질의유형에 기반한 웹 검색의 성능 향상

강 인 호† · 안 동 언††

요 약

인터넷의 발달로 인해 웹에서 얻을 수 있는 정보의 종류와 수는 급진적으로 증가하고 있다. 이에 따라 사용자가 요구하는 정보는 문서뿐만 아니라 사이트 그리고 서비스 단위로 확장되고 있다. 기존의 연구에서 웹 검색을 위해 사용되었던 정보들과 이들의 일률적인 결합형태는 다양한 사용자의 요구를 만족시키기 어렵다. 보다 좋은 결과를 얻기 위해서는 검색에 사용하는 정보의 특성을 분석하고, 질의에 따른 알맞은 정보의 사용이 필요하다. 본 연구에서는 사용자 질의유형에 따른 정보들의 유용성을 살펴보고 적절한 사용법을 분석한다. 그리고 차츰 대두되고 있는 서비스 검색을 위한 서비스 링크정보를 제안한다.

Improving the Performance of Web Search using Query Types

In-Ho Kang† · Dong-Un An††

ABSTRACT

The Web is rich with various sources of information. Due to the massive and heterogeneous web document collections, users want to find various types of target pages. Each type of information for Web search has designated queries. If a user query is not a designated query, then we cannot have good result documents. Different strategies are needed to utilize the goodness of each type of information for a search engine. If we know the property of information, then we can refine candidate pages and rank them delicately. Various experiments are conducted to show the properties of each type of information. Therefore, we show an appropriate combining formula to utilize the properties of each type of information. In addition, for a service finding task, we propose Service Link Information that utilizes the existence of mechanisms for a user interaction.

키워드 : 웹 검색(Web Search), 사이트 검색(Site Search), 서비스 검색(Service Serach), 질의유형(Query Type), 서비스 링크정보(Service Link Information), 정보검색(Information Search)

1. 서 론

웹 검색기는 웹을 이용하여 작업을 수행하고자 하는 사람에게 없어서는 안 될 중요한 도구이다. 웹 검색기는 사용자가 원하는 정보를 바로 찾아내는 일뿐만 아니라, 관련된 문서를 찾아서 사용자가 알지 못했던 개념을 구체화시켜 나가거나 혹은 의도되지 않은 호기심을 충족시켜서 새로운 작업을 유도한다. 최근 급진적으로 증가하고 있는 웹 환경에서 정보의 종류와 수는 사용자에게 다양한 형태의 작업을 가능하게 한다. 기존의 단순 문서 읽기에서 사이트를 방문하여 음악을 듣고, 물건을 구입하고, 공연을 예매하는 등 다양한 활용법이 나타나고 있다. 웹 검색 또한 이러한 사용자의 다양한 요구를 만족시킬 수 있어야 한다.

기존의 웹 검색엔진은 사용자의 작업 의도를 고려하지 않은 채, 여러 경우에서 좋은 성능을 보이는 정보를 찾는데 주력했다[1]. 이를 위해 링크정보나 URL 정보 등이 개발되어 문서 순위화의 정보로 사용되고 있다[2]. 하지만 TREC (http://www.nist.gov)과 같은 연구에서는 실제 검색엔진으로써 성공을 거둔 구글 검색기(http://www.google.com)가 사용하는 페이지랭크(PageRank)를 이용하여 좋은 성능을 얻지 못했다[3-5]. 이는 실제 사용자가 제시하는 질의를 무시한 채 내용 기반 문서검색 위주의 질의 형태만을 고려하였기 때문이다. 검색에 사용되는 정보는 목표로 하고 있는 결과 문서 집합이 존재하며 대상으로 하는 여러 형태의 질의들이 있다. 보다 나은 웹 검색을 위해서는 해당 정보들의 대상 결과물과 대상 질의의 특성을 분석할 필요가 있다.

"*How is a tornado formed?*" 혹은 "*tornado formed*"라는 사용자 질의에 대해서 페이지랭크와 같은 상대적으로 유명한 문서를 강조하는 링크정보를 사용할 경우 *tornado*에 대한 설명 대신, *tornado*로 유명한 마을에 대한 홈페이지가 상위 결과로 제시된다. 반면 "*Where is the Sony Corporation?*"

※ 본 연구는 한국과학재단 목적기초연구(R01-2003-000-11588-0) 지원으로 수행되었음.
† 정 회 원 : 삼성종합기술원 Computing LAB 전문연구원
†† 종신회원 : 전북대학교 전자정보공학부 부교수
논문접수 : 2004년 5월 18일, 심사완료 : 2004년 6월 15일

혹은 "*Sony Corporation*"의 경우 단어 매칭 정보를 강조한다면, *Sony* 사이트 입구 페이지 대신 *Sony*라는 단어가 많이 나타나는 회사 전화번호부나 *Sony*사의 경영정보 관련 문서가 상위결과로 제시된다. 그리고 "*download game*"의 경우, 단어 매칭 정도를 강조하거나 링크정보를 강조할 경우, 사용자가 다운로드 받을 수 있는 게임을 가진 문서 대신, 게임명만 텍스트 형태로 있는 문서를 결과로 제시한다. 이와 같이 특정 상황에서 좋은 성능을 보이는 정보라 하더라도 모든 상황에서 좋은 성능을 보이는 것은 아니다.

웹 환경에서 사용자의 정보 요구를 나누어 보면 크게 세 가지로 볼 수 있다[6]. 첫째로 원하는 정보를 설명하는 혹은 정보와 관련된 문서를 찾는 내용 검색, 둘째로 사용자가 관심 있는 개인이나 단체의 사이트 입구를 찾는 사이트 검색, 셋째로 사용자가 이용하고자 하는 서비스를 제공하는 웹 페이지를 찾는 서비스 검색을 들 수 있다. 기존의 웹 검색은 내용 검색과 사이트 검색에 좋은 성능을 보이고 있다. 그러나 차츰 중요성이 커지고 있는 서비스 검색에 대해서는 좋은 성능을 보이고 있지 못하다. 본 연구에서는 서비스 검색을 위해서 문서가 가지고 있는 서비스 매개장치를 표현하는 서비스 링크정보를 제안한다.

본 연구에서는 세 가지 질의유형에 대해서, 기존연구에 제시되었던 단어 매칭 위주의 내용정보 그리고 페이지랭크로 대표되는 링크정보 그리고 URL 정보의 특성을 보인다. 그리고 각 질의유형에 따른 문서 순위화 알고리즘의 특성을 보인다.

2. 관련 연구

웹 문서검색에서 순위화를 위하여 사용되는 정보로 크게 내용정보, 링크정보 그리고 URL 정보를 들 수 있다. 내용정보가 특정 단어에 대한 연관도를 나타내는 정보라면 링크정보와 URL 정보는 문서가 가지는 정보이다.

2.1 내용 정보

질의와 연관되는 정도를 알아보기 위해서, 질의로 사용된 단어의 분포와 빈도 그리고 다른 문서들에서 해당 단어의 분포와 빈도를 이용한다. 이러한 단어는 발생 위치에 따라서 제목, 본문, 그리고 anchor text로 나눌 수 있다[1, 7]. Anchor text는 하이퍼링크로 연결된 문서에 대해서 간략한 설명을 가진다. 때때로 이는 연결된 문서에 나타나지 않는 단어와 설명을 가지고 있다. Anchor text를 이용할 경우 하이퍼링크로 연결된 이미지 파일이나 아직 웹 로봇이 가지고 오지 못한 문서에 대해서도 검색이 가능하다.

단어의 관련도는 *tf*와 *df*로 표현된다. *tf*는 웹 문서에서 특정 단어의 출현 빈도를 나타낸다. 이는 특정 단어가 얼마나 문서의 내용을 잘 표현하고 있는지를 나타낸다. *df*는 특정 단어가 출현한 문서의 수를 나타낸다. 이는 특정 단어가 가지는 변별성 정도를 나타낸다. *df*가 큰 단어는 관련 문서와 비관련 문서를 구분하는데 별로 유용하지 못한 반면 *df*가 작은 단어는 구분하는데 유용함을 뜻한다[8, 9]. *tf*와 *df*를 이용하여 문서의 관련도를 계산하는 기법엔 여러 가지가 있다. 주로 TF-IDF 방식으로 사용되는데, *tf*와 *df*의 역수를 곱하는 형태로 각 단어의 중요도와 유사도를 계산한다. TF-IDF 방식 외에 2-Poisson 모델에서 기원한 OKAPI 방식이 많이 사용된다. 색인어 t에 대한 가중치 w_t는 다음과 같이 계산한다[10].

$$w_t = q_t \times tf \times \frac{\log\left(\frac{N - df + 0.5}{df + 0.5}\right)}{0.5 + 1.5 \times \frac{doc\_length}{avg\_doc\_length} + tf}$$

여기에서 q_t는 질의에서 t의 가중치를 나타내며, N은 전체 문서의 수를 나타낸다. *doc_length*는 t가 포함된 문서의 길이를 나타내며 *avg_doc_length*는 문서의 평균 길이를 나타낸다.

언어모델에 기반한 색인어 가중치 기법도 많이 사용되는데 Kullback-Leibler Divergence 방식은 질의를 위한 언어모델과 검색용 문서를 위한 언어모델을 만들어 두 모델의 거리를 계산하는 방식이다[11].

$$D(\theta_Q \| \theta_D) = -\sum_t p(t \mid \theta_Q) \log \frac{p(t \mid \theta_Q)}{p(t \mid \theta_D)}$$

θ_Q는 질의를 위한 언어모델을 나타내며, θ_D는 색인 문서를 위한 언어모델을 나타낸다.

2.2 링크 정보

웹 문서들은 서로 링크로 연결되어 있다는 점에서 일반적인 문서와는 다른 구조적 특징을 가진다[12]. 웹 문서들의 상대적인 중요성을 측정하기 위해 제안된 것이 페이지랭크이다. 페이지랭크는 질의와 무관한 정보이지만, 색인 작업을 할 때 미리 계산할 수 있어 빠른 검색이 가능하다. 따라서 페이지랭크는 상업적인 정보 검색기에 많이 사용된다. 페이지랭크는 링크를 통해 연결되어 있는 웹 문서들을 그래프 구조로 생각하고 웹 문서들의 순위를 계산한 것이다. 많은 문서들이 가리키고 있는 문서가 더 중요하다고 생각하고 그것에 대한 값을 수치로 표현한다. 페이지랭크를 계산하는 수식은 다음과 같다[2].

$$PR(A) = (1-d) + d \times \left(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)} \right)$$

$PR(A)$는 문서 A의 페이지랭크, $PR(T_i)$는 문서 A를 가리키는 문서 T_i의 페이지랭크, $C(T_i)$는 문서 T_i가 가리키는

문서 개수, d는 사용자가 특정 문서에서 만족하지 못하고 다른 문서로 이동할 확률을 나타낸다. 전체 웹 문서들의 페이지랭크를 구하기 위해 초기에 시작값을 설정하고, 반복적으로 페이지랭크를 계산하는 방식을 적용한다. 웹 문서의 페이지랭크는 반복 계산 과정을 통해 초기값에서 원래 페이지랭크의 근사치로 수렴한다[13].

2.3 URL 정보

동일한 사이트에서 중요한 문서는 상위 디렉토리에 위치할 가능성이 크다. 또한 사이트명이나 웹 문서가 가지는 정보와 유사한 형태로 URL이 기술될 가능성이 크다. 이러한 URL 정보를 이용하는 가장 쉬운 방법은 URL의 깊이를 보거나 사용자 질의와 URL을 일치시켜 보는 것이다. URL의 깊이를 보기 위해서는 URL에 나타난 '/' 개수를 이용한다. Westerveld(2001)는 URL의 깊이를 4가지 형태로 나누어 설명했다[14].

- root : 호스트명만 나타나는 형태
  (예 : http://trec.nist.gov)
- subroot : 호스트명에 하나의 디렉토리가 나타나는 형태
  (예 : http://trec.nist.gov/pubs)
- path : 호스트명에 임의 길이의 디렉토리가 나타나는 형태(예 : http://trec.nist.gov/pubs/t9/paper)
- file : index.html외의 파일 이름으로 끝나는 형태
  (예 : http://trec.nist.gov/ps/t9/t9.html)

Westerveld(2001)는 각 URL 형태가 사이트 출입 페이지(entry page)일 확률(*URLprior*)을 계산하였다. 각 URL 유형에 따른 *URLprior*는 다음과 같이 계산한다.

$$URLprior = P(entrypage \mid URLtype = t)$$

본 연구에서는 질의유형에 따라서 내용정보, 링크정보 그리고 URL 정보의 특성을 보인다. 그리고 보다 나은 성능을 위해서 각 질의유형에 따른 정보들의 결합 방식을 제시한다.

2.4 웹 검색에서의 질의유형

사용자 질의는 의도에 따라서 세 가지로 구분된다[6].

- 내용 검색(informational need)
- 사이트 검색(navigational need)
- 서비스 검색(transactional need)

내용 검색은 사용자가 알고자 하는 정보와 관련이 있는 순서대로 순위화 된다. 예를 들어 "*What is a prime factor?*" 혹은 "*prime factor*"와 같은 질의는 사용자가 '*prime factor*'에 대해서 알기를 원한다. 반면 사이트 검색은 사용자가 찾고자 하는 사이트 순서대로 순위화 된다. 예를 들어 "*Where is the site of John Hopkins Medical Institutions?*"나 "*John Hopkins Medical Institutions*"와 같은 질의는 사용자가 "*John Hopkins Medical Institutions*" 사이트를 방문하기를 바란다. 사이트 검색에서는 비록 그 사이트와 관련된 문서일지라도 사이트의 중심 출입 문서만을 정답으로 채택한다. 그리고 서비스 검색은 사용자가 찾고자 하는 서비스를 제공하는 순서대로 순위화 된다. 예를 들어 "*Where can I buy concert tickets?*"나 "*buy concert tickets*"와 같은 질의는 사용자가 콘서트 티켓을 구매할 수 있는 웹 문서를 찾기를 바란다. 이와 같이 검색 대상과 사용자의 의도가 다양해짐에 따라서 이에 따른 문서를 순위화 시키는 정책이 달라져야 한다.

3. 서비스 링크정보를 이용한 서비스 검색

서비스 검색에서 좋은 정답 문서는 사용자에게 원하는 서비스를 제공할 수 있어야 한다. 이러한 서비스를 제공하기 위해서는 사용자와의 일련의 작업이 필요하다. 이러한 작업은 물건을 구입하거나, 음악 파일을 다운로드 하거나 영화 티켓을 예매하기 위해서 사용자가 정보를 입력하거나 구입 버튼을 누르는 과정을 포함한다. 예를 들어 사용자가 "*winamp download*"라는 질의를 하는 경우에는 사용자가 웹 문서의 하이퍼링크를 클릭하여 *winamp* 프로그램을 다운로드 받으려는 의도를 가지고 있는 것이다. 따라서 정답 문서에는 다운로드를 받을 수 있는 버튼이나 링크가 존재해야 한다. 이와 같이 서비스를 제공하는 페이지에는 CGI program이나 하이퍼링크의 작동 장치를 포함한다. 이러한 작동 장치의 존재 여부를 이용한다면 서비스 검색의 정답 문서로의 유용성을 알 수 있다. 이러한 작동 장치는 다음과 같이 하이퍼링크의 URL을 이용하여 구별할 수 있다.

- site : 호스트 명이나 index.html로 끝나는 형태
  (예 : http://cs.kaist.ac.kr/index.html)
- service: CGI 관련 프로그램으로 끝나는 형태
  (예: http://www.google.com/search?q=IR)
- media : 실행 파일이나 멀티미디어 관련파일로 끝나는 형태(예 : http://download.nullsoft.com/winamp/client/winamp3_0-full.exe)
- html : index.html 이외의 html로 끝나는 형태
  (예 : http://www.kaist.ac.kr/naat/interdept.html)

URL만으로는 쉽게 분류할 수 없는 것 중에JavaScript가 있다. 이러한 경우, 널리 알려진 forward, backward와 같은 history와 관련된 경우에는 site 유형으로 간주하여 할당한다. 그 외의 경우는 service 유형으로 할당한다.

서비스 검색의 결과로서 유용한 장치를 가졌는지를 알려주는 서비스 링크정보를 다음과 같이 정의한다.

$$Link(d) = \frac{\#service\_links}{\#service\_links + 0.5 + 1.5 \times \frac{link\_count(d)}{avg\_link\_count}}$$

#service_links는 service_link와 media_link의 개수를 합친 것을 나타낸다. 그리고 link_count(d)는 문서 d에 나타난 모든 하이퍼링크의 개수를 의미한다. 한 문서에 나타나는 하이퍼링크는 중복하여 계산하지는 않는다. 따라서 동일한 하이퍼링크가 여러 번 나타나도 한 번으로 취급한다.

$$Link\_count(d) = \#site\_links + \#html\_links + service\_links + \#media\_links$$

서비스 링크정보 $Link(d)$는 link_count에 의해서 정규화된다. 즉 하이퍼링크가 얼마나 많이 존재하는 문서에서 나왔는지에 따라서 정규화된다. 이러한 서비스 링크정보는 기존 정보에 선형적으로 결합할 수 있다.

$$rel_{new}(d) = \alpha \times rel_{old}(d) + \beta \times Link(d)$$

4. 질의유형별 웹 검색기의 특징 분석

본 장에서는 기존의 검색엔진에서 사용된 여러 정보들의 특성에 대해서 알아본다.

4.1 분석용 데이터

본 연구에서는 검색에 사용되는 정보들의 특성을 살펴보기 위해서 WT10g 데이터 컬렉션을 사용한다. WT10g는 10G의 웹 문서 집합으로써 CSIRO에서 배포한다[15]. 질의로는 TREC-2001 Web Track의 adhoc task 질의 세트(501~550)와 homepage finding task 질의 세트(1~145)를 이용하였다. Adhoc task 질의 세트는 내용 검색을 위해 사용하고 homepage finding task 질의 세트는 사이트 검색을 위해 사용하였다. 서비스 검색을 위해서 웹 검색기 Lycos의 질의 로그 파일 중 서비스 검색 관련 질의 100개를 추출하였다. 이 중 50개는 학습에 사용하고 나머지 50개는 실험에 사용하였다. 사용된 질의는 stemmer를 통해 활용 형태를 복원하며 stopwords를 이용하여 불용어를 제거하였다.

내용 검색의 성능을 측정하기 위해 평균 정확도(average precision)를 사용한다[8, 9]. 그리고 사이트 검색의 평가를 위해서는 MRR(Mean Reciprocal Rank)을 사용한다[4]. MRR은 검색엔진이 첫번째 정답 문서를 몇 위로 제시하였는가를 측정한다. n개의 질의에 대해서 MRR을 측정할 때는 각 질의의 결과에 대해 첫번째 정답 문서의 순위 r의 역수를 구한 후 이 값을 평균한다.

$$MRR = \frac{1}{n}\sum_i \frac{1}{r_i}$$

서비스 검색을 평가하기 위해서는 상위 n개 문서에서 정답 문서의 포함율을 이용하여 계산하며 P_n으로 표현한다. n의 값으로는 주로 1, 5, 10을 사용한다. 정답 문서는 사용자가 원하는 정보를 바로 제공할 수 있어야 한다. 추가의 탐색이 필요하거나 프로그램의 버전이 일치하지 않을 경우 정답으로 간주하지 않는다.

4.2 검색 정보에 대한 특징 분석

다양한 정보의 사용 예를 고려하기 위해서 정보의 출처, 결합 그리고 질의와의 연산자를 고려하였다. 정보 출처 면에서는 먼저 색인어의 출현 위치를 크게 두 가지 경우로 나누어 생각한다. 제목이나 anchor text에 나타나는 경우와 문서 본문에 나타나는 경우로 나눈다. *Anchor* 방식은 문서 본문을 제외한 제목과 해당 문서를 연결하고 있는 다른 문서의 anchor text 만을 가지고 색인을 수행한다. 반면 *Common* 방식은 해당 문서의 전 부분에 대해서 색인을 수행한다. 정보 결합 측면을 고려하기 위해서, *tf*와 *df*만을 이용한 내용정보 방식과 페이지랭크와 *URLprior* 정보를 내용정보에 결합(*CMB*)하는 두 가지 방식을 고려한다. 내용정보(Content Information)와 페이지랭크 *URLprior* 정보를 결합하는 수식은 다음과 같다.

$$rel(d) = 0.65 \times Content\ Information(d) + 0.25 \times URLprior(d) + 0.1 \times PageRank(d)$$

그리고 연산자에 대해서는 '*and*' 연산자와 '*sum*' 연산자를 고려한다. 'and' 연산자는 질의에 사용된 모든 단어를 가지고 있는 문서만이 정답으로 채택된다. 반면 'sum' 연산자는 질의에 사용된 단어 중 하나라도 가지고 있는 문서를 정답으로 채택한다.

<표 1>은 위의 각 경우에 대한 내용기반 검색과 사이트 검색에 대한 성능을 나타낸다. 예를 들어 *Anchor and CMB* 방식은 제목과 anchor text 만을 이용해서 색인을 수행했으며, 'and' 연산자를 이용하여 질의와 부합 정도를 비교한다. 그리고 내용정보 외에 페이지랭크와 *URLprior* 정보를 결합하여 사용한 경우를 나타낸다. <표 1>에서 *MAX*와 *AVG*는 TREC-2001에 제출된 시스템들 중 최고 성능과 평균 성능을 뜻한다.

<표 1>을 통한 내용 검색에서는 *Common* 방식이 *Anchor* 방식에 비해 월등히 나은 성능을 보이고 있는 반면, 사이트 검색에서는 *Common* 방식과 *Anchor* 방식이 큰 차이를 보이지 않고 있음을 알 수 있다. 제목과 anchor text로는 내용 검색에 필요한 충분한 정보를 제공하지 못한다는 것을 알 수 있다. 그리고 사이트 검색에서는 페이지랭크와 *URLprior*를 결합하여 보다 나은 성능을 얻을 수 있는 반면, 내용 기반 검색에서는 오히려 성능이 감소함을 알 수 있다.

〈표 1〉 정보에 따른 내용 검색과 사이트 검색의 성능 비교

	내용 검색	사이트검색
Model	P_{avg}	*MRR*
Anchor and	0.031	0.297
Anchor and CMB	0.031	0.431
Anchor sum	0.034	0.351
Anchor sum CMB	0.034	0.583
Common and	0.131	0.294
Common and CMB	0.122	0.580
Common sum	**0.182**	0.355
Common sum CMB	0.169	**0.673**
MAX	0.226	0.774
AVG	0.145	0.432

내용 검색에서는 사용자가 알고자 하는 개념의 중요 단어나 설명하는 단어가 질의로 사용된다. 그러나 사용자가 제시하는 설명 형태와 동일한 형태로 문서에 나타난다고 보장할 수 없다. 따라서 내용 기반 검색에 있어서 *'and'* 연산자로는 좋은 성능을 얻을 수 없다. 사이트 검색에 있어서도 *'and'* 연산자가 *'sum'* 연산자보다 성능이 다소 떨어짐을 알 수 있다. 이는 *'and'* 연산자를 이용하여 결과를 제시하지 못하는 경우가 있기 때문이다. 그러나 결과를 얻을 수 있는 경우에는 좋은 성능을 보이고 있다. 결과가 없는 경우를 보완하기 위해 사이트 검색에서 최고 성능을 보이는 *Common sum CMB*와 *Anchor and CMB*를 결합하여 정렬할 경우, 0.73의 성능을 얻을 수 있었다. 이는 *Common sum CMB* 보다도 나은 성능이다. 마찬가지로 내용 기반 검색에서 최고 성능인 *Common sum*과 *Anchor and*를 결합했을 경우에는 0.173으로 성능이 떨어졌다. 이를 통해서 사이트 검색에 있어서는 *'and'* 연산자가 유용함을 알 수 있다.

서비스 검색 또한 WT10g 문서 집합을 대상으로 동일한 정보와 결합방식으로 분석을 수행했다. <표 2>는 각 경우에 대한 서비스 검색의 성능을 나타낸다. 상위 문서 10개에 대해서 정답 문서의 포함율을 계산하였다. 실험에 사용한 Lycos 질의로그와 WT10g가 동시대에 작성된 것이 아니라서 좋은 성능을 얻을 수 없었다. 예를 들어 mp3에 관련한 결과물이 WT10g에는 존재하지 않았다. 50개의 질의 중, 16개에 대해서는 결과를 얻을 수 없었다. *Anchor* 표현이 *Common*에 비해 좋은 성능을 나타낸다. 사이트 검색과 같이 서비스와 관련된 정보가 Anchor text와 title에 나타남을 알 수 있다. 결과에서 Anchor sum이 가장 좋은 성능을 얻었다. 이는 사이트 검색과 달리 페이지랭크와 URL정보를 추가하기 때문에 성능이 낮아지는 결과를 보인다. 이를 통해서 서비스 검색 또한 내용 검색과 다른 정보의 결합 형태가 필요함을 보인다.

〈표 2〉 정보 사용에 따른 서비스 검색의 성능

Model	P_{10}
Anchor and	0.05
Anchor and CMB	0.05
Anchor sum	0.07
Anchor sum CMB	0.06
Common and	0.05
Common and CMB	0.03
Common sum	0.05
Common sum CMB	0.03

<표 3>은 본 연구에서 제안하는 서비스 링크정보를 결합했을 때의 성능 향상율을 보인다. 서비스 링크정보를 결합함으로 해서 성능이 향상됨을 알 수 있다. 따라서 본 연구에서 제안하는 서비스 링크정보가 서비스 검색에 유용함을 알 수 있다.

〈표 3〉 서비스 링크정보 사용에 따른 성능 변화

Model	*Imp. of* P_{10}
Anchor and Service	8.7%
Anchor sum Service	23.5%
Common and Service	78.3%
Common sum Service	92.0%

<표 4>는 서비스 링크정보를 구글 검색기 결과물에 적용했을 때의 결과를 보인다.

〈표 4〉 서비스 링크정보 적용에 따른 구글 검색기의 성능 변화

Model	*Train*			*Test*		
	P_1	P_5	P_{10}	P_1	P_5	P_{10}
Google	0.28	0.25	0.34	0.38	0.31	0.35
Google + Service	0.38	0.33	0.39	0.48	0.34	0.38

동일 질의에 대해서 구글 검색기를 이용해 100개의 상위 결과 문서를 얻고 이 문서에 대해서 서비스 링크정보를 추가하였다. 구글 검색기에서 각 문서에 대한 질의와의 유사도를 제공하지 않기 때문에 순위에 기반하여 유사도를 추정하였다. 본 연구에서는 아래와 같은 형태로 각 문서에 서비스 링크정보를 추가하여 재순위하였다.

$$rel_{new}(d) = \frac{1}{e^{r(d)}} + 0.9 \times Link(d)$$

$r(d)$는 문서 d의 순위를 나타낸다. 0.9는 학습용 질의셋을 이용하여 계산했다. 정확한 유사도를 알지 못하는 이유와 구글이 동일 사이트 결과를 연달아 출력하는 형태로 많은 향상을 얻을 수는 없었지만 성능 향상이 있음을 알 수 있다. 따라서 본 연구에서 제안하는 서비스 링크정보가 상용 검색기에도 적용될 수 있는 좋은 정보임을 알 수 있다.

4.3 문서 순위화 알고리즘에 대한 특징 분석

검색에 사용되는 정보의 종류에 더하여, 문서 순위화하는 알고리즘에 따른 웹 검색의 변화를 살펴본다. 본 연구에서는 다양한 문서 순위화 알고리즘을 적용시켜보기 위해서 Lemur라는 툴킷을 사용한다[16]. Lemur라는 툴킷은 TF-IDF와 OKAPI 방식 외에도 Kullback-Leibler 방식과 같은 언어 모델 방식과 Feedback 방식[17] 그리고 다양한 Smoothing 방식을 제공한다. 본 연구에서는 OKAPI, TF-IDF 그리고 Kullback-Leibler에 Dirichlet smoothing(KL-DIR)을 적용한 세 가지 방식을 비교한다. 그리고 앞에서 살펴본 정보별 특징에 기반하여 세 가지 질의유형을 위한 세 가지 검색 모델을 만든다. *Lemur* 툴킷은 내용 검색을 위해서 사용하고, 이 *Lemur* 툴킷에 페이지랭크와 URL 정보를 추가하여 *H-Lemur*를 만든다. 그리고 *Lemur* 툴킷에 서비스 링크정보를 추가하여 *SLemur*를 만든다. 각각의 모델은 *Lemur* 툴킷을 이용하여 1,000개의 상위 문서를 얻은 뒤, 추가 정보를 결합하여 재순위화하여 결과를 제시한다.

$$
\begin{aligned}
Lemur &= Content\ Information \\
HLemur &= 0.65 \times Content\ Information \\
&\quad + 0.25 \times URLprior + 0.1 \times PageRank \\
SLemur &= Content\ Information + 0.9 \\
&\quad \times Sevice\ Link\ Information
\end{aligned}
$$

<표 5>, <표 6>, <표 7>은 각 질의유형에 대한 OKAPI, TF-IDF, KL_DIR의 성능을 보인다. <표 5>는 내용 검색, <표 6>은 사이트 검색 그리고 <표 7>은 서비스 검색에서의 성능을 보인다. 문서의 모든 위치에서 색인어를 추출하였으며 대상 문서 집합으로 WT10g를 이용하였다.

〈표 5〉 문서 순위화에 따른 내용 검색의 성능 변화

Model	OKAPI	TF-IDF	KL-DIR
Lemur	0.182	0.170	0.210
HLemur	0.157	0.145	0.187
SLemur	0.154	0.141	0.179

〈표 6〉 문서 순위화에 따른 사이트 검색의 성능 변화

Model	OKAPI	TF-IDF	KL-DIR
Lemur	0.355	0.340	0.181
HLemur	0.691	0.664	0.558
SLemur	0.278	0.256	0.170

〈표 7〉 문서 순위화에 따른 서비스 검색의 성능 변화

model	OKAPI	TF-IDF	KL-DIR
Lemur	0.050	0.068	0.034
HLemur	0.028	0.036	0.030
SLemur	0.096	0.094	0.068

예상했던 대로 *Lemur* 툴킷은 내용 검색에서, *HLemur* 툴킷은 사이트 검색에서 그리고 *SLemur* 툴킷은 서비스 검색에서 제일 좋은 성능을 보임을 알 수 있다. 언어모델 방식이 내용 검색에서 좋은 성능을 보이며, OKAPI 모델이 사이트 검색에서 좋은 성능을 보임을 알 수 있다. 그리고 TF-IDF 방식이 서비스 검색에서 좋은 성능을 보임을 알 수 있다. 이와 같이 사용자 질의유형에 따라서 문서 순위화 알고리즘 또한 다른 영향을 미침을 알 수 있다. 이에 따라 보다 나은 성능을 얻기 위해서는 사용자 질의유형에 따라서 웹 검색에 사용하는 정보가 달라져야 하며, 또한 웹 문서 순위화 알고리즘 또한 달라져야 함을 알 수 있다. 이는 각 순위화 방식에서 색인어에 가중치를 할당할 때, 색인어가 가중치로 가질 수 있는 최고값과 *tf*에 따른 가중치의 상대적인 차이를 주는 정도에 따라 결과가 달라진 것으로 보인다. OKAPI 방식이 세 가지 유형에서 다른 두 순위화 방법에 비해 고른 성능을 보임을 알 수 있다.

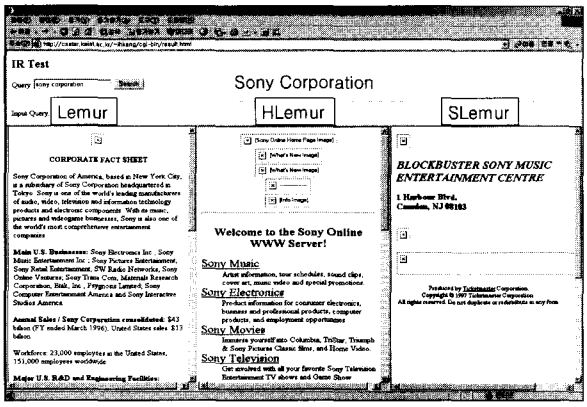
4.4 토 의

(그림 1)은 사이트 검색어인 "*Sony Corporation*"을 질의로 WT10g에서 얻어낸 1위 문서를 나타낸다. 기존의 내용 정보 기반으로 검색을 수행할 경우에는 Sony 사의 경영정보가 상위결과로 나타남을 알 수 있다. 반면 구글과 같이 페이지랭크와 링크정보를 추가할 경우, Sony의 웹 사이트를 상위결과로 얻을 수 있었다. 또한 본 연구에서 제안하는 서비스 링크정보를 추가할 경우, Sony에서 판매하는 Video와 DVD를 구매할 수 있는 페이지가 상위결과로 얻어진다. 이는 Video와 DVD 상품을 구매하는 버튼에 기인한다. (그림 2)는 서비스 검색인 "*download game*"을 질의로 WT10g에서 얻어낸 결과를 나타낸다. 기존의 내용정보 기반 방식과 페이지랭크와 링크정보를 추가한 방식이 게임 리스트를 나열한 페이지를 상위결과로 제시하는 것에 반해, 서비스 링크정보를 사용할 경우, 게임 리스트와 함께, 사용자가 다운로드 받을 수 있는 페이지가 상위결과로 나타난다.

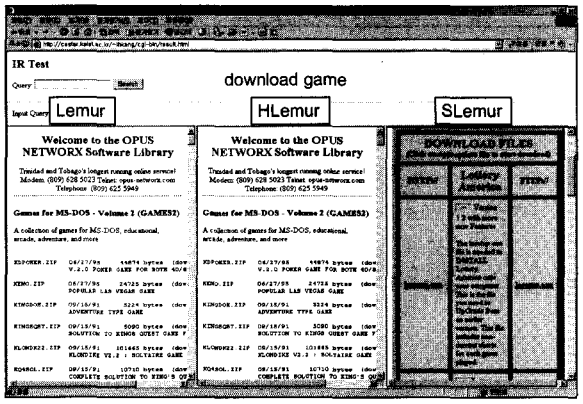
최근의 웹 검색기는 사용자에게 검색 영역과 방법을 선택할 수 있게 한다. 혹은 각 종류별로 검색을 수행하고 그 결과를 합쳐서 나타낸다. 사용자의 요구 형태가 알려진 영역이나 사용자가 요구 사항에 대해서 정보를 제공할 경우, 본 연구에서 분석된 결과를 이용한다면 보다 정확한 결과를 제공할 수 있다. 또한 본 연구에서 제안하는 결과는 특정 질의유형을 위해, 사용하는 정보의 종류와 결합 방식의 변화만으로도 질의유형에 맞는 검색 결과를 얻을 수 있음을 보여준다.

본 연구에서는 하이퍼링크를 네 가지로 나누었다. 그러나 media의 경우 확장자에 따라서 더욱 세분화할 수 있다. *mp3*나 *wav*와 같은 음악 파일, *pdf*와 *hwp* 같은 문서 파일, 그리고 *jpg*나 *gif* 같은 그림 파일로 나눌 수 있다. 서비스

링크정보를 더욱 세분화된 하이퍼링크 유형에 맞추어 정의할 경우, 사용자의 요구 유형을 더욱 세분화하여 대응할 수 있다.



(그림 1) 질의 "Sony Corporation"에 대한 각 유형별 검색 결과



(그림 2) 질의 "download game"에 대한 각 유형별 검색 결과

5. 결 론

인터넷의 발달로 인해 웹에서 얻을 수 있는 정보의 종류와 수는 급진적으로 증가하고 있다. 아울러 사용자의 요구 또한 다양해지고 있다. 이러한 사용자의 요구는 내용 검색, 사이트 검색, 그리고 서비스 검색으로 나눌 수 있다. 내용 검색은 원하는 정보를 설명하는 혹은 정보와 관련된 문서를 찾는 것이며, 사이트 검색은 사용자가 관심 있어 하는 개인이나 단체의 사이트 입구를 찾는 것이고, 서비스 검색은 사용자가 관심 있어 하는 서비스를 제공하는 웹 페이지를 찾는 것을 말한다. 질의유형에 따라서 검색에 사용되어야 할 정보는 차이를 보인다. 내용 기반 검색을 위해서는 문서 내부의 정보가 유용한 반면, 사이트 검색을 위해서는 사이트 내에서 문서의 상대적인 위치에 따른 정보가 유용하다. 마지막으로 서비스 검색을 위해서는 사용자와의 작업을 구현하는 장치의 빈도 정보가 유용하다. 문서 순위화 알고리즘 또한 질의유형에 따라서 차이를 보인다. 내용 검색에서는 KL-DIR 방식이 좋은 성능을 보이며, 사이트 검색에서 OKAPI 그리고 서비스 검색에서는 TF-IDF 방식이 좋은 결과를 보인다. OKAPI를 기본으로 문서를 추출하고 각 정보를 결합하여 문서를 순위화할 경우, 세 가지 유형에 있어서 좋은 성능을 얻을 수 있었다.

본 연구에서 제안하는 방법을 적용하기 위해서 사용자 질의유형을 자동으로 구분하는 방법에 대한 연구가 필요하다. 또한 세 가지 질의유형 형태에서 서비스 검색에 대한 더욱 세분화된 유형 구분 방법 또한 연구가 필요하다.

참 고 문 헌

[1] Croft, W. B., "Combining Approaches to Information Retrieval : Recent Research from the Center for Intelligent Information Retrieval," Kluwer Academic Publishers, pp. 1-36, 2000.

[2] Brin, S. and Page, L., "The Anatomy of a Large-scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, Vol.30, No.1-7, pp.107-117, 1998.

[3] Craswell, N., Hawking, D. and Griffiths, K., "Which Search Engine is best at Finding Airline Site Home Pages?," (Tech. Rep.), CSIRO Mathematical and Information Sciences, 2001.

[4] Craswell, N., Hawking, D. and Robertson, S., "Effective Site Finding using Link Anchor Information," In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, pp.250-257, 2001.

[5] Yang, K., "Combining Text and Link-Based Retrieval Methods for Web IR," In Text REtrieval Conference (TREC-10), Gaithersburg, Maryland, pp.609-618, 2001.

[6] Broder, A., "A Taxonomy of Web Search," SIGIR Forum, Vol.36, No.2, 2002.

[7] Ogilvie, P. and Callan, J., "Combining Document Representations for Known-Item Search," In Proceedings of the 26th Annual International ACM SIGIR conference on Research and Development in Information Retrieval, Toronto, Canada, pp.143-150, 2003.

[8] Baeza-Yates, R. and Ribeiro-Neto, B., "Modern Information Retrieval," Essex England : Addison-Wesley Pub Co, 1999.

[9] Salton, G. and McGill, M. J., "Introduction to Modern Information Retrieval," New York : McGraw-Hill, 1983.

[10] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. and Gatford, M., "Okapi at TREC-3," In Text REtrieval Conference (TREC-3), Gaithersburg, Maryland, pp.109-126, 1994.

[11] Zhai, C. and Lafferty, J., "A Study of Smoothing Methods

for Language Models Applied to ad hoc Information Retrieval," In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, pp. 334-342, 2001.

[12] Amento, B., Tervenn, L. and Hill, W., "Does authority mean quality? Predicting expert quality ratings of Web documents," In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 2000.

[13] Page, L., Brin, S., Motwani, R. and Winograd, T., "The PageRank Citation Ranking: Brining Order to the Web," (Tech. Rep.), Stanford Digital Library Technologies Project, 1998.

[14] Westerveld, T., Kraaij, W. and Hiemstra, D., "Retrieving Web pages using content, links, urls and anchors," In Text REtrieval Conference (TREC-10), Gaithersburg, Maryland, pp.663-672, 2001.

[15] Bailey, P., Craswell, N. and Hawking, D., "Engineering a Multi-Purpose test Collection for Web Retrieval Experiments," Information Processing and Management, Vol.39, No.6, pp.853-871, 2003.

[16] Ogilvie, P. and Callan, J., "Experiments using the Lemur Toolkit," In Text REtrieval Conference (TREC-10), http://www-2.cs.cmu.edu/~lemur, Gaithersburg, Maryland, pp. 103-108, 2001.

[17] Harman, D., "Relevance Feedback and Other Query Modification Techniques," In W. B. Frakes & R. Baeza-Yates (Eds.), Information Retrieval Data Structures & Algorithms, Englewood Cliffs, New Jersey : Prentice Hall, pp. 241-263, 1992.

강 인 호

e-mail : inho97.kang@samsung.com

1997년 경북대학교 컴퓨터공학과(공학사)

1999년 KAIST 전산학과(공학석사)

2004년 KAIST 전산학과(공학박사)

2004년~현재 삼성종합기술원 Computing LAB 전문연구원

관심분야 : 정보검색, 정보추출, 한국어 정보처리

안 동 언

e-mail : duan@moak.chonbuk.ac.kr

1981년 한양대학교 전자공학과(공학사)

1987년 KAIST 전산학과(공학석사)

1995년 KAIST 전산학과(공학박사)

1995년~현재 전북대학교 전자정보공학부 부교수

2001년~2002년 전북대학교 정보검색시템연구센터 센터장

관심분야 : 정보검색, 한국어정보처리, 문서분류, 문서요약