JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 19, No. 4, pp. 719-729, Apr. 2018

# 웹 검색엔진 분류 및 하이브리드 검색엔진의 필요성

**백 주 련**
평택대학교 데이터정보학과

# Classification of Web Search Engines and Necessity of a Hybrid Search Engine

**Juryon Paik**
Department of Digital Information & Statistics, Pyeongtaek University, Pyeongtaek-si, Gyeonggi-do, 17869, Korea

**[요 약]**

2017년 기준, 데스크탑과 모바일 영역에서 90% 이상의 압도적인 점유율을 보이는 검색엔진은 Google로써, 대다수의 사람들은 Google 이 검색하는 영역이 웹의 전체라고 생각할 것이다. 그러나 웹 연구 결과에 의하면 전체 웹 데이터의 불과 10% 만이 Google 에 의해 검색가능하다고 한다. 대부분의 영역은 딥 웹이라고 불리며 Google 과는 다른 종류의 검색엔진들에 의해 검색된다. 해당 엔진들은 자신만의 딥 웹 데이터베이스를 구축 후 특화된 알고리즘을 사용하여 높은 정확성과 전문성의 검색결과를 제공한다. 현재 사용되고 있는 검색엔진들 중, 전체 웹 영역을 검색하는 엔진은 존재하지 않는다. 광범위에 걸쳐 그리고 유효하면서 정확·신속한 검색을 수행하기 위한 최적의 방법은 Google 같은 일반적인 검색엔진과 딥 웹 검색엔진들을 동시에 적용하여 결과를 도출하는 것이다. 본 논문에서는 이러한 검색엔진을 하이브리드검색엔진이라 명하고 기존 검색엔진들에 비해 갖는 차이점 및 특징에 대해 살펴본 후 개괄적인 프레임을 제시한다.

**[Abstract]**

Abstract In 2017, it has been reported that Google had more than 90% of the market share in search-engines of desktops and mobiles. Most people may consider that Google surely searches the entire web area. However, according to many researches for web data, Google only searches less than 10%, surprisingly. The most region is called the Deep Web, and it is indexable by special search engines, which are different from Google because they focus on a specific segment of interest. Those engines build their own deep-web databases and run particular algorithms to provide accurate and professional search results. There is no search engine that indexes the entire Web, currently. The best way is to use several search engines together for broad and efficient searches as best as possible. This paper defines that kind of search engine as Hybrid Search Engine and provides characteristics and differences compared to conventional search engines, along with a frame of hybrid search engine.

**색인어 :** 검색엔진, 딥웹, 범용검색엔진, 수직검색엔진, 하이브리드검색엔진

**Key word :** Search engine, Deep web, General search engine, vertical search engine, Hybrid search engine

**http://dx.doi.org/10.9728/dcs.2018.19.4.719**

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Received** 29 March 2018; **Revised** 08 April 2018
**Accepted** 25 April 2018

***Corresponding Author; Juryon Paik**

**Tel:** +82-31-659-8354
**E-mail:** jrpaik@ptu.ac.kr

## Ⅰ. 서 론

기하급수적인 웹 데이터 증가의 주요 원인 중 하나는, 정보의 제공 단위가 과거 기관 중심으로 이뤄졌다면, 현재는 개인 중심으로 이루어지고 있는 점이다. 2014 년 9 월을 기점으로 웹 사이트들의 수가 10 억 개를 넘었다는 연구 결과가 Web Server Survey에 발표되었으며 이는 NetCraft에 의해 공인된 수치이다 [1]. 그 보다 먼저 2006 년 5 월 작성된 Digimind 기술 보고서에 의하면 쉽게 접근 가능한 웹 페이지들은 700 억 개 이상 산재해 있으며 매년 그 양은 두 배 정도씩 증가한다고 기술하였다 [2]. 2016 년 구글 Inside Search는 웹에는 60 조 개 이상의 웹 페이지들이 존재하며 계속해서 증가하고 있다고 계측했지만 그 중 3000 분의 1 정도만 구글에서 색인 가능하다고 추산한다. 현존하는 최고의 검색엔진이라고 여겨지는 구글조차도 검색할 수 없을 정도의 웹 페이지들, 일명 **딥 웹 (Deep Web)** 이라 칭하는 세상이 존재한다 [3,4,5,6].

그림 1[1])은 현재 웹에 존재하는 데이터들의 분포 비중과 영역에 대한 명칭을 보인다. 그림에서 나타나듯이, 우리가 일상적으로 사용하는 구글, 야후 등의 검색 엔진 뿐만 아니라 페이스북이나 트위터 등의 대부분의 사회관계망 서비스 또한 **표면 웹 (Surface Web)** 이라 명명하는 영역에 위치해 있는 것을 알 수 있다. 표면 웹은 구글이나 야후 같은 검색 엔진들이 색인할 수 있는 문서와 데이터들로 이루어져있지만 해당 정보들이 웹에서 차지하는 비중은 10% 에도 미치지 못한다 [2,7]. 2016 년 3 월, www.worldwidewebsize.com 에서는 46 억 6 천만 개 정도의 검색 또는 색인 가능한 웹 페이지들이 산재한다고 발표했지만 해당 수치는 딥 웹에 존재하는 자료들은 전혀 고려하지 않은 수치로 이는 딥 웹에 있는 대다수의 문서나 데이터들은 기존 검색 엔진으로는 찾을 수 없기 때문이다.

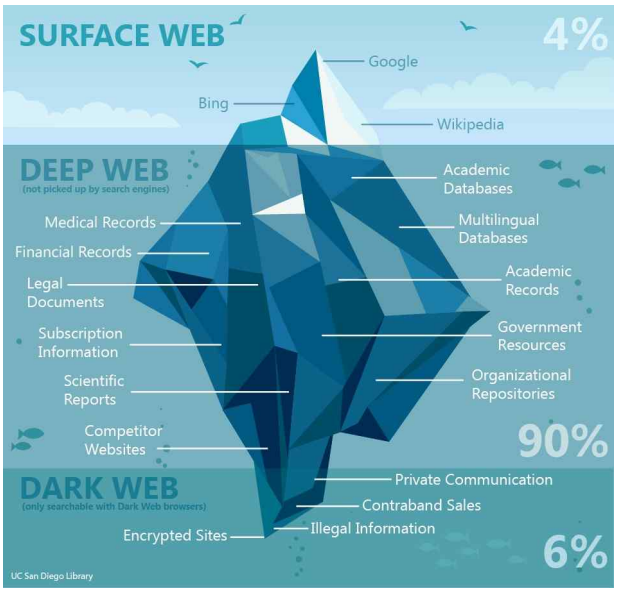


**그림 1.** 표면 웹 vs. 딥 웹
**Fig. 1.** Surface Web vs. Deep Web

전체 웹 정보의 90% 이상 (일반적으로, 96% 정도의 비중을 차지하고 있는 것으로 고려)을 차지하고 있는 딥 웹의 데이터들[2])은, 표면 웹에 존재하는 데이터와 비교해서 훨씬 더 전문적이고 정확하게 수준 높은 지식을 전달하는 도메인 특화된 데이터라고 할 수 있다. 그림 1은 딥 웹을 구성하고 있는 전문 영역들에 대한 개괄적인 분류 또한 보인다. 학계, 언어, 의학, 컴퓨터, 과학, 정부 등 다양한 분야의 전문 자료들이 딥 웹을 구성하고 있음을 알 수 있다. 전문가들은 딥 웹에 존재하는 정보의 크기는 표면 웹의 정보들보다 500 배 이상 많은 것으로 보고 있다. 뿐만 아니라, 딥 웹은 전체 웹 영역에서 가장 빠르게 데이터가 증가하고 있는 영역이다. 문제는 우리가 손쉽게 이용하는 구글, 야후, Bing 등과 같은 검색 엔진으로는 접근이 허용되지 않는다는 것이다. 왜냐하면 딥 웹을 구성하고 있는 대다수의 데이터원들은 전문적인 데이터베이스로 구축되어 있기에 허가받지 못한 사용자들은 접근 자체를 할 수 없을 뿐 더러 해당 데이터들은 일반 검색엔진들에게 색인을 허용하지 않기 때문이다. 끊임없이 변하고 있는 표면 웹 데이터와 더불어 그 양을 측정할 수 없는 딥 웹의 데이터까지, 사용자의 요구에 부합하는 정보를 얼마나 정확하고 빠르게 그리고 가능한 많이 웹 세상으로부터 도출하는가는 데이터 주도의 빅데이터 세상에서 검색 엔진들이 필수적으로 갖추어야 하는 능력일 것이다.

본 논문은 전체 웹의 90% 이상을 차지하고 있는 딥 웹과 그와 관련된 문제점들을 기술한 후 검색 결과의 질을 높이기 위한 표면 웹과 딥 웹의 결합에 대한 프레임 워크를 소개한다.

## Ⅱ. 검 색 엔 진

### 2-1 검색엔진의 한계

90 년 대 Tim Berners-Lee 에 의해 개발된 웹은 그 편의성과 대중성으로 자유롭게 발전했으며 생활에서 차지하는 비중은 급속도로 빠르게 증가해왔다. 2000 년대 들어서면서 스마트 기기의 기술적인 발달과 보급력, 그리고 스마트폰을 중심으로 하는 페이스북, 트위터 등의 사회 관계망 서비스 (Social Network Services)의 폭발적인 성장과 맞물려 정보와 콘텐츠 양은 그 어느 때보다 기하급수적으로 팽창하고 있다 [8,9]. 이와 같은 데이터의 바다, 정보의 바다로 들어가는 입구가 검색엔진이며 검색엔진은 빅데이터 세상을 보는 주요한 기술 중 하나로, 웹 세계의 눈 역할을 한다고 할 수 있다.

검색 엔진[10]이란 데이터 셋을 사용해서 검색하는 프로그램을 지칭한다. 로봇이라 칭하는 특별한 프로그램을 웹에 있는 데이터들을 자동으로 수집 후 색인하여 검색 폼을 제공하여 원

1) https://ucsd.libguides.com/c.php?g=90839&p=4299442

2) 본 논문은 Dark Web은 고려하지 않는다.

하는 정보를 찾을 수 있도록 해주는 시스템이다. 검색엔진은 웹 태동으로 탄생했으며 90 년 대 후반 등장한 구글은 지금까지 부동의 1위를 지키고 있으며 2017 년 데스크탑과 모바일에서의 점유율을 보면 87.52% 와 94.67% 라는 절대적인 우위를 유지하고 있다 3). 그러나, 구글이 앞으로도 계속적인 우위를 차지하리라고는 단언할 수 없다. 사물인터넷 환경의 지속적인 빅데이터 생산, 수십 년간 지속적으로 축적된 컴퓨팅 데이터, 스마트 기기와 SNS를 통한 상상을 초월하는 수준의 데이터 생산 등 늘어나는 정보의 양과 웹의 팽창 속도는 지금의 검색엔진 능력으로 따라가기에는 역부족으로 보인다. 증가하는 데이터 속에서 기존 검색엔진으로 적합한 정보를 찾을 수 없는 경우가 빈번하게 발생하고 있기 때문이다. 게다가, 웹으로부터 정보를 얻고자 하는 사용자들의 최근 추세는 일반적이고 방대한 양의 정보보다는 특정 관심 영역이나 분야에 국한된 전문화되고 세분화된 그러면서도 정확한 정보를 선호하는 방향으로 나아가고 있다. 이는 웹에 존재하는 데이터의 양과도 무관하지 않다. 구글이 하루에 처리하는 정보량이 24 페타바이트4) 에 이른다고 하지만 초기와는 다른 검색 속도의 저하, 해당 페이지가 존재하지 않는 등의 유용하지 않은 결과, 광고성 웹 페이지 같은 불필요한 정보 제공 등의 다양한 취약점이 노출되고 있다 [11]. 구글 역시 지속적인 투자로 다양한 검색 결과를 제공하고 있지만 비대해지는 인터넷 데이터에 비해 느린 관리는 검색 엔진 사용자들의 불만을 높이고 있다. 구글이 야후나 알타비스타 같은 1세대 검색 엔진의 약점을 해결하고 지금까지 부동의 1위를 지키고 있듯이 구글 검색 엔진의 한계를 해결하는 독창적인 아이디어를 갖춘 차세대 검색 엔진에 대한 필요성이 높아지고 있다.

### 2-2 검색 엔진의 분류

인터넷 검색 엔진은 동작 방식이나 검색 대상 등에 따라 다양하게 분류할 수 있다. 본 논문은 검색 엔진이 사용하는 검색 대상 즉, 데이터 세트에 따라 다음과 같이 분류한다.

**범용검색엔진 (General Search Engine: GSE)**- 구글, 야후, Bing 등과 같이 전반적으로 모든 관심 영역에 대해서 정보를 검색한다. 일반 사용자들에게 가장 익숙한 검색 엔진으로, 대다수 사용자들의 관심 주제들과 연관이 있는 웹 페이지들을 최대한 많이 수집하여 색인한 후 주제별 그룹으로 나눈 뒤 사용자의 검색어에 따라 결과를 제공한다.

**수직검색엔진 (Vertical Search Engine: VSE)**[12,13] - 영역 특수 검색 엔진 (Domain-Specific Search Engine)이라고도 지칭하는 검색 엔진으로, 모든 관심 영역을 다루는 것이 아닌 특정 관심 분야나 영역에 중점을 두고 정보를 검색한다. 정확하게 한 분야를 겨냥하여 검색하고자 하는 해당 주제를 곧바로 찾아가는 것이다. 이는 최상급의 범용검색엔진에서 대면하게 되는 수십억 개의 검색 결과와는 차별적인 것으로, 특정 목표가 있는 검색에 최적화된 검색 엔진으로 관련된 분야의 보다 정확하고 높은 수준의 검색 결과를 도출하도록 한다. 일반적으로, 사용자 트래픽은 범용검색엔진에 비해서 상대적으로 낮을 수 있으나, 방문자의 관심도는 훨씬 높다. 2-3절과 2-4절에서 수직검색엔진에 대해서 상세히 다룬다.

**메타검색엔진 (Meta Search Engine: MSE)** [14,15] - 여러 검색 사이트에 사용자가 입력한 키워드를 질의하여 실질적인 검색 사이트로부터 검색 결과를 가져와 해당 정보를 보여주는 검색 엔진이다. 실제로는 자체 데이터베이스가 없지만 다양한 정보 검색을 지원해주는 검색 엔진이며, 여러 검색 사이트를 방문할 필요가 없이 한 곳에서 여러 사이트들의 검색 결과를 얻을 수 있다. 다중의 검색엔진을 적용하는 방식으로 3-2절에서 보다 더 자세히 다루도록 한다.

대중적으로 가장 많이 사용되고 있는 검색엔진 유형은 범용검색엔진으로 용이한 사용성, 원활한 접근성, 데이터의 보편성 등의 특징으로 급속하게 성장했지만, 현재의 데이터 빅뱅 상황에서 늘어나는 정보의 양과 웹의 팽창속도를 실시간이 아닌 정기적으로 크롤링되고 색인화 된 검색결과를 제시하는 범용검색엔진이 따라가기에는 역부족인 상황이다. 게다가, 갈수록 증가하는 딥 웹 자료들은 세분화·전문화를 무기로 하여 수준 높은 정보들을 제공하여 사용자의 만족도를 높이고 있지만 대부분의 딥 웹 데이터베이스들은 구글 같은 범용검색엔진 색인에 포함되지 않기에 해당 서비스를 제공받기가 쉽지 않은 상황이다. 이러한 틈을 타고 다양한 차세대 검색엔진 모델들이 전 세계에서 꾸준히 개발되고 있으며 검색엔진 개발자들은 점점 독특하고 개성 있는 아이디어와 기술력으로 구글에 대항하며 구글을 뛰어넘으려는 차세대 검색엔진 구축에 박차를 가하고 있으며 크게 다음 세 가지 특징을 갖는다 [11].

- Better Technology: 보다 편리하게 검색할 수 있는 기술
- Better User Interface: 다양한 사용자 인터페이스를 통한 알기 쉬운 검색결과 제공
- Vertical Search: 특정 정보분야와 전문영역의 검색을 목적으로 하는 수직검색엔진

그 중 딥 웹과 밀접한 연관이 있는 수직검색에 대해 중점적으로 살펴본다.

### 2-3 수직검색엔진 (Vertical Search Engine)

영역특수검색엔진 또는 전문검색엔진이라고도 불리는 수직검색엔진은 특정 관심분야에 중점을 두기 때문에 정보를 제공하는 해당 분야에 대해서는 더 많은 자료들을 보유할 수 있다. 일례로, 구글이 다루는 전체 웹 페이지가 80 억 개 이상인 반면, 자동차·주택 관련 자료만을 다루는 vast.com의 경우 30 억을 훨씬 상회하는 웹 페이지를 보유하고 있으며 이는 계속 증가하고 있다. 단일 영역의 웹 페이지가 차지하는 비율이 구글의 37% 이상인 것이다. 정확한 정보의 빠른 전달 및 광고와 스팸성 페이지 제공을 최소화하는 수직검색엔진들은 검색의 용이성, 양질의 결과와 정렬, 스팸 제거, 그리고 신속한 업데이트라는 장

3) http://gs.statcounter.com/search-engine-market-share
4) 1 Petabyte = 1024 Terabyte, 1 Tetabyte = 1024 Gigabyte

점을 갖는다.

웹에 존재하는 모든 데이터들을 검색하는 엔진은 현재 존재하지 않는다. 광범위에 걸쳐 그리고 유효하면서 정확·신속한 검색을 수행하기 위해 지금의 검색 기술로 적용할 수 있는 최적의 방법은 여러 개의 범용검색엔진과 수직검색엔진들을 동시에 구성하여 결과를 도출하는 것이다. 검색엔진 구성에 앞서서 수직검색엔진이 결과를 도출하는 다양한 방법 [16,17,18,19] 과 많이 사용되고 있는 대표적인 수직검색엔진들에 대해서 먼저 살펴본다.

### 2-4 수직검색엔진 결과 도출 방법

#### 1) API를 직접 이용하는 방법

수직검색엔진에서 선 정의된 모듈형 API를 사용자 응용프로그램에 직접 삽입하여 결과를 도출하는 방법이다. 선 정의된 대부분의 모듈들은 웹기반 응용프로그램에 최적화되어 사용자가 쉽게 모듈 추가를 할 수 있도록 지원해야한다. 따라서, 실제 검색 결과를 가져오는 모듈은 호스팅 서버에 위치하고 그 결과를 요구할 수 있는 코드를 HTML 코드에 내장된 단일 코드로 삽입하거나 또는 PHP 내에서 동작할 수 있는 포맷으로 인코딩한다. 대표적인 케이스가 Eurekster의 Swicki[5]로, 사용자는 자신의 블로그 내에 직접 Swicki 툴바를 설치함으로써 검색 서비스를 제공하게 된다. 그림2는 수직검색엔진에서 API를 직접 사용하여 검색결과를 사용자에게 제공하는 전체적인 진행과정을 보인다.

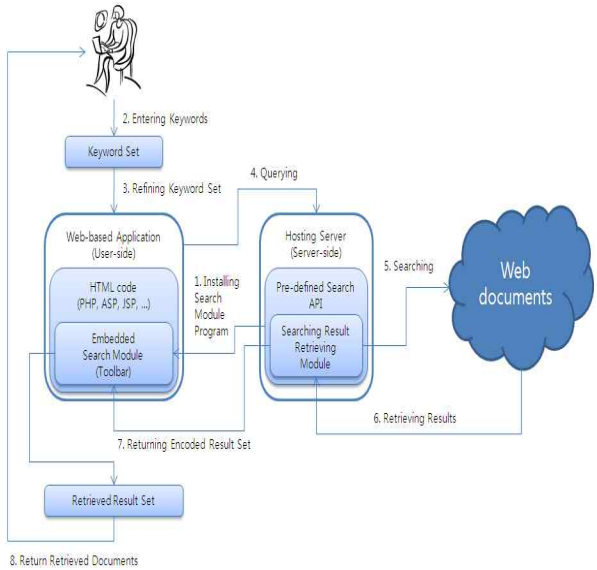


**그림 2.** API를 직접 이용한 검색결과 도출 과정
**Fig. 2.** Retrieval Process Steps by Directly Using API

5) https://techcrunch.com/tag/eurekster/

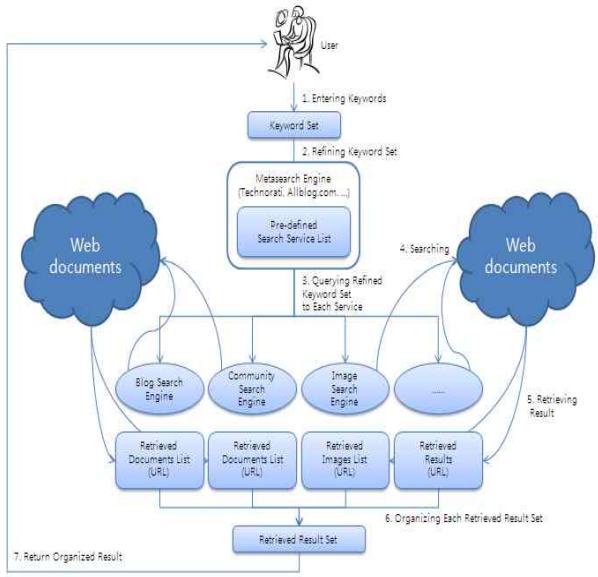


**그림 3.** 메타검색엔진을 사용한 검색결과 도출 과정
**Fig. 3.** Retrieval Process Steps by Using Meta Search Engine

#### 2) 메타검색엔진의 결과를 이용하는 방법

사용자의 응용프로그램에서 직접 메타검색엔진의 결과는 해당 키워드의 결과로써 제공하는 방법이다. 대다수 수직검색엔진들은 특정 도메인, 또는 사업자 내에 국한되어 분야별 딥웹 데이터베이스를 통한 양질의 결과를 얻어내는 것을 목표로 하기에, 전체 웹 공간에서의 정보 탐색이라는 측면에서는 약점을 가질 수 있다. 이를 해결하기 위한 방법이 메타검색엔진의 적용이다. 사용자가 검색 키워드를 입력하면 단일 메타검색엔진에서 사전 정의된 수직검색엔진 집합들 대상으로 해당 키워드를 입력하였을 때의 URL을 사용자에게 제공하는 방식이다. 대표적인 예로써 메타 블로그 검색엔진들이 있는데, 블로그 사업자들이 자체적으로 제공하고 있는 검색엔진들의 URL을 미리 확보하여, 사용자가 자신의 응용프로그램 내에서 검색 키워드를 입력하면 동일 키워드를 블로그 사업자가 만들어 놓은 검색엔진에 그대로 반영 후 결과 URL을 사용자에게 검색 결과로 직접 제공한다. 그림3은 메타검색엔진의 결과를 사용하여 수직검색엔진을 통해 결과를 제공하는 전체적인 과정에 대한 도식이다.

#### 3) 범용검색엔진의 결과를 이용하는 방법

구글이나 야후 등의 범용검색엔진을 이용하여 개개의 수직검색엔진의 결과를 이끌어 내는 방법이다. 기존 범용검색엔진 결과와의 차이는 범용검색엔진을 통해 도출된 결과들이 한 단계 더 캡슐화 (encapsulation) 된다는 것이다. 이 경우, 사용자가 직접 수직검색엔진들을 방문하여 키워드를 입력하는 결과와 범용검색엔진을 통해 나오는 결과가 상이할 수 있다는 차이점이 있지만, 일반적인 검색엔진은 메타검색엔진에서 도출하는 콘텐츠들을 거의 포괄하고 있기 때문에 검색결과의 질적인 차

이점은 그다지 크지 않다고 할 수 있다. 게다가 대부분의 검색 사업자들은 사용자의 응용프로그램에 쉽게 자사의 검색 모듈이 포함될 수 있도록 관련 API를 제공하는데, 구글 API가 그 대표적인 예이다. 특히 구글은 API 계층조차 사용자가 신경 쓰지 않도록 하기 위해 단순히 HTML 코드 추가를 통해 사용자의 블로그나 해당 블로그가 속해 있는 사회관계망 내에서의 검색이 가능하도록 관련 기능을 제공하고 있다. 그러나, 범용검색엔진을 적용한다는 것은 범용검색엔진이 가지고 있는 단점 또한 그대로 포함하고 있다는 것으로, 수직검색엔진들은 범용검색엔진이 자체적으로 색인하고 있는 수많은 웹 문서들을 검색해야 하며 사용자의 판단에 따라 검색 결과들을 재평가해야 한다는 문제점이 존재한다. 그림 4는 범용검색엔진을 이용하여 수직검색엔진의 결과를 도출해내는 과정을 보인다.

## Ⅲ. 다 중 검 색 엔 진 구 성

펜실베니아 대학의 연구 보고서[20]에 의하면 범용검색엔진의 대표 주자들인 구글, 야후 그리고 Ask가 동일 질의어에 대해 제공한 결과들 중 각 엔진의 첫 번째 페이지에 나열된 결과들만 비교했을 때, 단지 3%의 결과만 동일했으며 85%의 결과들은 세 개의 엔진 중 하나에서만 결과로 제공되었다고 한다. 최대한 많은 영역을 취급하는 범용 엔진들은 자료의 양은 방대해 보이지만 각 영역에 대해 색인하고 있는 자료들은 극히 미미한 양이기 때문에 이러한 결과를 보이는 것이다. 그에 비해 수직검색엔진은 딥 웹을 기반으로 하므로 단일 영역에 대한 데이터 보유량은 범용검색엔진의 해당 영역 보유량을 압도하지만 타 영역들에 대한 데이터는 거의 보유하지 않는다는 단점을 갖는다.

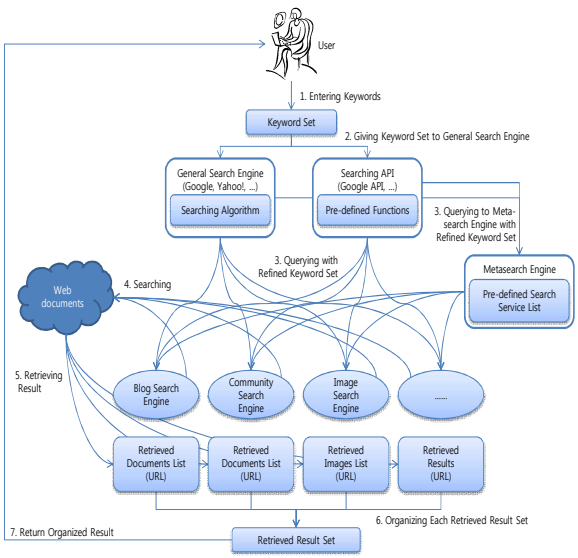


**그림 4.** 범용검색엔진을 사용한 검색결과 도출 과정
**Fig. 4.** Retrieval Process Steps by Using General Search Engine

각 검색엔진들의 웹 영역별 중복성은 미미하기 때문에, 각 결과들을 종합하여 사용자에게 제공할 경우 개별적인 결과보다는 더 넓은 범위의 그러면서도 세부적이고 정확한 정보들을 포함하는 검색결과 제공이 가능해진다. 결과들을 취합할 때 주요하게 다루어져야 할 것은, 주어진 질의어에 얼마나 정확하게 부합하는 정보인지, 최신의 정보를 반영하는지, 그리고 복합 질의어들에 대해서 키워드들 간의 연관성을 얼마나 적절하게 반영하여 검색을 수행하는지 등을 심도 있게 고려해야 한다. 다중검색엔진 구성 방식은 크게 세 가지로 구분할 수 있다; 다중범용검색, 다중수직검색, 그리고 하이브리드검색이다. 최근 검색엔진들이 나아가는 추세는 다중수직 또는 하이브리드 검색엔진으로 얼마나 효과적으로 딥 웹 데이터베이스 적용을 할 수 있는가가 큰 비중을 차지한다고 할 수 있다.

### 3-1 다중범용검색엔진 vs. 다중수직검색엔진

구글, 야후, Ask 등의 범용검색엔진을 개별적으로 사용하는 것이 아니라 동일하게 같이 사용하여 검색결과를 도출하는 방식이 다중범용검색엔진 구성으로 이 방식은 다시 1) 사용자 스스로 이용하고자 하는 검색엔진들을 동적으로 선택 후 추가하는 방법과 2) 범용검색엔진 자체에서 정해진 타 범용검색엔진들을 사용하는 방법으로 나뉜다. 그러나, 다중범용검색엔진 방식은 거의 사용되지 않는다. 왜냐하면 범용검색엔진들은 전체 웹의 4% 정도만을 차지하는 표면 웹의 데이터들을 색인하는 이유로 검색결과의 전문성이 현저히 떨어지기 때문이다.

다중수직검색엔진 방식은 여러 개의 특화된 수직검색엔진들을 사용하는 방법으로 모든 도메인들에 적용되는 것이 아니라, 의학이면 의학, 과학이면 과학, 컴퓨터면 컴퓨터로 특정 도메인에 대한 검색만을 수행하되 범용검색엔진보다는 더 전문적으로 그리고 최대한의 스팸 제거 하에 결과를 도출하게 된다. 다중수직검색엔진 방식은 수직검색엔진들의 그룹화에 따라 크게 두 가지로 분류된다.

1) 단일 도메인에 해당하는 수직검색엔진들만을 그룹화
   예. 여행관련 검색엔진, 의학관련 검색엔진 등
2) 여러 도메인에 해당하는 수직검색엔진들을 유용하다고 생각되는 카테고리로 그룹화
   예. Search Engine Gide, Virtual Free Sites 등

첫 번째 방법의 경우, 특정 도메인을 정한 후 해당 도메인에 대한 보다 전문적이고 자세한 검색결과를 제공하는 것이 목적으로, 특정 도메인에 대한 검색을 제공하는 여러 개의 단일수직검색 사이트들을 나열하는 것이 특징이다. 그러나 이 방법은 수직검색엔진들을 그룹화하는 하나의 사이트가 특정 도메인을 색인하는 수직검색 사이트들을 모아 놓은 후 해당 엔진의 특징들을 간략하게 기술해 놓아 사용자가 자신이 원하는 도메인의 수직검색엔진들을 선정해 놓은 사이트를 개별적으로 검색하여 찾아야 한다는 문제점이 있기에 사라진 구성 방식이다.

개선한 방법이 두 번째 방식으로 수많은 수직검색엔진들을 색인하는 도메인에 따라 그룹화하여 정리한 사이트를 제공하

는 방법으로 SearchEngineGuide를 예로 들 수 있다. 대분류로 나뉜 디렉터리들은 다시 중분류로 나뉘어 특정 수직검색 사이트들을 관리한다. 대부분의 메타검색엔진이 이 방식에 속하며 클러스터링 엔진이 파생되었다.

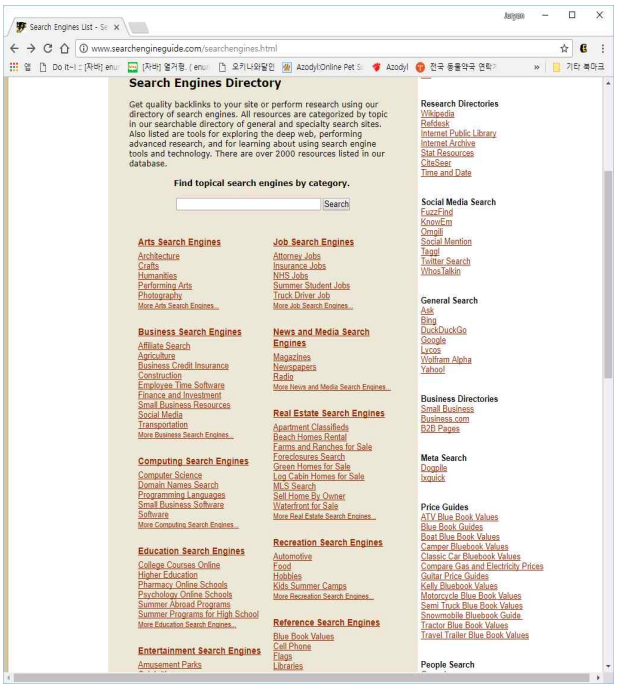


**그림 5.** SearchEngineGuide의 대분류 디렉토리 일부
**Fig. 5.** Partial Directories of a Category of SearchEngineGuide

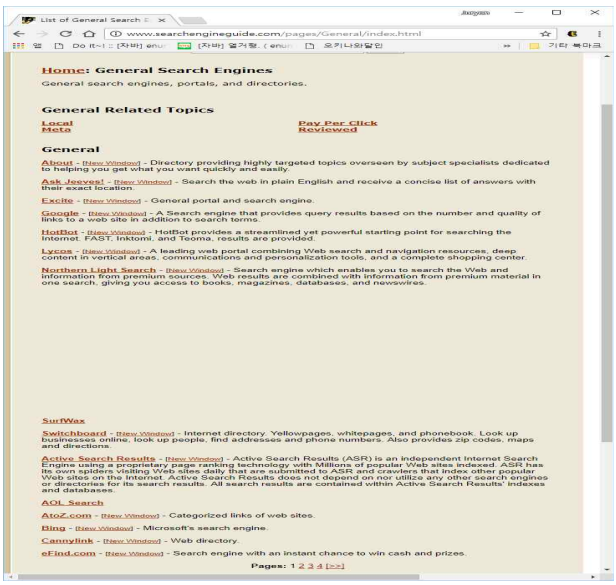


**그림 6.** 주제 General Search Engine에 대한 수직검색사이트들
**Fig. 6.** Vertical Search Sites of the Topic 'General Search Engine'

그림 5는 SearchEngineGuide의 대분류 디렉터리들의 일부를 캡쳐한 것이며 그림 6은 대분류 General Search Engine을 선택 시 보이는 서브 주제와 관련된 모든 수직검색사이트들을 페이지별로 정리해서 보이는 화면이다. 디렉터리 분류는 정리한 사이트에 따라서 약간의 차이점이 있지만, 대부분 계층 구조를 따른다. SearchEngineGuide의 경우, 대분류로 나뉜 디렉터리들을 다시 중분류로 나눠 사용자가 좀 더 세부적으로 수직검색사이트를 찾을 수 있도록 관리하는 것을 알 수 있다.

### 3-2 클러스터링 검색엔진

범용검색엔진과 수직검색엔진의 장점을 적용하여 구축하는 검색엔진이다. 메타검색엔진의 일환이지만 범용검색엔진의 특징을 많이 적용한 엔진으로, 범용검색엔진의 광범위한 데이터에 수직검색엔진의 전문성과 정확성을 접목하는 방법이다. 예를 들어, '딸기에 들어있는 비타민'에 대한 정보를 검색하고자 한다면, Google에서 검색한 '딸기에 들어있는 비타민'에 대한 결과들과 의학 관련 수직검색엔진들에게서 검색한 '딸기에 들어있는 비타민'에 대한 결과들을 서로 병합하여 최종적으로 사용자에게 결과를 제공하는 방식이다. 이 때, 스팸성 자료들이 최대한 제거되는 것을 보장해야 하며 최종 결과들 또한 검색자의 요구 사항에 가장 근접한 결과들이어야 한다. 클러스터링검색엔진의 대표라 할 수 있는 것은 Vivismo 사에서 개발한 Clusty로, 현재는 IBM Yippy 로 업그레이드되었다. 본 절에서는 Clusty와 Yippi 모두 살펴보기로 한다. 그림 7은 이전 엔진 Clusty를 이용하여 Barack Obama를 검색한 결과의 일부이다 (Clusty는 현재 사용할 수 없다).

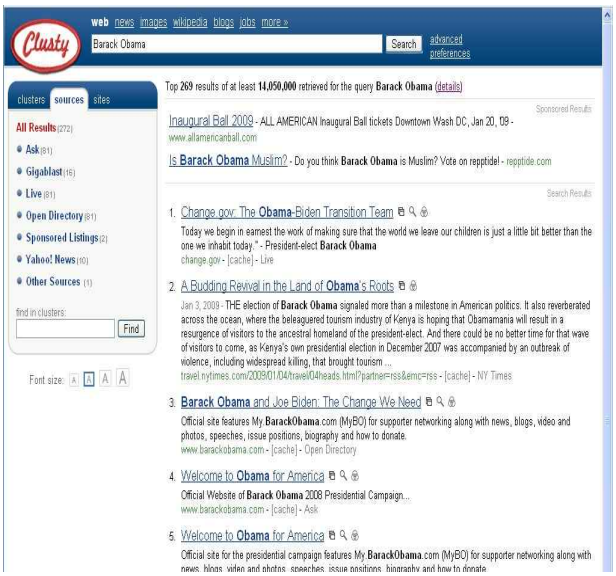


**그림 7.** Clusty를 사용한 'Barack Obama' 검색 결과
**Fig. 7.** Search Result of Clusty for 'Barack Obama'

좌측 메뉴 중 sources를 클릭하면 어떤 검색엔진들을 근원으로 하여 결과를 도출했는지 확인할 수 있는데 범용검색엔진인 Ask와 수직검색엔진의 영역을 그룹화 하는데 기준이 되는 Open Directory가 포함되어 있는 것을 볼 수 있다. 또한 스팸이나 광고성 페이지 제공을 최소화하기 위해 Sponsored Listings 라는 범주가 존재한다. 그 외, 질의어와 관련된 정보를 다수 제공하는 수직검색엔진들을 순차적으로 sources에 나열한다. 그러나 Clusty에는 몇 가지 문제점이 있다. 메뉴 sources에 나열된 범주는 일반적인 기준에 근거한 분류이기에 검색자는 스스로 자신의 질의어에 대해 가장 정확하고 많은 정보를 제공하는 범주를 찾아야 한다는 것이다. 만약 Sponsered Listings 에 관련된 많은 데이터가 들어있다면 검색자는 이 범주 역시 일일이 검토해보아야 한다. 이는 모든 메타검색엔진들이 갖는 문제점이기도 하다. 또 다른 문제는 복합 키워드를 사용하여 검색 수행 시 키워드들 간의 연관성을 적용하여 결과를 제공하는 대신 각 키워드의 빈도수를 주요하게 반영하여 검색을 수행하기 때문에, 검색자 의도에 부합하는 정보를 찾기에는 역부족이라고 할 수 있다. 그림 8은 이에 대한 Clusty의 문제점을 잘 보여준다. 검색자는 딸기에 들어있는 비타민 관련하여 의학적이면서도 건강과 관련된 정보들을 찾고자 'vitamin' 'strawberry'라고 키워드를 입력하였다. 관련 데이터에 대한 정보를 제공하는 검색엔진들의 개수가 앞 장의 단일 키워드에 비해 현저하기 적을 뿐만 아니라, 검색된 웹 페이지들은 주로 쇼핑과 vitamin C 약에 관한 정보가 대부분이다. 즉, 검색자가 의도한 목적과는 거리가 먼 전혀 별개의 검색 결과가 얻어진 것이다.

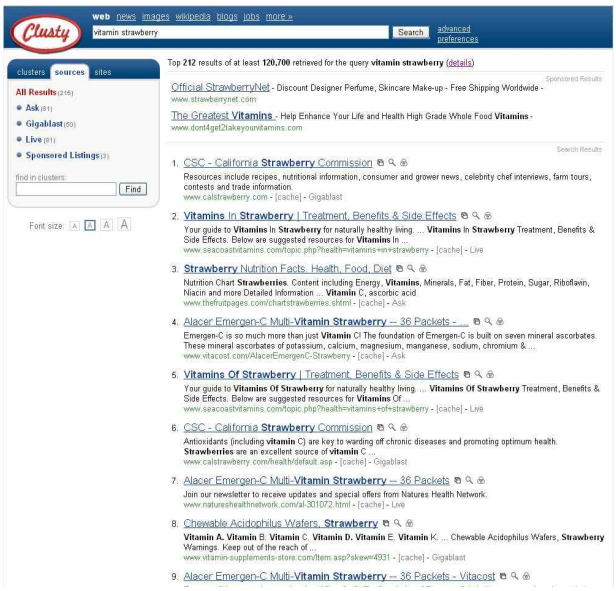


**그림 8.** 복합질의어에 대한 Clusty 검색결과
**Fig. 8.** Search Results of Clusty for the Compound Keywords

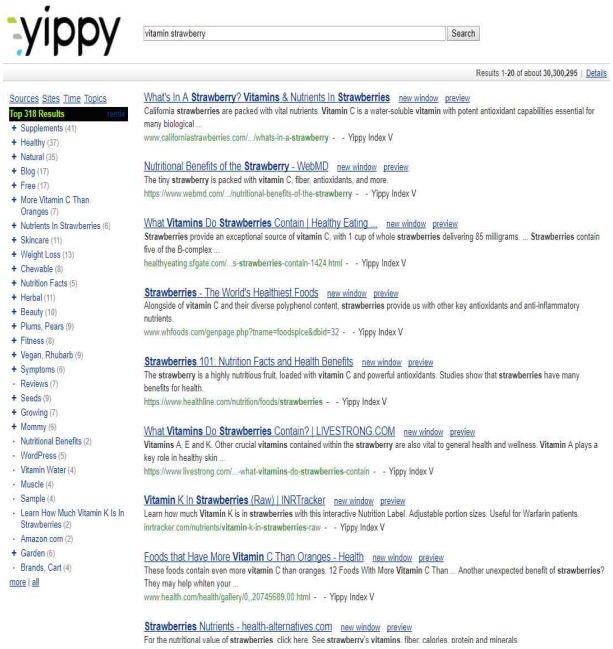


**그림 9.** 복합질의어에 대한 Yippy 검색결과
**Fig. 9.** Search Results of Yippy for the Compound Keywords

**그림 10.** 검색엔진 Yippy의 메뉴 탭들
**Fig. 10.** Menu Tabs of the Yippy

2012 년 IBM에 인수된 Clusty 는 Yippy로 변경된다. 단순히 이름만 변경된 것이 아니라 Search Everything 표방 하에 웹 데이터 및 딥 웹 DB들을 최대한으로 색인하거나 연합하여 강력한 검색 결과 제공을 목표로 한다. 그림 9는 구 버전인 Clusty 와 동일한 복합 키워드 'vitamin'과 'strawberry'를 Yippy에 입력한 결과이다. 가장 큰 차이점을 보이는 부분은 좌측 메뉴이다. Topics 메뉴 탭이 추가되었으며 기본 결과로 제공되는데 기존 Clusty 의 Sources 의 범주가 적용되는 탭이다. 그러나 Clusty 와는 다르게 범주가 다양화·세분화 되었으며 각 범주별 하위분류 또한 전문성에 근거하여 생성되었다.

그림 10은 Sites, Time 그리고 Sources 탭 각각에 대한 검색결과 범주를 보인다. Clusty와 달리 Sites는 도메인 유형 또는 국

가별로 분류되었으며 Sources 탭은 일반 색인과 수직검색 색인 두 종류로 단순분류 하였다. Time 탭은 새롭게 추가된 것으로 최근 일주일부터 과거 90일 동안의 시간에 따라 결과를 분류한다. 그러나, Yippy 역시 메타검색엔진에 속하기 때문에 검색자가 범주별로 확인하며 의도에 맞는 검색결과를 선택해야하는 문제점은 여전히 존재한다. Yippy처럼 다중수직검색엔진을 이용한 전문성·정확성에 Google과 같이 범주를 분류하지 않는 간편성을 겸비한 검색엔진에 대한 고찰이 필요하다.

## Ⅳ. 하 이 브 리 드 검 색 엔 진

### 4-1 필요성

빅데이터, IoT와 결부되어가는 웹은 점점 더 다양하고 세분화된 딥 웹 영역 데이터를 빠르게 양산하고 있다. 이는 검색에 있어서도 과거와는 다른 추세를 보이는데 그것은 바로 검색결과의 전문성·정확성 등에 많은 중점을 둔다는 점이다. 대량을 넘어서는 웹 데이터의 생산은 검색자에게 진짜 필요한 정보가 무엇인지 스스로 찾아야 하는 검색에 대한 피로도를 빠르게 증가시킨다. 새로운 검색엔진에 대한 요구가 급격하게 증대하는 중요 이유이기도 하다.

하이브리드검색엔진 역시 다중검색엔진처럼 둘 이상 검색엔진의 병렬 사용을 기반으로 한다. 차이점은 검색엔진 종류와 결과 제공 방식에 있다. 하이브리드검색엔진은 구 Clusty처럼 범용과 수직검색엔진을 모두 적용한다는 것과 범주 분류 없이 Google처럼 결과를 제공한다는 점이다. 즉, 주어진 질의어에 대한 검색을 서로 다른 성격을 갖는 다수의 검색엔진들이 수행 후 해당 결과를 통합하여 단일화 된 리스트를 검색자에게 제공하는 방식이다. 현재 사용되고 있거나 개발되고 있는 검색엔진들 중 웹에 있는 모든 데이터들을 검색하는 엔진은 존재하지 않는다. 최상의 방식은, 최대한의 검색엔진들을 사용하여 검색자의 피로도를 최소한으로 하는 검색엔진의 개발이다.

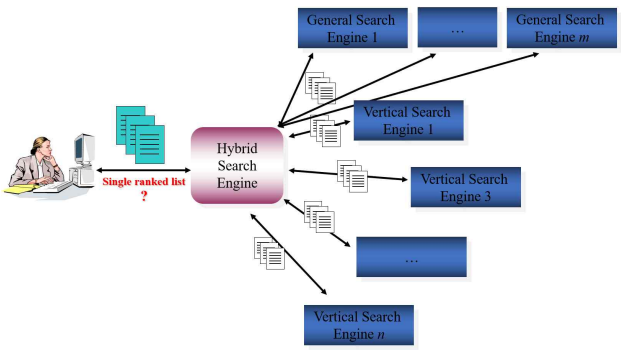


**그림 11.** 하이브리드검색엔진 개요도
**Fig. 11.** The Outline of a Hybrid Search Engine

표면 웹 데이터들을 주관하는 범용검색엔진과 딥 웹 데이터베이스들을 분야별로 관장하는 수직검색엔진들과의 협업을 기반으로 하는 하이브리드검색엔진에서 주요하게 다루어져야 하는 과정이 각 검색엔진들로부터 전달받은 개별적인 검색 결과들을 어떤 기준에 따라 통합하고 정렬하여 순서를 정해야 하는가이다. 그림 11은 이를 반영한 하이브리드검색엔진의 개략적인 구성을 보이는 개요도이다. 앞서 언급했듯이, 가장 주요하게 처리되어야 하는 두 과정은 통합을 위한 병합 (Merging) 과정과 정렬을 위한 서열화 (Ranknig) 과정으로, 검색결과가 얼마나 정확한지 그리고 검색엔진의 특성에 얼마나 부합한 알고리즘을 선택하느냐에 따라 검색자에게 제공되는 최종 결과에 대한 신뢰성과 정확성이 결정된다. 검색엔진에 적용되는 다양한 병합과 서열화 전략들[21,22,23,24,25,26]을 주요 특징에 따라 4-2절에서 정리한다.

### 4-2 주요 관련 기술

1) 병합 (Merging) 전략

검색결과들을 통합하는데 있어서 개개의 검색엔진들이 얼마만큼의 정보를 병합하는 엔진에게 전달하느냐에 따라 크게 두 방향으로 나뉘는데 융합방식 (Integrated Method)과 고립방식 (Isolated Method)이다. 하이브리드검색엔진이나 메타검색엔진처럼 병합이 수행되어야 하는 검색엔진들은 질의에 대한 검색을 수행하는 말단 검색엔진들에게 병합을 위한 특별한 정보 또한 같이 제공해 줄 것을 요구하는데 이것이 융합방식이다.

특별한 정보는 문서 빈도를 위해 필요한데 문서 빈도란, 주어진 질의어나 용어가 최소 한 번 이상 발생한 웹 문서들의 개수로, 이는 질의어에 대한 상대적인 중요성을 측정하는 가중치로 사용되거나 해당 문서와 주어진 질의어 간의 상대적인 연관 가능성을 측정하는 용도로 쓰인다. 융합방식은 이러한 문서 빈도를 얻거나 합치기 위해 말단 검색서버들의 능력과 특정 프로토콜들을 사용한다. 따라서, 말단 검색서버들이 얼마나 정확하게 그리고 빠르게 문서 빈도를 제공하는가 여부에 따라 통합하는 서버 쪽에서의 검색결과에 상당한 영향을 미치게 된다.

고립방식은 각 말단 검색서버에서 언제든지 제공할 수 있는 정보들을 사용하여 통합과 정렬을 수행하는 방식으로 말단 서버 개개의 특정 능력이 필요하지 않다. 언제든지 제공받을 수 있는 정보원들은 세 종류로 분류 가능하다; ① 말단 검색서버에 의해 문서에 할당된 일반적위 순위, ② 말단 검색서버에 의해 문서에 할당된 스코어, ③ 서버로부터 사용자에 의해 직접 다운로드 된 웹 문서의 내용들이 해당된다. 정보원 분류에서 스코어란 유용성 측정을 위해 사용되는 수치를 의미한다. 사용자로부터 질의어를 입력받게 되면 해당 질의어를 대표하는 성격을 갖는 정보가 각 검색엔진이 해당 질의어와 관련해서 얼마나 필요한 정보를 보유하고 있는지 수치로 계산하는데 이것이 스코어이다. 스코어에 따라 개개의 검색엔진들이 주어진 질의어에 대한 결과 제공자로서의 가·부가 결정된다고 할 수 있다. 고립방식은 통합하는 엔진 쪽에서는 말단 검색엔진들의 능력에 신

경 쓸 필요 없이 기본적으로 제공하는 정보들 중에 적합한 것을 선택해서 넘겨받은 후 통합하는 엔진에서 정렬을 수행하는 방식으로 융합방식보다 선택의 폭이 다양하다. 고립방식이 융합방식보다 자주 적용되기 때문에 고립방식에 기반한 대표적인 서열화 방식들을 소개한다.

2) 서열화 (Ranking) 기술

고립방식에 근거하여 자주 사용되는 서열화 기술들은 5가지 정도로 정리해 볼 수 있다; 직접계산 (straightforward)방법, 순위이용방법, 문서요약방법, 문서순위 및 요약방법, 웹문서 이용방법. 각 방법에 대해서 조금 더 자세히 알아본다.

**직접계산(Straightforward)방법** 가장 단순한 방식으로 통합을 담당하는 엔진에서 모든 말단 검색엔진들로부터 질의어에 대한 결과를 제공받은 후, 유사성 측적 알고리즘을 사용하여 질의어와의 관련성을 직접 계산하여 해당 관련성을 기준으로 검색결과들의 순위를 정렬하는 방식이다. 가장 간단해 보이는 방법이지만, 최종적으로 결과를 제공받는 사용자 입장에서는 오랜 시간 해당 결과를 기다려야 한다는 심각한 문제와 통합 엔진이 모든 과정을 다 수행해야 하므로 엔진이 무거워진다는 문제점이 발생한다.

**순위이용방법** 말단 검색엔진들은 자체적으로 자신들의 검색결과에 순위를 부여하는데 이를 통합 엔진에서 재정렬하는 방식이다. 서로 다른 검색서버에서 제공받은 동일한 결과의 웹 문서들은 순위를 합산하게 되며, 순위의 숫자가 높을수록 상위에 위치한다. 각 말단 검색서버에서 한 번씩만 발생한 문서들은 최종 결과에서 자신이 부여받은 순위 숫자에 따라 적절한 위치에 삽입된다. 하나 이상의 검색서버에서 발생한 문서들의 경우, 얼마나 많은 수의 서로 다른 검색서버에서 발생했는지가 해당 문서의 최종 순위를 결정짓는다고 할 수 있다.

**문서 제목과 요약 이용방법** 각 문서의 제목과 요약을 함께 사용해서 문서를 색인하는 방식으로, 제목과 요약은 각 말단 검색서버의 결과 리스트에서 제공되는 부분이다. 특히 요약은 주어진 질의어를 해당 문서의 어느 부분에서 포함하고 있는지를 보이는 내용이다. (제목, 요약) 쌍을 이용하여 계산된 벡터 값은 질의어와 얼마나 관련이 있는지를 나타내는 가중치 역할을 수행하며 해당 색인을 대표한다. 같은 URL을 갖는 문서지만 검색서버에 따라 서로 다른 요약을 갖는 경우, 요약 부분은 서로 연결되어 하나로 합쳐진 후 벡터 값이 계산되는데 요약이 길수록 벡터 수치는 높아져서 결과적으로 가중치를 높이는 효과를 갖는다.

**문서 순위, 제목, 요약 이용방법** 문서 제목과 요약 이용방식에 순위를 추가해서 좀 더 정확한 가중치를 계산하고자 하는 방식으로 최종적인 결과 작성에 있어서 효율성을 높이고자 한 방식이다.

**웹 문서를 직접 다운로드 하는 방법** 앞서 열거한 방식들은 말단 검색서버에서 기본적으로 제공하는 정보들에 의존하는 공통점을 갖는다. 즉, 결과 문서 자체를 사용하여 병합과 정렬을 수행하지 않는다. 이 점에 착안해서 제안된 방식이, 웹 문서 자체를 직접 다운로드 하여 분석 후 최종 통합 리스트의 문서 결과 순위를 정하는 방식이다. 그러나 이 방식은 실시간 다운로딩과 분석이 반드시 제공되어야 한다는 문제가 발생하며 이는 결론적으로 실시간 제공을 위한 추가적인 데이터 전송용 대역폭과 고사양의 컴퓨팅 환경을 요한다.

검색엔진의 특징, 데이터 크기 등 사용 환경에 따라 최적의 결과를 보이는 서열화 기술을 적용해야 하며 시뮬레이션이 필연적으로 동반되어야 한다.

### 4-3 하이브리드 검색엔진 아키텍쳐

그림 12는 지금까지 정리한 관련 기술들 및 필요성에 근거한 하이브리드검색엔진에 대한 프레임 아키텍쳐이다. 대표적 범용검색엔진인 Google과 다수의 수직검색엔진들을 병렬로 사용하여 구성한다. 본 논문에서 정리한 하이브리드 검색엔진 기반 시스템의 전반적인 아키텍쳐로 사용자, Google, 다수의 수직검색엔진들, 그리고 결과 웹 페이지들이 어떤 방법으로 연계되어 있으며 어떻게 사용자 질의어를 웹에 보내어 최종 결과를 도출하는지를 개괄적으로 보이는 전체적인 과정을 담고 있는 블록도이다. Google과 각 수직검색엔진들을 하이브리드 방식으로 연계하는데 있어 주요한 역할을 담당하는 부분은 검색엔진 API 부분으로 해당 API가 실질적으로 검색을 수행하는 말단 수직검색엔진들을 작동하게 하는 역할을 한다고 볼 수 있다. 또한 선 정의된 주요 관심 분야 범주와 질의어에 따라 실시간으로 범주를 구성하는 클러스터링을 통해 최적의 수직검색엔진을 선정하도록 구성하였으며 이 부분이 하이브리드 검색 엔진의 성공적인 검색결과를 제공한다고 할 수 있다.

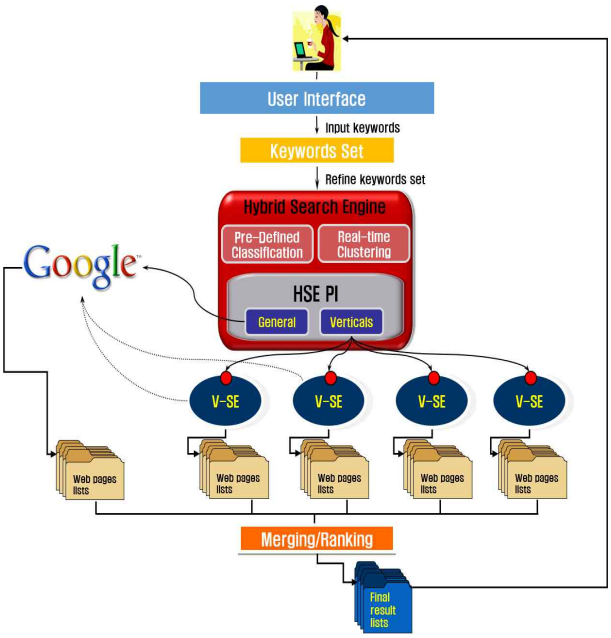


**그림 12.** 하이브리드검색엔진 프레임 아키텍쳐
**Fig. 12.** The Frame Architecture of Hybrid Search Engine

본 논문에서 제안하는 아키텍처가 기존 검색엔진들과 비교하여 갖는 차별성은 다음과 같다. 우선 범용검색엔진에서 제공하는 API 수정과 주요 수직검색엔진들을 통합하는 API를 자체 개발하여 두 종류의 API를 하이브리드 검색엔진에서 적용한다는 것이다. 즉, 범용뿐만 아니라 수직엔진의 결과를 응용프로그램들에 삽입 가능하게 한다는 것이다. 둘째는, 사용자가 입력한 검색어들에 기반을 둔 범주 생성에 있어서 자동범주 생성과 개발자의 개입을 허용하는 전문범주 생성을 가능하게 함으로써 검색 결과의 분류 체계 개발에 있어서 정확성과 전문성을 강화한다는 것이다. 셋째로, 사용자 인터페이스 설계에 있어서 범용과 수직검색엔진들의 백-엔드 사용을 보장하도록 한다는 것이다.

## Ⅴ. 결 론

검색엔진의 킬러 애플리케이션이라 하는 Google의 검색결과는 전체 웹의 10% 미만을 차지하고 있는 표면 웹에 국한됨에도 불구하고, 스마트 기기와 SNS를 통한 상상을 초월하는 수준의 데이터 생산 등 늘어나는 정보의 양과 웹의 팽창 속도는 Google의 능력으로도 따라가기에는 역부족으로 보인다. 증가하는 데이터 속에서 기존 검색엔진으로 적합한 정보를 찾을 수 없는 경우가 빈번하게 발생하고 있기 때문이다. 현존하는 최고의 검색엔진이라는 Google 만으로는 현재의 웹 보유 데이터는 물론 앞으로도 빠르게 증가할 전체 웹 데이터를 모두 다룬다는 것은 불가능하며 이는 특히 딥 웹이라는, Google조차도 검색이 힘든, 웹의 대부분을 차지하고 있는 세계와 무관하지 않다. 딥 웹에 존재하는 정보들은 일반적으로 특정 도메인 영역만을 검색하는 수직검색엔진들을 사용하여 검색을 수행하며 수직검색엔진들은 자신만의 딥 웹 데이터베이스를 구축하여 Google 같은 범용검색엔진들에서 나타나는 문제점들을 최소화한다. 그러나, 검색 영역은 극히 제한적·전문적이다.

본 논문은 Google 등의 범용검색엔진이 갖는 광범위성·검색결과의 친숙성 등에 전문검색엔진의 정확성·전문성 등을 결합하여 리스트 제공을 목적으로 하는 하이브리드검색엔진 프레임을 제안하였다. 프레임에 기반하여 설계와 개발을 구체화하고 있으며 현재 주요하게 연구하고 있는 부분은 범용검색엔진과 전문검색엔진들의 통합 API 관련 부분과 100,000 개 이상 난립하고 있는 전문검색엔진들 중 양질의 엔진들을 선정하는 부분이다. 이와 별도로, 검색 결과의 가장 많은 영향을 끼치는 병합전략 및 서열화전략에 대한 연구도 병행하고 있다.

## 감사의 글

본 연구는 2016학년도 평택대학교 학술연구비의 지원에 의하여 연구되었음.

## 참고문헌

[1] Total number of websites. Available: http://www.internetlivestats.com/total-number-of-websites/

[2] C. Asselin, Discover and exploit the invisible web for competitive intelligence, Digimind, New York, 2006.

[3] M. K. Bergman (2001, August). “The deep web: surfacing hidden value,” The Journal of Electronic Publishing [Online]. 7(1). Available: https://brightplanet.com/2012/06/the-deep-web-surfacing-hidden-value/

[4] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang, “Accessing the deep web,” Communications of the ACM, Vol. 50, No. 5, pp. 95-101, May 2007.

[5] Y. Ru and E. Horowitz, “Indexing the invisible web: a survey,” Online Information Review, Vol. 29, No. 3, pp. 249-265, 2005.

[6] A. Alba, V. Bhagwan, and T. Grandison, “Accessing the deep web: when good ideas go bad,” in Proceedings Companion to the 23rd ACM SIGPLAN Conference on Object-Oriented Programming Systems Languages and Applications, Nashville, USA, pp. 815-818, October 19-23, 2008.

[7] A. Ghani (2017, March). How to access the deep web safely [Internet]. Available: http://www.techglows.com/access-deep-web-safely/.

[8] S. Lawewnce and CL. Giles, “Searching the world wide web,” Science Magazine, Vol. 280, No. 5360, pp. 98-100, April 1998.

[9] A. Gulli and A. Signorini, “The indexable web is more than 11.5 billion pages,” in Proceedings of the 14th International Conference on World Wide Web, Chiba, Japan, pp. 902-903, May 10-14, 2005.

[10] W. B. Croft, D. Metzler, and T. Strohman, Search engines information retrieval in practice, Pearson, pp. 1-28, 2009.

[11] ATLAS Research & Consulting, The global trends for the post-Google and the requirements for the next generation of the search engines, DigiEco, June 2008. Available: http://digieco.co.kr/KTFront/report/report_issue_trend_view.action?board_id=strategy&board_seq=756&sort_order=new&list_page=#

[12] B. A. Galitsky and B. Kovalerchuk, “Building a repository of background knowledge using semantic skeletons,” in Proceedings of AAAI Spring Symposium 2006 – Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, CA, USA, pp. 22-27, March 27-29, 2006.

[13] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, “A

machine learning approach to building domain-specific search engines," in Proceedings of the 16th International Joint Conference on Artificial Intelligence, Vol.2, Stockholm, Sweden, pp. 662-667, July 31 − August 6, 1999.

[14] E. J. Glover, S. Lawrence, W. P. Birmingham, and C. L. Giles, "Architecture of a metasearch engine that supports user information needs," in Proceedings of the 8th International Conference on Information and Knowledge Management, Missouri, USA, pp. 210-216, November 2-6, 1999.

[15] M. Nanoj and J. Elizabeth, "Information retrieval on internet using meta-search engines: a review," Journal of Scientific & Industrial Research, Vol. 67, No. 10, pp. 739-746, October 2008.

[16] R. Shettar and R. Bhuptani, "A vertical search engine − based on domain classifier," International Journal of Computer Science and Security, Vol. 2, No. 4, pp. 18-27, November 2008.

[17] M. Cui and S. Hu, "Search engine optimization research for website promotion," in Proceedings of International Conference on Information Technology, Computer Engineering and Management Sciences, Jiangsu, China, pp. 100-103, September 24-25, 2011.

[18] G. Luo, C. Tang, H. Yang, and X. Wei, "MedSearch: a specialized search engine for medical information retrieval," in Proceedings of the 17th ACM Conference on Information and Knowledge Management, California, USA, pp. 143-152, October 26-30, 2008.

[19] X. Y. Xu and D. Zhao, "Research on the development of vertical search engines," Advances in Future Computer and Control Systems, Vol. 1, pp. 579-584, 2012.

[20] Dogpile.com, University of Pittsburgh, and Pennsylvania State University, Different Engines, Different Results, DOGPILE, 30 pages, April 2007.

[21] F. Yuan and J. Wang, "An implemented rank merging algorithm for meta search engine," in Proceedings of International Conference on Research Challenges in Computer Science, Shanghai, China, pp. 191-193, December 28-29, 2009.

[22] Y. Lu, W. Meng, L. Shu, C. Yu, and K.-L. Liu, "Evaluation of result merging strategies for metasearch engines," Lecture Notes in Computer Science, Vol. 3860, pp. 53-66, 2005.

[23] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswll, "LambdaMerge: merging the results of query reformulations," in Proceedings of the 4th ACM International Conference on Web Search and Data Mining, Hong Kong, China, pp. 795-804, February 2009.

[24] H. Jadidoleslamy, "Search result merging and ranking strategies in meta-search engines: a survey," International Journal of Computer Science Issues, Vol. 9, No. 3, pp. 239-251, July 2012.

[25] M. Khaled Abd El-Fatah, Merging multiple search results approach for meta-search engines, Doctoral Dissertation, University of Pittsburgh, School of Information Sciences, PA, January 2006.

[26] S. Oh and B. Kim, "Query processing model for internet ontology data change," Journal of Digital Contents Society, Vol. 17, No. 1, pp. 11-22, February 2016.

**백주련**(Juryon Paik)

1997년 : 성균관대학교 정보공학과 (이학사)
2005년 : 성균관대학교 대학원 (공학석사)
2008년 : 성균관대학교 대학원 (공학박사-데이터마이닝)

1997년 : 삼성SDS
2009년~2015년: 성균관대학교 박사후연구원, 연구교수
2016년~현 재: 평택대학교 데이터정보학과 조교수
※관심분야 : 데이터마이닝(XML, 트리), 빅데이터(웹, 센서)