 검색문서에 대한 정확도이다. 표 2에서 구분항목에 표시된 1부터 13까지의 의미는 검색 키워드이며, 상단에 기술된 것은 비교대상의 정보원 명칭이다. 표 2를 통해 본 논문에서 제안한 메타검색 시스템은 전체 13개의 검색 키워드 중에서 7개의 키워드에 대해 높은 정확도를 나타내었다.

6. 결 론

본 논문에서 제안한 RDI 기반 메타검색 시스템은 질의와 정보원사이의 유사도에 따른 새로운 메타검색 시스템 모델이다. 이는 메타검색에 참여하는 정보원에 대한 평가를 실시간으로 추정하는 방법으로써, 정보원

표 2. 상위 30개 검색문서에 대한 정확도

구분	InfoSeek	Excite	AltaVista	Yahoo	SavvySearch	ProFusion	HoleInOne
1	0.97	0.63	0.40	0.87	0.50	0.37	0.87
2	0.90	0.63	0.40	0.67	0.47	0.50	0.83
3	0.87	0.73	0.63	0.60	0.47	0.57	0.80
4	1.00	0.80	0.50	0.83	0.70	0.73	0.87
5	0.67	0.67	0.50	0.63	0.47	0.60	0.80
6	0.70	0.60	0.60	0.50	0.40	0.47	0.80
7	0.33	0.43	0.70	0.47	0.23	0.47	0.90
8	0.50	0.43	0.57	0.40	0.20	0.73	0.63
9	0.73	0.60	0.67	0.37	0.50	0.50	0.77
10	0.80	0.73	0.70	0.77	0.60	0.67	0.80
11	0.70	0.97	0.67	0.67	0.67	0.57	0.80
12	0.87	0.90	0.90	0.40	0.70	0.70	0.97
13	0.90	0.73	0.80	0.37	0.63	0.57	0.90

에 대한 대용량의 데이터베이스를 생성하지 않아도 된다는 장점을 가진다. 또한 인터넷과 같이 동적인 환경에 적용할 때 적은 비용으로 시스템을 구축할 수 있는 장점을 가진다. 이와는 대조적으로 제안된 메타검색 시스템은 정보원 평가를 사용자가 검색을 요청하는 시기와 동일하게 수행하여 정보 검색 시 정보원 평가에 대한 추가 오버헤드를 발생시킨다. 또한 제안된 메타검색 시스템 모델을 실제 응용하기 위해서는 다음과 같은 문제를 보완해야 한다. 첫 번째는 질의어 자동변환에 대한 연구이다. 이는 메타검색 시스템이 정보원에 대한 질의 문법으로 자동변환 시켜주는 알고리즘이다. 본 논문에서는 정보원에 대한 질의 문법을 제한적으로 축소하여 사용하였다. 두 번째는 검색 문서의 관련성 판단 알고리즘에 대한 연구이다. 본 논문에서는 3.3.1절에서 정의한 메타데이터를 통해 검색문서를 재평가하였다. 그러나 검색문서를 평가하기 위해서는 보다 더 체계적인 메타데이터 혹은 알고리즘 개발이 필요하다.

앞으로의 연구 과제는 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 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 International ACM SIGIR 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, National Institue of Standards and Technology, Special Publication pp. 216-225., 1994.

[3] E. M. Voorhees, N. Gupta, and B. Johnson-Laird., "Learning Collection Fusion Strategies," *ACM SIGIR '95*, pp. 172-179, 1995.

[4] C. L. Viles and J. C. French, "Dissemination of Collection Wide Information in a Distributed Information Retrieval System," *ACM SIGIR '95*, 1995.

[5] A. Moffat and J. Zobel, "Information Retrieval Systems for Large Document Collections," *The Third Text REtrieval Conference (TREC-3)*, pp. 85-94., 1994.

[6] 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.

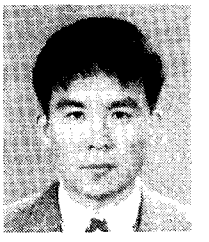
[7] C. Baumgarten, "Probablistic Information Retrieval in a Distributed Heterogeneous Environment," *PhD Thesis,* Dresden Univ. of Techn., Accepted, 1999.

[8] C. Baumgarten, "A Probabilistic Solution to the Selection and Fusion Problem in Distributed Information Retrieval," *ACM SIGIR '99*, 1999.

[9] J. C. French, A. L. Powell, J. Callan, C. L. Viles, T. Emmitt, K. J. Prey and Y. Mou, "Comparing the Performance of Database Selection Algorithms," *ACM SIGIR 99*, 1999.

[10] N. Fuhr, "Resource Discovery in Distributed Digital Libraries," *ACM SIGIR '99*, 1994.

[11] 금기문, 남세진, 신동욱, 김태균, "문서 클러스터링 정보를 이용한 컬렉션 융합," *한국정보과학회 추계학술 논문발표집*, pp. 147-149., 1998.
[12] 김현주, 김상준, 배종민, "관련성 분포 정보를 이용한 컬렉션 융합," *한국정보처리학회 춘계학술 논문발표집*, pp. 907-910., 1999.
[13] 김현주, 김상준, 배종민 "디지털 도서관에서 사용자 질의어와 컬렉션 사이의 관련성 분포정보를 이용한 컬렉션 융합," *한국 정보처리학회 논문지 제6권 제10호*, pp. 2728-2739., 1999.
[14] 김연곤, 엄채임, 변정용, "빈 연결을 제거하는 메타 검색 엔진의 구현," *한국멀티미디어학회 추계학술발표회*, pp. 359-364., 1998.
[15] 김현주, 배종민, "통합 검색에서 관련성 분포 정보를 이용한 정보원 선택에 관한 연구," *경상대학교 전산개발연구소 제14권*, 1999.
[16] 김현주, 배종민, "메타 검색에서 질의와 컬렉션 사이의 관련성 분포정보를 이용한 컬렉션 선택," 한국멀티미디어학회 논문지 제4권 4호, pp.287-296., 2001.
[17] ProFusion, (http://www.profusion.com/).
[18] SavvySearch, (http://www.savvysearch.com/).
[19] Yahoo, (http://www.yahoo.com/).
[20] InfoSeek, (http://www.infoSeek.com/).
[21] AltaVista, (http://www.altavista.com/).
[22] Excite, (http://www.excite.com/).



김 현 주

1988년 경상대학교 전산통계학과(이학사)
1990년 숭실대학교 전자계산학과(공학석사)
2000년 경상대학교 전자계산학과(공학박사)
1994년~1997년 제일정밀공업(주) 연구원
2000년~2002년 경남정보대학 컴퓨터정보계열 전임강사
2002년~현재 진주산업대학교 컴퓨터공학과 전임강사
관심분야 : 정보검색, 디지털 도서관, 웹 프로그래밍, XML
E-mail: khj@jinju.ac.kr

배 종 민

1980년 서울대학교 사범대학 수학과(이학사)
1983년 서울대학교 계산통계학과(이학석사)
1995년 서울대학교 계산통계학과(이학박사)
1982년~1984년 한국전자통신연구원 연구원
1984년~현재 경상대학교 컴퓨터과학과 교수
관심분야 : 병렬 프로그래밍 언어, 디지털 도서관, 정보검색
E-mail: jmbae@nongae.gsnu.ac.kr

교 신 저 자

김 현 주 660-758 경남 진주시 칠암동 150번지 진주산업대학교 컴퓨터공학부