

# 메타 검색 엔진을 위한 질의 매핑 전략 및 설계\*

조용우                      오정석                      이상호

승실대학교 컴퓨터학과

envy@orion.ssu.ac.kr    dbstar@orion.ssu.ac.kr    shlee@computing.ssu.ac.kr

## Strategy and Design Query Mapping for Meta-search Engines

Yong-Woo Cho    Jeong-Seok Oh    Sang-Ho Lee  
School of Computing, Soongsil University

### 요 약

메타 검색 엔진은 여러 검색 엔진의 결과를 통합해 한 화면에 보여주기 때문에 사용자가 한번에 많은 검색 엔진을 사용하는 효과를 준다. 메타 검색에서 사용하고 있는 불리언 질의는 다른 검색 엔진의 연산자를 모두 수용할 수 없기 때문에 고유의 연산자를 정의하여 사용하고 있다. 이때, 메타 검색에서 사용하는 연산자가 검색 엔진에서 제공되지 않으면 사용자 입력 질의를 정확하게 검색 엔진으로 전달하지 못하게 된다. 본 논문에서는 각 검색 엔진의 연산자를 조사 및 분석하여 검색 엔진 대표 연산자를 추출하고, 연산자에 우선순위를 두어 검색 엔진에서 제공하는 연산자로 질의를 재구성하는 방법을 이용함으로써 이러한 문제를 해결하였고, 메타 검색을 위한 새로운 질의 변환 전략을 제시한다. 그리고, 연산자 매핑 테이블과 URL 매핑 테이블을 이용한 메타 검색의 질의 처리 구조를 설계한다. 이것은 메타 검색으로 입력된 사용자 질의를 정확하게 검색 엔진으로 전달하여 사용자가 원하는 정보를 제공함으로써 사용자 신뢰도를 향상시키게 된다.

### 1. 서론

일반적으로 많이 사용하는 검색 엔진은 각각 다른 인덱스를 구축하고 있어 사용자에게 상이한 결과를 보여준다. 또한, 원하는 검색 결과를 얻기 위해서는 다수의 검색 사이트를 찾아야 하기 때문에 많은 검색 사이트의 위치를 알고 있을 필요가 있다. 이와는 달리, 메타 검색 엔진은 다수의 검색 엔진의 결과를 병합하여 사용자에게 보여주기 때문에 원하는 정보를 검색하려면 다수의 검색 엔진을 이용할 필요가 없다. 대부분의 검색 엔진은 사용자에게 정확한 정보를 제공하기 위해, 질의에 특정 기능을 수행하는 연산을 포함하며, 각 검색 엔진마다 연산에 부여하는 의미와 순서가 다를 수 있다. 그러나 메타 검색 엔진은 다수의 검색 엔진에 대해 검색을 수행하기 때문에, 각 검색 엔진에 적합한 질의 형태로 변환시키고, 각 검색 엔진에서 연산에 부여

된 의미와 순서에도 부합되어야 한다. 이를 정확하게 수행하지 못하면 사용자가 원했던 정보를 제공하지 못할 수 있다.

본 논문은 사용자가 입력한 질의를 각 검색 엔진에서 사용되는 연산으로 변환시켜 사용자에게 정확한 결과를 제시할 수 있는 질의 변환 전략을 제시한다. 즉, 메타 검색 엔진의 연산자를 정의하며, 각 검색 엔진에서 수행되고 있는 연산자와 그 의미를 분석하고, 분석된 검색 엔진의 연산자와 메타 검색 엔진의 연산자 사이의 매핑 전략을 제시한다. 그리고, 연산자 매핑 테이블과 URL 매핑 테이블을 이용한 메타 검색의 질의 처리 구조를 설계한다.

### 2. 관련연구

[표 1]은 9개의 국내 주요 검색 엔진의 연산자 종류를 표기한 것이다. 각 검색 엔진은 고유한 질의 연산자를 제공하며, 연산자는 하나 이상이 제공된다. 예를 들어 네이버에서 AND 연산자로서 'AND', '&', '그리고', 등을 연산자로 제공한다. 각 항목에서 '×'의 표기는 해당 연산이

\*본 연구는 한국과학재단 목적기초연구(2000-2-51200-002-3) 지원으로 수행되었음.

검색 엔진에서 제공되지 않음을 의미하고, '○'의 의미는 해당 연산이 검색 엔진에서 지원됨을 의미한다. 또한, 구글을 제외한 8개의 검색 엔진에서 공통적으로 제공하는 연산자는 AND, OR, NOT이며, 이를 '공통 연산자'라 명명하고, 공통 연산자가 아닌 연산자를 '부수 연산자'라 명명한다. 부수 연산자의 종류로는 구문(phrase) 연산, 괄호 연산, 공백 연산, 복합 연산, Near 연산, Within 연산이 존재하며, 의미와 표기법은 검색 엔진에 따라 상이할 수 있다. 예를 들어 파이어볼에서는 공백 연산이 OR 연산으로 간주되어 처리되고, 심마니에서는 공백 연산이 AND 연산으로 처리된다. 야후에서는 복합 연산자 '{}'를 제공하며, 세 개 이상의 단어를 이용해서 AND와 OR 연산을 수행할 때 필수적으로 사용해야 하는 연산자이다.

[표 1] 국내 검색 엔진 연산자 분류

	네이버	라이코스	파이어볼	엠포스	알타비스타	심마니	구글	야후	한미르
AND 연산	&, AND, and, 그리고	AND, &, 공백	+, AND	&	AND, &	AND, &, 공백, ...	공백	공백	*
OR 연산	+,  , OR, 또는	OR,	공백, OR		OR,	OR, +,  , ...	×	( )	+
NOT 연산	!, -, NOT	NOT, !	-, NOT, AND, NOT	!	AND, NOT, !	&!, !&	-	-	-
Phrase 연산	×	" "	" "	공백	×	×	" "	" "	" "
괄호 연산	×	○	×	×	○	○	×	OR 연산	○
공백 처리	언급 없음	AND 연산	OR 연산	Phrase	언급 없음	AND 연산	AND 연산	AND 연산	언급 없음
Near 연산	~n, NEAR, NEAR/n	NEAR, NEAR/n	~, near	~n	NEAR, ~				( )
Within 연산	~n, WITHIN, WITHIN/n	OADJ, OADJ/n		~n					[ ]
기타								복합 연산 : { }	

### 3. 연산자 매핑

모든 검색 엔진에서 사용하고 있는 불리언 연산은 효율적이고 정확한 검색을 위해 검색어 사이의 관계를 지정해 주는 것을 말한다. 메타 검색 엔진에서도 불리언 연산을 사용한다. 기본 연산자의 종류는 검색 엔진에서 많이 사용하는 연산자와 표기법을 이용한다. 메타 검색 엔진에서 사용되는 연산자는 [표 2]와 같으며 사용자는 이

러한 기준으로 질의를 작성하게 된다. 각 연산에 대한 처리는 일반적인 연산자 처리 기준에 의한다. AND 연산

[표 2] 메타 검색의 기본 연산자

항 목	연산자
AND 연산	&
OR 연산	
NOT 연산	!
괄호 연산	( )
구문 연산	" "
공백 연산	&

은 검색어로 입력된 단어가 모두 포함된 문서를 검색한다. OR 연산은 검색어로 입력된 단어 중 하나라도 포함된 문서를 검색한다. NOT 연산은 검색어로 입력된 단어가 포함되어 있지 않는 문서를 검색한다. 괄호 연산은 질의가 3개 이상의 검색어로 구성된 경우 먼저 검색될 단어의 조합을 결정한다. 구문 연산은 입력받은 질의 형태 그대로 존재하는 구문을 포함하는 문서를 검색한다. 마지막으로 공백 연산은 AND 연산과 동일한 의미로 사용된다. 연산자의 우선 순위는 구문 연산, 괄호 연산의 순이며, 나머지 연산자의 우선순위는 좌에서 우로 한다. 메타 검색 엔진에서 수행되는 연산자와 각 검색 엔진에서 수행되는 연산자는 표기 방식이 틀리기 때문에 메타 검색 엔진의 연산자를 직접 특정 검색 엔진에서 사용할 수 없는 경우가 많다. 이를 위해 기준 연산자와 해당 검색 엔진에 대한 연산자 매핑이 필요하다. 연산자 매핑은 메타 검색 엔진에서 입력받은 질의에 포함되는 연산자를 해당 검색 엔진의 연산자로 변환시켜주는 작업을 수행한다.

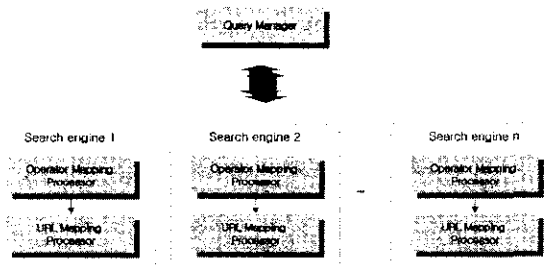
[표 3] 연산자 매핑 테이블

	네이버	라이코스	파이어볼	엠포스	알타비스타	심마니	구글	야후	한미르
AND 연산	&	&	+	&	&	&	공백	공백	*
OR 연산			공백			+	×	( )	+
NOT 연산	!	!	-	!	AND, NOT	&!	-	-	-
Phrase 연산	×	" "	" "	공백	×	×	" "	" "	" "
괄호 연산	×	○	×	×	○	○	×	OR 연산	○
공백	&	공백	+	&	&	공백	공백	공백	*
기타								복합 연산 : { }	

[표 3]은 입력받은 연산자를 해당하는 검색 엔진의 연산자로 변환할 때 기준이 되는 매핑 테이블이다. 2장에서 언급한 바와 같이, 각 검색 엔진은 특정 기능에 한 연산자만 제공하는 것이 아니라 하나 이상의 연산자를 제공하기 때문에, [표 3]에 정의된 연산자는 각 검색 엔진의 대표 연산자를 추출하여 지정한 것이다.

#### 4. 질의 처리기의 구조

질의 처리기는 사용자 질의, 페이지 검색 결과 개수, 범주(category), 검색 엔진 이름을 사용자로부터 입력받고, 입력된 정보를 이용해서 질의 연산자 매핑과 URL 매핑을 수행한다. 질의 연산자 매핑은 [표 3]과 같이 구성된 테이블에 의해 입력된 연산자를 해당되는 검색 엔진의 연산자로 변환을 수행한다. URL 매핑은 URL 매핑 테이블에 의해 검색 엔진의 사이트에 접속할 수 있는 URL 및 검색 조건을 구성한다. 출력되는 항목은 사용자 질의, URL 질의, 검색 엔진 이름, 검색 엔진 연결 개수이며, 각 검색 엔진으로 질의를 보내는 래퍼(wrapper) 처리기로 결과를 출력하게 된다. 이러한 작업은 검색 엔진별로 구분된 스레드(thread)로 처리가 된다. [그림 1]은 질의 처리기 전체 구성도를 나타낸 것이다.



[그림 1] 질의 처리기 관련 구조도

#### 5. 결론 및 향후계획

9개의 국내 검색 엔진을 분석한 결과, 메타 검색 엔진에서 제공하는 기본 연산자 중에서 AND, OR, NOT, 공백 연산자의 경우 구글을 제외한 모든 검색 엔진에서 제공하고 있지만, 괄호 연산자와 구문 연산자는 모든 검색 엔진에서 제공되지 않고 일부 검색 엔진에서만 쓰여진다는 것을 알 수 있다. 사용자 입력 질의를 검색 엔진으로 전달할 때, 괄호 연산자와 구문 연산자를 제공하지 않는 검색 엔진에서는 연산자 처리에 문제가 발생하게 되어, 사용자 의도에 맞추어 질의를 전달하지 못하는 경우가

발생된다.

이러한 문제는 괄호 연산자의 경우, 연산자에 우선순위를 부여해서 사용자가 입력한 질의를 모든 검색 엔진에서 제공되는 AND, OR, NOT, 등의 연산자로 재구성하는 방법을 사용하고, 구문 연산자는 검색 엔진에서 제공하는 Within 연산자를 이용하여 질의를 재구성하는 방법을 사용함으로써 해결한다. 향후, 괄호 연산과 구문 연산 문제를 해결할 수 있는 알고리즘을 수행하고, 여기서 나온 규칙들을 수식으로 정립하여 새로운 질의 변환 모델을 제시할 것이다.

#### 6. 참고문헌

- [1] K. Chang, H. G. Molina, A. Paepcke, Boolean Query Mapping Across Heterogeneous Information Sources, IEEE Transaction on Knowledge and Data Engineering, vol 8, no 4, August 1996.
- [2] W. B. Frakes, R. Baeza-Yates, Information Retrieval - Data Structures & Algorithms, Prentice Hall PTR, 1992.
- [3] 네이버 도움말, <http://help.naver.com/search/advanced.html>
- [4] 라이코스 도움말, [http://www.lycos.co.kr/search/search\\_help.html](http://www.lycos.co.kr/search/search_help.html)
- [5] 파이어볼 도움말, <http://dir.daum.net/help-search.html>
- [6] 엠파스 도움말, [http://www.empas.com/help/bul\\_search.html](http://www.empas.com/help/bul_search.html)
- [7] 알타비스타 도움말, [http://www.altavista.co.kr/help/today\\_extend.htm](http://www.altavista.co.kr/help/today_extend.htm)
- [8] 심마니 도움말, [http://supp.simmani.com/guide/fuide\\_07.html](http://supp.simmani.com/guide/fuide_07.html)
- [9] 구글 도움말, <http://www.google.com/intl/ko/help.html>
- [10] 야후 도움말, <http://kr.help.yahoo.com/help/ysearch/searchhelp.html>
- [11] 한미르 도움말, [http://www.hanmir.com/info/web\\_help.html](http://www.hanmir.com/info/web_help.html)