커뮤니티 기반 효율적인 웹 검색 시스템 설계

박상관, 박건우, 이상훈
국방대학교 전산정보학과
parksangkwan@naver.com, pgw4050@hotmail.com, hoony@kndu.ac.kr

Design of Efficient Web Search System Based on Community

Sang-Kwan Park, Gun-Woo Park, Sang-Hoon Lee
Dept. Computer Science and Information, Korea National Defense University

요 약

웹 상에 존재하는 정보의 량이 방대해 질수록 사용자가 원하는 정보를 찾는데 더 많은 노력이 필요하게 되었다. 따라서 사람들은 인터넷상에서 원하는 정보를 보다 효율적으로 검색하기 위해 많은 검색 알고리즘들을 개발하였다. 하지만 지금까지 개발된 알고리즘들은 웹 검색자들의 검색의도, 즉 관심사를 파악하는데 어려움이 있다. 따라서 검색자들의 의도에 맞는 정보를 보다 정확하고 효율적으로 검색하기에는 많은 제한사항들이 있다. 본 논문에서는 사용자의 검색 질의와 가장 유사한 커뮤니티를 검색하고 검색된 커뮤니티를 기반으로 보다 효율적인 검색 결과를 획득하기 위한 시스템을 제안한다.

1. 서 론

웹 검색 시스템은 웹에서 원하는 정보를 보다 쉽게 찾기위한 도구로서 그 중요성이 점차 부각되고 있다. 최근 들어 웹 검색 시스템에서 사용자의 질의에 대해 정확도가 높은 충분한 양의 검색 결과를 제공하고자 하는 연구들이 활발하게 진행되고 있다. 그 대표적인 연구로는 제한 검색(limit search), 포커스 크롤러(focused crawler), 웹 문서 클러스터링(web document clustering) 등이 있다. 제한 검색은 현재 입력한 검색어의 검색 결과를 줄이고자 할 때 이용하는 검색 방식으로 검색 범위를 특정 사이트 또는 도메인으로 한정시켜 검색 결과를 제공하는 방법이다[1]. 포커스 크롤러는 웹 문서가 가지는 정보들은 URL을 이용하여 문서간에 연결되어 있다는 특성을 이용하여 질의가 주어진 시점에 질의와 관련 있는 웹 페이지들만을 수집하여 결과로 반환하는 방법이다[2][3][4]. 웹 문서 클러스터링은 클러스터를 구하기 위해 많은 양의 사이트들 또는 웹 페이지들을 서로 관련 있는 웹 페이지들끼리 클러스터링하는 방법이다[5].

그러나 위에서 설명한 연구들은 다음과 같은 단점을 가지고 있다. 제한 검색은 검색의 범위를 URL에 의해 명시되는 사이트 또는 도메인들로만 제한할 수 있을 뿐이며, 의미적으로 관련된 사이트들로 제한할 수 없다. 포커스 크롤러는 질의 시점에 웹 페이지들을 수집하기 때문에 질의 처리 시간이 오래 걸린다. 웹 문서 클러스터링은 클러스터를 구하기 위해 많은 양의 사이트들 또는 웹 페이지를 대상으로 복잡한 처리를 수행하므로 공간적 시간적 비용이 크다.

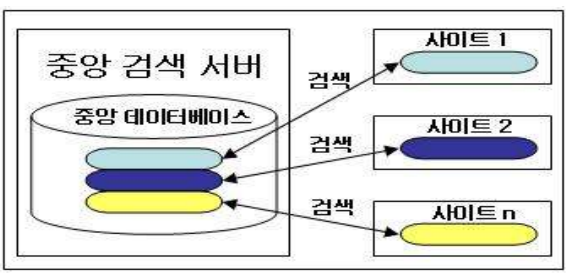
이러한 문제점을 해결하기 위해서 본 논문에서는 정보의 공유를 목적으로 하는 커뮤니티를 대상으로 웹 서비스기술을 적용 함으로써 제한 검색이 가지는 의미적으로 관련된 사이트를 찾지 못하는 문제점과 포커스 크롤러 및 웹 문서 클러스터링이 가지는 질의 처리 시간에 대한 문제를 해결하려 한다. 일반적으로 인터넷 상에서의 온라인 커뮤니티의 정의는 “네티즌들이 직접 정보를 생산, 공유하고 이들이 모여 활동할 수 있는 인터넷 상의 공간”이다[6].

2. 관 련 연 구

2.1 제한 검색(limit search)

사이트 제한 검색은 <그림 1>과 같이 중앙 데이터베이스에 페이지서의 기본 동작 방식은 서비스 기술, 서비스 등록 및 발견, 서비스 간의 통신 관점에서 정의된다.

사이트 제한 검색은 저장된 전체 데이터 중에서 지정된 사이트의 데이터만을 검색하는 기능이다. 제한 검색을 사용하기 위하여 웹 사이트 관리자가 자신의 사이트를 검색 시스템에 등록하면, 웹 로봇[7]이 등록된 사이트에 포함되어 있는 웹 페이지를 수집하여 중앙 데이터베이스에 저장한다. 이때 어떤 사이트로부터 수집된 웹 페이지인지 나타내는 정보를 함께 저장하며, 사이트 제한 검색 요청이 들어오면 이 정보를 사용하여 해당 사이트로부터 수집된 웹 페이지에 대해서만 검색을 수행한다. 사이트 제한 검색 기능을 사용하면 웹 사이트에 검색 엔진을 설치하지 않고도, 마치 웹 사이트에 검색 엔진을 설치하여 운영하는 효과를 볼 수 있다.



<그림 1> 사이트 제한 검색의 개념.

2.2 웹 문서 클러스터링 (web document clustering)

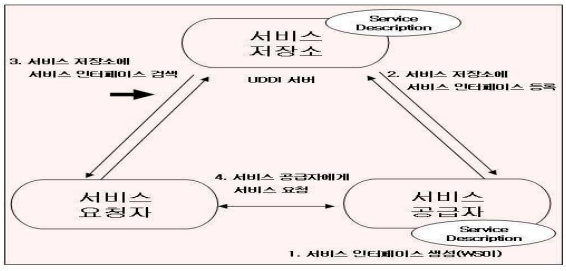
웹 페이지의 클러스터링은 단어 또는 링크 등을 이용하여 웹 페이지 간의 유사도를 측정하는 과정과 측정된 유사도를 바탕으로 기존의 데이터 클러스터링 알고리즘을 적용하는 과정으로 이루어 진다.

Minimum spanning tree(MST)는 클러스터링 알고리즘 중 하나로 MST를 subtree들로 나누어 클러스터를 구하는 방법이다[8]. MST 클러스터링은 클러스터의 개수를 사전에 정하지 않아도 클러스터를 구할 수 있으며 여러 가지 데이터 분포에서도 잘 동작하는 장점이 있다.

2.3 웹 서비스(web services)

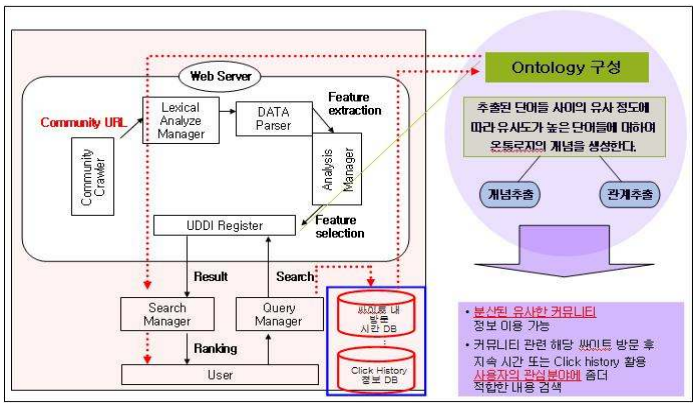
웹 서비스는 웹 상에서 정의된 모듈화 된 소프트웨어 컴포넌트로서, 개방형 표준 데이터 표현 기법인 XML과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해 탄생된 분산 컴퓨팅 기술이다. W3C는 웹 서비스를 URI에 의해 인식되는 소프트웨어 애플리케이션으로서, 인터넷 기반의 프로토콜을 통하여 교환되는 XML기반 메시지를 사용하여 다른 소프트웨어 에이전트들과의 직접적인 상호작용을 지원한다고 정의한다[9].

웹 서비스는 SOAP, WSDL, UDDI를 통해 SOA의 주요 요소인 메시지, 서비스 인터페이스, 서비스 공개 및 발견 체계를 구현한다. 따라서 웹 서비스는 SOA 구축에 필요 한 표준 기술들을 제공한다. <그림 2>에서 볼 수 있듯이 웹 서비스의 기본적인 아키텍처는 SOA를 채택하고 있다. 웹 서비스의 기본 동작 방식은 서비스 기술, 서비스 등록 및 서비스 간의 통신 관점에서 정의 된다.



<그림 2> 웹 서비스 아키텍처

3. 웹 서비스를 이용한 커뮤니티 검색 시스템 구조



<그림 3> 시스템 구조

<그림 3>은 커뮤니티를 기반으로 효율적인 웹 검색을 하기위한 시스템에 대한 구조를 나타낸 것이다. 우선, 커뮤니티들의 특성을 확인하고 사용자들의 의도를 잘 반영하고 있다면 해당 커뮤니티를 웹 검색에 이용한다. 이때 추출된 단어들 사이의 유사도에따른 온톨로지를 구축하고 사용자 별 특정 싸이트에 머문 시간 및 클릭 히스토리 정보를 DB화 하여 검색에 적용함으로써 보다 정확한 검색 결과를 획득하고자 한다. 이에 대한 각각의 모듈 별 처리 과정 및 내용은 다음과 같다.

3.1 RSS Crawler

RSS Crawler는 웹 서비스의 제공자(Provider) 역할을 담당한다. RSS Crawler는 사전에 관리자에 의해 수동으로 등록된 RSS 주소들을 통해 각 커뮤니티에서 문서들을 읽어 들인다. 각 커뮤니티에서 읽어 들인 문서들은 각 커뮤니티의 특징을 추출하는 기초 자료들로 사용된다. RSS Crawler는 수집한 각 문서들을 색인 하기 위해 형태소분석관리자(Lexical Analysis Manager)와 상호작용한다. RSS Crawler는 주기적, 순차적, 병렬적으로 동작한다.

3. 2 형태소분석 관리자

형태소분석 관리자는 RSS Crawler로부터 전달된 각 문서를 형태소 분석하고 분석 결과를 데이터 파서에게 전달하는 역할을 한다. 형태소 분석은 한글형태소분석기(HAM: Hangul Analysis Module)를 활용하며, 이는 기본적인 어휘분석과 함께 불용어 제거, 어근 추출 과정을 한다. 분석된 형태소들은 각 커뮤니티의 특징 벡터를 만들기 위해 데이터 파서에게 전달된다.

가중치 값을 통해 선택된 각 단어들은 다시 벡터 모델로 표현되어 검색을 위해 사용될 수 있도록 UDDI에 저장되는데 그 형태는 다음과 같다.

$$<t_0 : tf_0 : df_0, \cdots, t_i : tf_i : df_i, \cdots, t_n : tf_n : df_n > \quad (1)$$

위와 같은 과정을 통해 최종적으로 뽑힌 최상위 가중치의 값을 가진 단어를 통해 커뮤니티에 있는 정보가 어떤 정보를 담고 있는지를 알 수 있다.

3.3 데이터 파서

데이터 파서(DATA Parser)는 RSS Crawler로부터 읽어들인 각 커뮤니티의 각 문서에 대해 형태소 분석기를 통해 얻어진 각 문서의 형태소 집합을 벡터 모델로 표현한다. 각 문서들을 벡터 모델로 표현 함으로써 해당 커뮤니티의 특징을 결정하기 위한 연산을 비교적 쉽게 할 수 있다. 예를 들면 각 커뮤니티의 특징을 결정하기 위해 각 커뮤니티에서 수집된 문서들에 공통적으로 많이 나타나는 키워드나 단어들을 비교적 쉽게 찾을 수 있다. 데이터 파서는 이러한 연산들을 비교적 쉽게 할 수 있도록 각 문서에 대한 벡터 모델을 다음과 같이 구성한다.

$<t_0 : tf_0,\cdots,t_i:tf_i,\cdots,t_n: tf_n>$ (2)

t_i 는 i번째 단어(term)을 의미하고, tf_i 는 t_i가 발생하는 빈도수를 나타낸다.
벡터모델로 표현된 문서들은 해당 커뮤니티의 특징을 추출 하기 위해 분석 관리자에게 전달된다.

3.4 분석 관리자

분석 관리자는 데이터 파서로부터 전달된 각 문서에 대한 벡터들에 근거해서 해당 커뮤니티의 특징을 결정하는 단어들의 집합을 추출하고 이를 다시 새로운 벡터 모델로 표현하고 검색에 사용할 수 있도록 UDDI에 저장하는 기능을 한다. 분석 관리자는 각 커뮤니티의 특징을 결정 하기 위해 해당 커뮤니티의 모든 문서들에서 가장 일반적으로 발생하는 단어들을 추출한다. 이를 위해 DF(Document Frequency)를 이용하며, 이는 각 문서들이 벡터 형태로 표현 되어 있어 비교적 쉽게 연산된다.

시스템에서는 DF값이 일정 임계값 (이상이 되는 단어에 대해서만 특징값으로 사용한다. 또한 단어들의 가중치 W를 계산(TF-DF : Term Frequency, Document Frequency)해서 최상위 가중치 값을 가진 단어들을 선택(Feature Selection)한다. 이때, 가중치값에 대한 임계값 (이상인 단어들만 선택한다.
본 논문에서는 IDF 대신 DF를 사용한다. 그 이유는 다음 과 같다. IDF는 전체 문서 집합에서 일부 문서에 집중적으로 나타나는 단어에 대해 높은 가중치를 할당하는 방법이다. 반면에, DF는 전체 문서 집합에서 해당 단어가 나타나는 빈도수를 의미한다. 즉, 보다 많은 문서에서 단어가 나타날수록 높은 가중치를 할당하는 방법이다. 제안 하는 시스템에서 커뮤니티를 특징 지을 수 있는 단어들을 추출하기 위해서는 각 단어가 커뮤니티 내의 여러 문서에서 두루 나타날수록 보다 효과적일 것이다. 때문에 IDF가 아닌 DF가 적합한 가중치 계산을 위한 한 요소가 된다.

특징 선택에 사용되는 가중치의 계산은 다음의 계산식을 통해 이루어진다.

$$W_{d,t} = tf_{d,t} \text{ X } df_{t,d} \quad (3)$$

$tf_{d,t}$는 문서 d에서 term t가 발생하는 빈도수를 나타내고, $df_{t,d}$는 전체 문서집합에서 term t가 발생하는 문서의 수를 의미한다. 유사도 측정은 식(3)을 사용해서 이루어지며, 이는 질의어와 커뮤니티간 유사도 측정 뿐만 아니라 질의어와 문서간 분석 관리자는 UDDI 레지스터에 등록된 해당 커뮤니티의 정보와 비교해서 중복성 검사를 통해 중복되는 데이터는 제거하고, 그렇지 않은 데이터에 대해 UDDI 레지스터를 갱신 하도록 한다.

3.5 UDDI 레지스트리

UDDI 레지스트리는 분석 관리자, 질의어 관리자, 검색 관리자와 상호작용하며, 각 커뮤니티의 특징 벡터를 저장하고 있다. UDDI 레지스터는 각종 정보들을 생성, 저장, 검색할 수 있는 XML 기반의 자료 저장장치를 의미한다[10]. 질의어 관리자나 검색 관리자는 UDDI 레지스트리의 접근을 위해 SOAP(Service Oriented Access Protocol)을 이용한다. SOAP은 XML 언어를 이용한 분산환경에서의 정보교환을 위한 프로토콜이다[10]. SOAP은 XML로 구성되어 있기 때문에 XML을 이해할 수 있는 모든 시스템은 SOAP을 통해서 통신할 수 있다. 즉, 이종의 플랫폼 응용프로그램간에도 정보를 교환할 수 있다는 장점이 있다.

UDDI 레지스트리가 XML 기반의 자료 저장장치이기 때문에 UDDI 레지스트리 개발언어 및 실행 플랫폼과는 상관 없이 UDDI 레지스트리 간의 데이터 교환이 자유롭다. 이는 데이터 교환 포맷으로 XML 문서를 사용하기 때문이다.

3.6 질의어 관리자

웹 서비스 요청자인 사용자와 직접 상호작용하며, 사용사용부터 웹 서버를 통해 질의어를 입력 받으면 질의어에 대한 전처리 과정을 수행 한다. 전처리 과정은 형태소분석과정과 분석된 질의어의 벡터 모델 표현 과정으로 이루어 진다. 먼저 형태소 분석 과정은 형태소분석 관리자와 상호작용을 통해 입력된 질의어를 형태소 단위로 분석하는 전과정을 의미한다. 두 번째 과정은 데이터 파서와의 상호작용을 통해 이루어 지며 분석된 질의어를 벡터 모델로 표현하는 과정이다.

전처리 과정이 끝난 질의어는 검색 관리자에게 전달되어 커뮤니티내 문서를 검색하는데 사용된다.

3.7 검색 관리자

UDDI 레지스트리에 있는 정보를 검색하고 질의어와의 유사도에 따라 검색된 결과를 웹 서버를 통해 사용자에게 반환한다. UDDI 레지스트리에서 문서들의 검색은 두 단계로 이루어 진다. 먼저 유사한 커뮤니티를 찾고 해당 커뮤니티에서 문서를 검색하는 과정으로 이루어 진다. 질의어와 커뮤니티간 유사도 측정은 입력된 질의어와 커뮤니티의 특징 벡터간 유사도를 측정함으로써 이루어 진다.

4. 결론 및 향후 연구

4.1 결 론

본 논문에서 제안한 커뮤니티 기반 효율적인 웹 검색 시스템 설계는 관련 연구에서 언급한 포커스 크롤링, 웹 문서 클러스터링, 제한 검색의 각 문제점을 해결하고자 하였다.

제안한 시스템은 커뮤니티 내의 대부분의 정보가 크롤러에 의해 특정 어휘나 주제에 의해 하나로 분류 될 수 있기 때문에 포커스 크롤링이 가지는 질의 처리 시간보다 상대적으로 빠르고 제한 검색이 가지는 검색 결과를 줄일 수 있을 것으로 예상된다.

웹 문서 클러스터링이 가지는 문제점은 문서의 분류에 대해 색인 등의 전처리 과정을 거치기 때문에 복잡하고 유지 보수가 어렵다는 것이다. 이러한 문제점은 UDDI에 등록된 커뮤니티의 특징들은 커뮤니티에 업데이트 되는 내용을 RSS를 통하여 분석하고 분석된 내용에 맞게 커뮤니티의 특징을 변화시켜 저장함으로써 상대적으로 유지보수가 쉬울 것이라 판단된다.

4.2 향후 연구

제안한 시스템의 효율성을 확인하기 위해 실험 및 평가를 실시한다. 실험을 위해 정보검색의 기본 척도인 정확율과 재현율을 계산하고 관련연구에서 언급된 각각의 검색방법과 질의 처리 시간을 비교한다. 커뮤니티 자체가 특정 질의 즉 주제에 대해 크롤링 되어있는 상태를 증명해 본다. 웹 사용자의 특정 관심사를 반영 할 경우 해당 커뮤니티는 신뢰성을 가지는 것으로 간주하여 검색에 반영 한다. 이와 같이 커뮤니티가 검색에 반영된 결과와 기존의 검색 방법을 통해 획득한 결과를 비교함으로써 검색의 효율성을 검증 할 것이다.

참 고 문 헌

[1] 이재길, 이민재, 김민수, 황규영, “오디세우스 객체관계형 DBMS를 사용한 사이트 제한 검색의 구현,” 한국정보과학회 춘계학술발표회 논문집, pp. 755-757, 2003년 4월.

[2] S. Sizov, J. Graupmann, M. Theobald. “From Focused Crawling to Expert Information: an Application Framework for Web Exploration and Portal Generation. “*VLDB*, 2003

[3] De Bra, G. Houben, Y. Koranatzky and R. Post. “Information Retrieval in Distributed Hypertexts.” Proceedings of the 4th RIAO Conference, 481-491, New York, 1994.

[4] S. Chakrabarti, M. van den Berg and B.Dom. “Focused crawling: a new approach to topic-specific Web resource discovery.” WWW-8. 1998

[5] Zamir, O. and Etzioni, O., “Web Document Clustering: a Feasibility Demonstration,” In Proc. 19 Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 46-54, Melbourne, Australia, June 1998.

[6] 네이버 용어 (http://terms.naver.com/item.nhn?dirId=200&docId=11311)

[7] Shkapenyuk, V. and Suel, T., “Design and implementation of a High Performance Distributed Web Crawler,” In Proc, of the 18th Int'l Conf. on Data Engineering, San Jose. California, Feb. 2002

[8] Zahn, C., “Graph Theoretical Methods for Detecting and Describing Gestalt Clusters,” IEEE Trans. On Computers, Vol. C-20, No. 1, pp. 68-86, Jan. 1971

[9] W3C Web Services WG. "Web Services Architecture", http://www.w3.org/TR/ws-arch/ W3C Working Group Note 11 February 2004.

[11] UDDI. “The UDDI Technical White Paper”, http://uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, UDDI.org, September 2000