사용자 패턴을 분석한 지능형 메타 검색 시스템 구현

범수한 김복용 이동원 서대영 오용철
한국산업기술대학교 컴퓨터공학과
e-mail:{bumsh, kby86, dw310, seody, oh}@kpu.ac..kr

User-patterns Analysis Intelligent Meta-search System Implementation

Su-Han Beom Bok-Yong Kim Dong-Won Lee Dae-Young Seo Yong-Chul Oh
Dept. of Computer Engineering, Korea Polytechnic University

요 약

최근 인터넷이 보편화되면서 검색에 대한 관심도가 높아지고 있다. 특히 사용자는 정확한 키워드의 입력 없이도 자신이 원하는 검색을 하고 싶어 한다. 그러한 욕구를 충족시키기 위해서 네이트의 '시맨틱', MSN의 'Bing' 등이 새로 제작되어 지고 있으며 네이버, google 등 대형 포털 사이트들도 검색분야에 투자를 아끼지 않고 있다. 본 논문은 사용자중심의 검색을 구현하기 위해서 패턴을 분석하여 연관규칙을 사용하여 검색시간을 단축함을 물론 검색결과의 정확성을 높였다. 구현을 위해서 네이버 사이트의 블로그로 검색의 범위를 한정 하여 데이터를 분석, 관리 및 시각화 하는 사이트를 개발하였다. 또한 검색을 위한 크롤러, 루씬 등을 실질적으로 직접 개발 활용 하였다. 시제품의 시험결과 정답사이트 도출 정확도는 google에 비해 20%, 재현율은 7.2%의 향상성을 보였다.

1. 서 론

EMC-IDC 보고서에 따르면 2010년도에 생성될 디지털 정보량은 1.2조 기가바이트(1.2 Zetabyte)가 된다고 추정하고 있다. 오늘날 전 세계 인터넷에서 찾을 수 있는 정보의 양은 우리가 예측조차 할 수 없는 분량이다. 앞으로 2020년에는 2009년 대비 44배 이상 증가 할 것이라 예측하고 있다. 또한, 전 세계에서 가장 많은 데이터를 가지고 있는 Google의 검색 엔진도 인터넷에 게시된 정보의 약 30% 정도만 색인하여 검색이 가능하다고 한다. ICT(정보통신기술) 연구자들에 의하면 정보를 생산하고, 저장하고 그리고 전달하는 기술은 어느 정도 해결되었다고 한다. 다만 아직도 해결하지 못하는 기술은 정보를 찾는 검색 그리고 정보를 발견하는 것이라고 말하고 있다.[1][2]

위의 문제들을 해결하기 위해서 검색엔진은 나날이 발전하여 왔으며 현재 4세대 검색 엔진, 즉 3세대 로봇 검색 엔진이 차세대 검색엔진이라고 불린다. 1세대부터 3세대 검색 엔진 까지 약 20년 동안 모든 검색엔진들은 좋은 정보를 찾아내는 방법으로 단순히 내용 기반 방법을 사용하고 있었다. 내용 기반 방법은 문서 내의 단어 빈도와 단어 위치를 이용하는 것이다. 그러나 많은 경우에 사용자는 단순히 문서 내의 단어 빈도와 단어 위치를 이용한 것이 아닌 다수의 사용자에 의해 검증된 대표성이 높거나 인기도가 높은 웹 페이지를 원한다. 그것이 4세대 엔진이며 대표적으로 Nate에서 개발된 시맨틱, 추가적으로 네이버 및 MSN에서 계속적인 검색엔진을 개발하고 있다.[3][4]

본 논문에서는 이러한 현상에 맞추어 찾는 검색과 발견하는 것을 향상 시켜 사용자는 단순히 주어진 단어가 많은 페이지가 아닌, 다수의 사용자에 의해 검증된 대표성이 높거나 인지도가 높은 웹페이지를 발견하도록 하는 방법을 제안하고, 이러한 방법을 적용한 메타 검색엔진의 프로타입을 구현하였다. 다른 웹 페이지들로부터의 링크가 많거나 사용자들로부터 클릭이 많다는 것은 그 페이지가 다수의 사용자들에게 중요한 페이지로 인식되어진다고 할 수 있기 때문이다.

본 논문에서는 정확도와 재현율(precision and recall)을 구하는 방법을 택하여 실험해 보았다. 실험결과, 제안 방법이 Google 사이트에 비해 향상된 것을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서 기존 사이트의 특징과 기법 등을 살피고, 3장에서 제안한 방법에 따른 각 시스템 내용과 기술을 설명한다. 4장에서 시스템 구성 및 실험을 통한 향상성을 증명하고, 마지막으로 5장에서 결론을 기술한다.

2.관련연구

2.1 검색기법 분석

인터넷 검색엔진은 웹상에서 존재하고 있는 정보에 대한 검색을 가능하게 하는 시스템이다. 검색엔진에는 자료수집을 하는 웹봇과 에이전트, 스파이더 등으로 링크사이

를 오가며 페이지 정보를 가져온다. 가져온 페이지 정보는 색인 프로그램인 루씬을 통하여 페이지의 단어를 검색자가 소화할 수 있는 형태로 조각조각 나누게 된다. 나누어진 단어들은 자료수집 프로그램에서 가져온 페이지 정보와 색인 프로그램에서 나누어진 단어 간의 목록이 생성되게 되면 이 단계에 이르면 검색 자가 원하는 단어를 검색엔진을 통해서 원하는 단어를 가진 페이지 목록들을 검색자에게 제공 한다.[5][6][7]

2.2 크롤러

WWW(World Wide Web)상의 많은 웹페이지를 사용자가 직접 일일이 확인하여 원하는 정보를 얻기란 힘들다. 이 작업을 대신하는 프로그램을 웹 크롤러라고 칭한다. 웹 크롤러는 웹서버를 자동으로 순회하며 각 웹서버의 웹 페이지를 분석하고 그 안에 있는 URL을 추출하여 URL에 접속하여 다시 페이지를 분석한다.[8]

2.3 루씬

색인과 검색 기능을 간단하게 추가할 수 있도록 지원하는 확장 가능한 고성능 정보검색(IR, Information Retrieval) 라이브러리다. 루씬은 초기 자바로 개발되었으며 속도를 개선하기 위해 C로 포팅 되어 사용 되었다. 일반적으로 루씬이라고 하면 영어 형태소 분석기로 한글 및 기타 언어를 분석하는데 부족한 점이 많이 있어 오픈 프로젝트로 한국 및 여러 국가에서 계속 개선 및 문제 등이 해결 되고 있다. 한국어 처리가 가능한 루씬을 한국어 형태소 분석기라고 명칭 되고 있다. 루씬은 텍스트 형태로 되어 있는 것 이라면 모든지 색인하고 검색할 수 있다.[9][10]

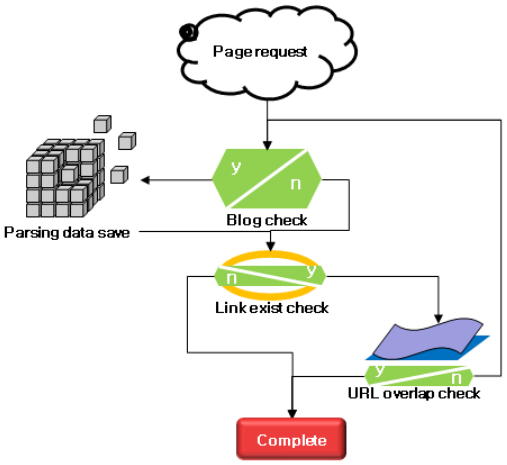
2.4 데이터마이닝

대규모로 저장된 데이터로부터 과거에는 알지 못했지만 데이터 속에서 도출된 새로운 데이터 모델을 발견하여 미래의 의사 결정에 이용하는 과정을 말한다. 즉 데이터 안에서 체계적이고 자동적으로 통계적 규칙이나 패턴을 찾아내는 것이다.[11]

3. 지능형 메타 검색 시스템 구조

3.1 지능형 메타 검색을 위한 크롤러

하나의 일반 웹 크롤러를 가지고 모든 웹 사이트의 페이지를 크롤링 하는 것은 사실상 불가능하기 때문에 블로그만 수집하는 크롤러를 개발한다. Smart and Simple Web Crawler를 활용 및 수정하여 수집 페이지에 대한 체크와 분석 결과를 파일로 출력한다. 그림 1 은 구현 크롤러의 순서도이다.[12]



(그림 1) 구현 크롤러 순서도

본 논문에서 크롤러의 순서는 웹 서버에 페이지를 요청하고 응답받은 페이지를 분석하여 블로그 데이터인지 확인한다. 블로그 데이터일 경우 본문 제목과 내용을 파일로 출력하여 본 서버에 저장을 하고 블로그 데이터가 아니라면 파일출력 없이 곧바로 페이지 안에 다른 URL의 Link여부를 체크하여 Stack에 저장한다. 이 과정에서 Link된 URL이 있을 경우, Database에 URL이 중복되었는지 체크하고, 중복이 되지 않은 URL이라면 페이지를 요청하여 분석 작업을 반복한다. 현재 분석하는 페이지에 Link된 URL이 없다면 현재 페이지 분석 작업을 마치고 Stack에 있는 URL정보를 POP하여 페이지를 호출하여 분석 작업을 반복한다. Stack에 URL정보가 없을 경우 크롤러는 작업을 끝낸다.

3.2 지능형 메타 검색을 위한 루씬

본 논문에서 루씬을 개발하는 것이 목적이 아니기 때문에 크롤러에서 가져온 웹페이지 정보를 색인 하는 부분은 강승식 교수의 한글공학연구소에서 개발된 라이브러리인 한국어 형태소 분석기 KLT(Korean Language Technology)를 활용 했다. KLT의 기능 중 텍스트의 문서를 분석하여 결과 텍스트 문서로 만드는 기능만을 활용하여 추가 적으로 무인 실행기(runLucen)와 인덱싱 프로그램(runQuery)을 제작 했다. 그림 2 는 루씬을 중심으로 구현된 runLucen과 runQuery 프로그램의 시스템 구성도이며 선수 작업으로 runLucen 동작되어 저장소에 있는 텍스트 파일을 한국어형태소 분석을 하여 다시 저장소에 분석된 결과를 저장해 둔다. runQuery는 분석된 결과를 가지고 구조해석, 단어 처리, 링크처리 및 랭킹을 조합 정리 하여 데이터베이스에 인덱싱한다.

(그림 2) 루씬 시스템 구성도

3.3 사용자 중심의 검색 구현

3.3.1 사용자 중심의 방문 로그 데이터 생성

본 논문에서 사용자가 입력한 질의와 상태정보를 조합하여 연관규칙 생성 데이터로 사용한다. 사용자 로그 생성은 검색에 사용된 질의와 방문 사이트의 정보 및 방문 시간 등의 필요한 항목을 추출하여 생성된 데이터를 테이블에 저장한 것이 그림 3 이다.

LogID	Ip	UrlID	WordID	LTime
230	0:0:0:0:0:0:0:1	26994	965527	20:37:5
231	10.50.3.8	26994	965527	20:23:41
232	10.50.3.8	26994	965526	20:24:11
272	192.168.10.10	120687,83085,53804	74668	14:28:5
235	0:0:0:0:0:0:0:1	5661	418	2:39:32
236	0:0:0:0:0:0:0:1	15066,5375,9776,17483	60732	3:1:34
237	10.50.3.58	15066,5375,9776,17483,7847,27967	60732	2:51:13
238	210.93.61.167	7847,27967	60732	2:50:38
270	192.168.10.3	72292,83929,94365	645198	14:24:7
269	192.168.10.1	72292,83929,94365	645198	14:23:47

(그림 3) 방문 로그 데이터 테이블

3.3.2 사용자 중심의 연관규칙 생성

본 논문에서는 표 1에서 트랜잭션 데이터를 이용하여 연관규칙을 생성한다. 생성된 로그의 트랜잭션 T가 주어지면, 연관규칙 탐색을 통하여 지정된 최소 지지도(Minimum Support)와 최소 신뢰도(Minimum Confidence)를 만족하는 연관 사이트를 찾는다. 여기서 연관 사이트를 찾는 알고리즘은 Apriori 기법을 사용하고 최소 지지도와 최소 신뢰도를 50%로 지정한다.

<표 1> 연관규칙 생성을 위한 트랜잭션 데이터

Transaction	Items	QueryID
1	URL1, URL2, URL3	10
2	URL1, URL2, URL3, URL4	10
3	URL4, URL5	10
4	URL1, URL4, URL7	10

연관규칙을 탐색하기 위해서는 첫 번째로 빈발 항목집합들(Large Itemsets)을 찾아야 한다. 빈발 항목집합(Large Itemsets)은 미리 지정한 최소 지지도(Minimum Support) 이상의 트랜잭션의 항목 집합들을 찾는다. 각 트랜잭션에 대한 지지도와 신뢰도는 표 2의 식 을 이용하여 계산한다.

<표 2> X->Y 일 때 지지도와 신뢰도에 대한 식

$$\text{Support} = \frac{(X \cup Y)\text{을 포함하는 트랜잭션 수}}{\text{전체 트랜잭션의 수}}$$

$$Confidence = \frac{(X \cup Y)\text{을 포함하는 트랜잭션 수}}{X\text{를 포함하는 트랜잭션 수}}$$

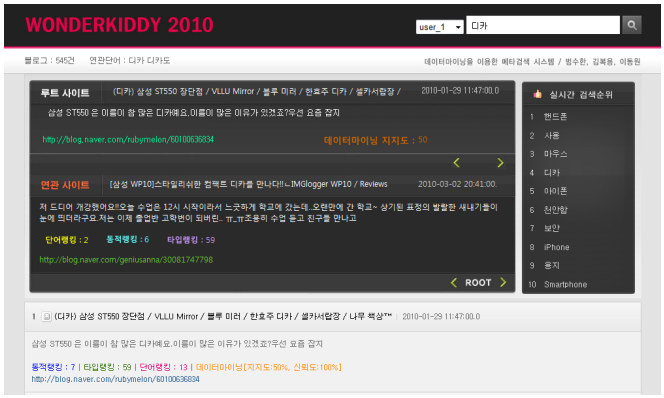
두 번째로 빈발 항목집합(Large Itemsets)을 이용하여 최소 지지도(Minimum Support)와 최소 신뢰도(Minimum Confidence)를 계산하여 후보 집합 항목을 생성하고 생성된 후보 집합 항목에서 최소 신뢰도 이상의 항목을 찾아 최종 집합항목을 생성한다. 여기서 최종 집합항목은 연관규칙을 적용한 사이트들이며 이 데이터는 데이터베이스에 저장된다.

<표 3> 최종 집합항목

Transaction	Items	Support	Confidence
1	URL1, URL2, URL3	50%	66%

3.3.3 사용자 중심의 연관규칙 적용 화면

그림 4는 데이터베이스에 저장되어 있는 최종 집합항목을 이용하여 플래시로 검색결과를 시각화 한 화면이다. 루트사이트는 항목의 첫 번째 사이트로 정하였으며 연관 사이트 리스트를 상위에 나타나도록 했다.

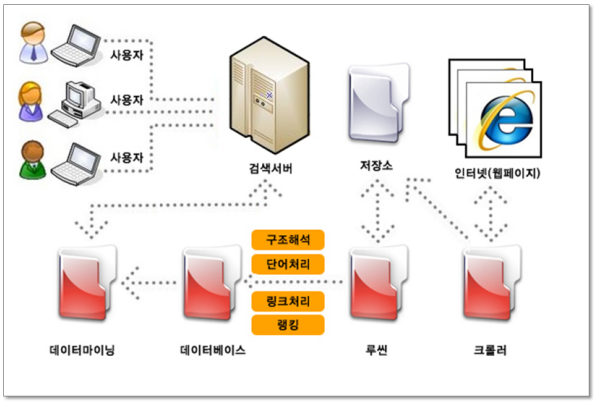


(그림 4) 최종 집합항목을 이용한 검색결과 시각화 화면

4. 시스템의 구현 및 적용

본 논문에서 제시한 시스템은 한국어 형태소 분석 모듈을 이용하여 지능형 메타 검색 루씬을 구현 하였고, Smart and Simple Web Crawler를 이용하여 지능형 메타 검색 크롤러를 구현 하였으며 Sun Microsystems JAVA를 이용하여 사용자 중심의 검색을 구현했다.
그림 5는 구현된 시스템 구성 도를 도식화 하였으며 개발

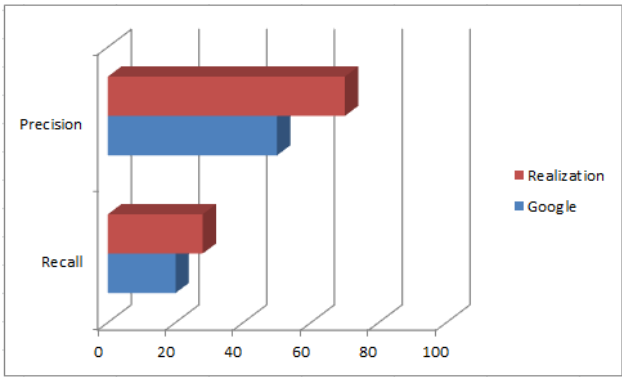
된 프로그램은 통합 개발 환경으로 구성했다.



(그림 5) 전체 시스템 구성도

본 논문에서 제안한 시스템의 검색결과 정확성을 파악하기 위하여 Precision and Recall 방법으로 측정 했다. Precision은 시스템이 찾아온 것(B)중 정답인 것(A∩B)의 비율을 말한다. 예를 들면 첫 페이지에 10개의 검색 결과가 있고 이 중 정답이 개수가 2개이면 Precision은 20%가 된다. Recall은 실존하는 전체 정답(A)중 정답인 것(A∩B)의 비율을 말한다. 전체 검색결과의 정답 개수가 25개이고 첫 페이지 정답 개수가 7개면 Recall은 28%가 된다.

그림 6은 Precision and Recall 방법을 이용하여 제안된 시스템과 google 검색 시스템을 비교 테스트 한 결과이다.



(그림 6) google 검색 시스템과 비교

5. 결론 및 향후 연구과제

본 논문에서는 많은 사용자들이 검색 했던 내용이 검색자가 원하는 결과라는 가정에서 시작하였다. 그 가정을 증명하기 위해 사용자의 검색 결과 탐색에 대한 정보를 바탕으로 하여 사용자가 원하는 정보를 상위에 나타냄으로서 검색시간을 단축하기 위한 소프트웨어를 구현했다. 이 과정에서 크롤러를 통하여 특정 사이트의 블로그 정보만을 수집하는 기능을 구현 했으며, 수집된 정보를 한국어 형태소 분석기를 통하여 단어별로 파싱하고 인덱스를 구축하는 기능을 구현 했다. 그리고 사용자에게 원하는 정보를 상위에 보여주기 위하여 데이터마이닝 기법 중 하나인 연관규칙을 이용해 사이트별 연관성을 구축하여 검색시간을 단축하고 시각화된 검색결과를 볼 수 있게 구현 하였다. 결과물을 Precision and Recall 방법으로 측정한 결과 정답사이트 도출 정확도는 google에 비해 20%, 재현율은 7.2%의 향상성을 증명 하였으며 이러한 검색결과 리스트는 타 검색시스템의 기능과 조합하여 적용했을 때 결과도출의 정확도 및 재현율을 향상시킬 수 있으리라 기대되며 차세대 검색 시스템의 개발 및 구축 시간 단축에 공헌되기를 기원한다.

참고문헌

[1] http://korea.emc.com
[2] Dean Allemang / Jim Dendler, Semantic Web for the Working Ontologist, SciTech, 2008
[3] http://ko.wikipedia.org/wiki/%EA%B2%80%EC%83%89%EC%97%94%EC%A7%84
[4] 양기철, 시맨틱 웹과 그 응용, 한국콘텐츠학회 2005 추계 종합학술대회 논문집, 제3권 제2호, 653, 2005
[5] E. Selberg and O. Etzioni, “The MetaCrawler architecture for resource aggregation on the Web,” IEEE Expert, Vol.12, No.1, pp. 8-14, 1997.
[6] S. Lawrence and C. L. Giles, “Inquirus, the NECImeta search engine,” 7th International World Wide Web Conference, pp. 95-105, 1998.
[7] 박상위, 오정석, 이상호, “메타 검색엔진을 위한 페퍼지 및 지능시스템학회 논문지 2004, Vol. 14, No. 6
[8] Jie Xu, Qinglan Li, Huiming Qu, and Alexandros Labrinidis, "Towards a Content-Provider-Friendly Web Page Crawler," In Proc. 10th Int'l ACM Workshop on the Web and Databases, 2007.
[9] Otis Gospodnetic / Erik Hatcher, Lucene In Action, acorn, 2005
[10] http://nlp.kookmin.ac.kr/
[11] 정보기술연구소 논문집, 데이터마이닝 기법을 이용한 효율적인 웹 검색엔진의 설계 및 구현, 제2집, 7, 2000
[12] https://crawler.dev.java.net/