클라이언트/서버 환경에서 효율적인 시소러스 작업 모형 설계

장유진*, 최종필, 김민구
아주대학교 컴퓨터공학과

Constructing Effective Thesaurus Working Model on Client/Server Environment

You-Jin Chang*, Jong-Pill Choi, Min-koo Kim
Dept. of Computer Engineering, Ajou University

요 약

정보 검색 시스템은 사용자의 질의어를 용어들과 용어들 사이의 관계 집합으로 구성된 일종의 용어 사전인 시소러스를 이용하여 문헌에 대한 색인과 검색을 정확하고 통제된 용어 형태로 바꾸어 색인과 검색 작업의 효율을 높인다. 클라이언트/서버 환경에서 시소러스를 이용하여 정보 검색을 할 때 서버에만 있던 시소러스를 클라이언트마다 분배함으로써 서버의 부담을 줄이며 전체적인 정보 검색 속도의 증가를 기대할 수 있다. 분산된 시소러스는 프로파일 정보를 가지고 운영되며 전문적 시소러스로 만들어진다. 본 논문에서 제안한 시소러스 작업 모형을 시뮬레이션 한 결과를 비교, 분석하고 클라이언트/서버 환경에서 효율적인 시소러스의 역할 및 작업 형태에 대해 제안한다.

1. 서론

정보 검색 시스템(Information Retrieval System)이란 사용자가 필요로 하는 정보를 수집하여 내용을 분석한 뒤 찾기 쉬운 형태로 조작하여 정보에 대한 요구가 발생하였을 때 해당정보를 찾아 제공하는 시스템을 말한다[1].이러한 정보 검색 시스템에서 전문가가 아닌 일반 사용자가 자신이 원하는 정보에 대한 정확한 색인어를 질의어로 입력하는 것은 어려운 일이다. 정보 검색 시스템이 가지고 있는 문서와 사용자의 질의의 내부 표현과정에서 사용되는 정보 혹은 지식베이스의 종류는 여러 가지가 있으나 일반적으로 시소러스(Thesaurus)가 사용된다[2]. 시소러스는 어떤 문헌에 대한 색인 작업 시 적절한 색인표목의 선택과 색인어의 통제를 위해 사용될 뿐만 아니라 검색 시에는 적절한 탐색어의 선택을 위해 사용된다[3]. 기존의 정보검색 시스템은 대부분 클라이언트/서버 환경으로 구축되어 있는데 사용자의 요구가 늘어날수록 서버에 부담이 늘어났고 그에 따라 좀더 효율적인 구조를 필요로 하게 되었다. 사용자의 질의에 도움을 주는 지식베이스인 시소러스를 클라이언트/서버 환경의 정보 검색 시스템에서 클라이언트쪽에도 분배하여 검색 효율을 높여 보고자 했다. 시뮬레이션을 통해 서버에서만 시소러스 작업을 수행 할 때보다 클라이언트마다 시소러스를 분산시켜 작업을 수행하면 전체적인 검색속도의 향상을 얻을 수 있음을 알았다. 또한 프로파일(profile) 정보를 이용하여 클라이언트의 성향을 알고 그에 따라 전문적 지식형태로 구축된 시소러스를 통해 검색결과의 질적 향상을 얻고자 연구했다.

본 논문의 구성은 다음과 같다. 2 장에서는 클라이언트/서버 환경에서 시소러스의 두 가지 작업 모형을 설명하고 두 시스템의 구조적 차이점을 비교한다. 3 장에서는 시뮬레이션을 통해 두 모형 간의 성능을 평가함으로써 속도측면의 검색효율에 대해 서술한다. 4 장에서는 검색결과의 질적 향상에 필요한 시소러스 구조에 대해 제시한다. 마지막으로 5 장에서는 결론과 향후계획에 대해 서술한다.

2. 클라이언트/서버환경에서의 시소러스 작업 모형

2.1 검색서버에서만 시소러스 작업을 하는 경우

클라이언트/서버환경의 검색시스템에서 시소러스가 어느 곳에서 어떤 역할을 하는가에 따라 다음 두 가지 모형을 생각할 수 있다. 먼저 검색서버가 거대한 양의 문서집합과 검색엔진, 색인파일, 시소러스를 가지고 클라이언트로부터의 요청을 처리하는 작업 모델이다. 사용자는 클라이언트에 접속하여 질의를 하고 클라이언트는 단순히 이러한 사용자의 요구를 서버와 연계하여 서비스를 요청하고 받아오는 작업을 수행한다. 검색 서버는 시소러스를 가지고 클라이언트로부터 받은 질의어를 확장하여 보다 정확한 결과 리스트를 클라이언트에게 돌려준다.

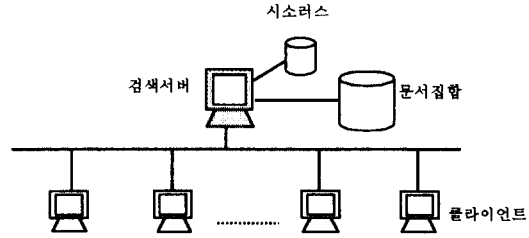

그림 1 클라이언트/서버환경에서 정보검색 시스템 구조

그림 1 과 같은 네트워크 상에 1 개의 검색서버와 다수의 클라이언트가 있는 시스템에서 클라이언트는 단순히 사용자가 검색할 수 있는 인터페이스만을 가지고 있어 질의가 들어오면 용

어(term)들을 서버에게 전달하는 역할만을 한다. 서버는 여러 클라이언트들로부터 들어온 작업들을 순서대로 처리하여 서비스하는 일을 한다. 이때 서버는 지능형 검색을 위해 시소러스를 사용하게 되는데 이 작업을 "Spreading Activation"이라 한다. 보다 양질의 지능형 검색을 위해서 시소러스에 이용하는 작업에 투자하는 시간이 많아지게 된다. 또한 많은 클라이언트들이 한꺼번에 질의를 함으로써 서버의 작업에 부담이 예상된다.

2.2 시소러스작업을 클라이언트로 분산할 경우

앞에서 언급한 기존 검색시스템의 문제점을 해결하고자 다음과 같이 수정된 클라이언트/서버 환경의 시소러스 분산 모형을 제안한다. 즉, 서버가 가지는 부담을 줄이기 위해서 클라이언트마다 자신의 특화 된 시소러스를 구축하는 것이다.

그림 2는 시소러스를 클라이언트들에게 분산시켜 나누어 주었을 때 모습이다.

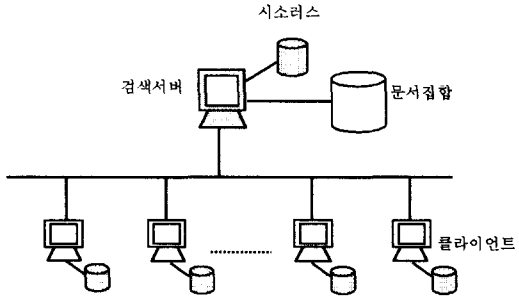

그림 2 클라이언트/서버환경에서 시소러스 분산 모형

시소러스에 들어가는 용어의 정의를 지식적 관점에서 보면 대부분 정의적 지식(Definition Knowledge)에 속한다[5]. 정의적 지식은 사용자의 개별적 특성에 따라 지역적 지식(Local Knowledge)과 전역적 지식(Global Knowledge)으로 나누어진다. 전역적 지식은 일반인들이 공통적으로 사용하는 지식, 즉 상식적 지식(Commonsense Knowledge)에 속하는 지식이다. 지역적 지식은 전문적 지식을 말하며 극히 일부 사람들에만 유용한 지식을 말한다. 물론 전역적 지식도 사람마다 다른 정의를 가질 수 있다. 이런 경우는 전역적 지식이 그 사람의 지역적 지식으로 바뀌어 사용될 수 있고, 의미는 전역적 지식의 의미를 누르고 지역적 지식의 의미로 사용된다. 시소러스 작업의 분산을 위해서 앞에서 설명한 전역적 지식은 서버에 위치시키고 지역적 지식은 클라이언트에 위치시킨다. 즉, 전역적 지식은 서버의 범용 시소러스로 표현되고 지역적 지식은 클라이언트의 특화 된 전문적 시소러스로 표현된다.

클라이언트가 가진 시소러스는 서버의 범용 시소러스의 부분집합(subset)이다. 처음 시소러스를 구축할 경우에는 서버의 시소러스로부터 일부를 가져오게 되는데 이때 클라이언트의 프로파일 정보를 참조하게 된다. 프로파일 정보 안에는 클라이언트의 정보, 성향, 관심분야 등이 들어있다[4]. 예를 들어, 서버의 시소러스가 컴퓨터 용어에 대한 어휘정보를 가진 시소러스이고 클라이언트의 주 사용자는 '정보검색'에 관심이 있다고 하자. 처음 클라이언트는 프로파일 정보에 의해 시소러스의 용어 중에서 '정보검색'에 대한 용어들을 서버의 시소러스로부터 자신의 시소러스로 가져오게 되며 이를 캐쉬(cache)처럼 사용하게 된다. 정보 검색의 과정에서 질의어가 입력되면 클라이언트에 구축된 시소러스부터 조사하게 된다. 입력 된 질의어에 대한 맞는 용어가 없어서 시소러싱이 불가능 할 경우에는 클라이언트는 용어에 대해 서버에게 질의어 확장을 요청하는 방식을 취한다.

서버와 클라이언트에 시소러스를 분산했을 때는 다음과 같은 장점이 기대된다. 클라이언트의 프로파일(profile) 정보를 가지고 시소러스를 구축하여 위에서 언급한 좀더 전문적인 지식베이스를 획득할 수 있으므로 보다 정확한 질의어 확장이 가능해진다. 질의어가 사용자의 요구조건과 문서집합에서 추출된 색인어 집합에 매칭될 확률이 높을수록 검색결과의 정확도가 높아지는 것은 당연하다. 또한 시소러싱 속도에 있어서도 용어의 양이 많아 크기가 큰 범용 시소러스보다 클라이언트측의 특화 된 시소러스는 작은 양의 용어들을 가지고 있기 때문에 빨라진다.

3. 클라이언트/서버환경에서 시소러스 작업 모형 시뮬레이션

시소러스를 서버와 클라이언트에 각각 분배했을 때 검색과 시소러스 작업, 그리고 시소러스 작업의 분담 비율에 따른 효과를 시뮬레이션[6]을 통해 비교,분석해 보았다.

네트워크 상에 검색서비스를 제공하는 서버와 문서집합, N개의 클라이언트들이 있다고 한다(그림 2 참조). 클라이언트의 평균 서비스 요청시간(Average Inter Request Time: AIRT), 문서 집합에 대한 서버의 평균 탐색시간(Average Retrieval Time:ART), 서버의 평균 시소러싱 시간(Average Server Thesaurusing Time: ASTT)와 클라이언트의 평균 시소러싱 시간(Average Client Thesaurusing Time: ACTT)이 각각 주어진다. 일어날 수 있는 검색 서비스의 종류는 다음 세가지로 제한한다. 시소러스를 사용하지 않는 단순 검색(Simple Retrieval:SR), 서버의 시소러스만을 사용(Retrieval with Thesaurusing in Server:RTSR)하는 경우, 클라이언트의 시소러스를 사용(Retrieval with Thesaurusing in Client:RTCR)하는 경우이다. 클라이언트별 서비스 요청에 대한 평균 반응시간(Response Time for Service Request:RTSR)으로써 성능을 측정할 수 있다.

표 1,2는 1개의 서버와 10개의 클라이언트 환경에서 평균 서비스 요구 도착 시간 간격을 60,100 단위시간에 따라 평균 대기시간, 서버의 서비스 처리시간, 반응시간(사용자가 서비스 요청 후 결과를 받을 때까지 총 걸린 시간)을 보여주고 있다. 이때, 서버의 단순 검색 속도는 5이고, 서버 시소러스 작업 속도와 클라이언트 시소러스 작업 속도는 각각 5,10 단위시간으로 하여 시뮬레이션 하였다.

표 1에서는 평균 서비스 요구 도착 간격시간이 클라이언트 수에 비하여 짧아서 대기 시간이 많이 발생한다. 그러나 시소러스 작업이 많고, 이 작업이 클라이언트에서 일어나는 비중이 늘어나면 날수록 분산환경으로 처리할 때의 효과가 급격히 높아지는 것을 알 수 있다.

표 2에서는 평균 서비스 요청시간이 표 1의 경우보다 길어지는 경우이다. 이때도 마찬가지로 시소러스의 작업비율이 늘수록 평균 반응시간이 줄어들면서 분산처리의 효과를 볼 수 있다. 일반적으로 클라이언트/서버환경의 검색 시스템에서 많은 사용자가 동시 다발적으로 사용한다고 한면 이는 평균 서비스 도착 간격이 짧아지게 되어 표 1과 같은 결과를 나타낼 것이다. 따

라서, 시소러스 작업을 클라이언트쪽으로 분산하여 검색할 경우 처리속도 면에서 검색효율을 높일 수 있다.

표 1. 클라이언트 수:10, AIRT=60, ART=5, ASTT=5,ACTT=10

단순검색 대 시소러스 검색의 비율	Server: Client 시소러스 작업의 비율	평균 대기시간	평균 서버 서비스 시간	평균 반응시간
4:1	1:0	236.0	6.1	242.1
	1:1	70.2	5.6	76.8
	1:3	31.9	5.3	38.8
3:1	1:0	423.8	6.2	430.0
	1:1	41.2	5.6	48.2
	1:3	34.8	5.4	42.1
2:1	1:0	1729.7	6.8	1736.6
	1:1	60.6	5.8	68.1
	1:3	32.4	5.5	40.5
1:1	1:0	3220.2	7.6	3227.8
	1:1	450.2	6.4	459.2
	1:3	62.2	5.8	71.9

표 2. 클라이언트 수:10, AIRT=100, ART=5, ASTT=5,ACTT=10

단순검색 대 시소러스 검색의 비율	Server: Client 시소러스 작업의 비율	평균 대기시간	평균 서버 서비스 시간	평균 반응시간
4:1	1:0	9.7	6.1	15.8
	1:1	6.6	5.5	13.1
	1:3	5.5	5.2	12.3
3:1	1:0	11.0	6.3	17.3
	1:1	6.7	5.7	13.8
	1:3	6.4	5.5	13.7
2:1	1:0	15.7	6.8	22.4
	1:1	6.1	5.8	13.6
	1:3	5.6	5.5	13.8
1:1	1:0	19.3	7.6	26.9
	1:1	8.4	6.3	17.3
	1:3	6.2	5.6	15.7

4. 검색결과의 질적 향상을 위한 분산 시소러스 모형

클라이언트/서버환경의 정보검색 시스템에서 클라이언트로 시소러스의 지식을 분산할 때 클라이언트에 배치된 시소러스는 어떤 지식구조를 가져야 효율적인지 알아보자. 서버가 가지고 있는 시소러스는 모든 도메인에 대한 용어를 가지고 전역적 지식을 표현하는 범용 시소러스이다. 이와 반대로, 각 클라이언트의 기호에 맞게 구축된 전문적 시소러스는 도메인이 한정적이며 용어의 수도 범용 시소러스보다 적다(한 용어에 대한 정의가 클라이언트간에 또는 서버와 클라이언트간에 다를 수도 있지만 그 점은 여기서 배제하기로 한다).

클라이언트의 시소러스는 해당 클라이언트의 속성에 가장 잘 맞도록 구축되어야 검색 결과의 질적 효과를 꾀할 수 있으며 이러한 작업을 위해서는 시소러스의 구축이 한번에 이루어질 수 없다. 대부분의 기존 시소러스에서 정보를 얻는데 소요되는 시간은 방법에 있어 차이는 있지만 상당한 시간이 걸리고 시소러스가 정적으로 고정되어있기 때문에 환경 변화에 적응 능력이 부족하다[7]. 이러한 시소러스 시스템의 단점을 해결하기 위하여 심볼릭 시소러스로부터 신경망 기반의 시소러스의 개념을 도입하여 신경망의 내재적인 능력[8]인 대규모 병렬처리를 이용하여 응답시간을 줄일 수 있고 신경망의 학습 능력을 이용하여 환경에 적응하는 시소러스가 연구되고 있다[9].

처음 검색 시스템의 구축 시 클라이언트가 범용 시소러스의 전역적 지식으로부터 자신에 맞는 지역적 지식을 구축할 때 클라이언트의 정보, 관심분야, 성향 등을 기록한 프로파일(profile)을 참조하여 지역적 지식을 가져오고 그 이후에 가장 전문적이고 적절한 시소러스 형태를 갖추기 위해 학습(learning)의 과정을 첨가한다. 이런 방식을 통해 좀 더 정확한 지식베이스로 구축된 시소러스로부터 사용자가 넣은 질의어를 확장할 경우 검색결과의 질적 향상을 예상할 수 있으며 계속 연구를 수행할 예정이다.

5. 결론 및 향후과제

클라이언트/서버환경에서 정보 검색 시스템을 설계할 때 시소러스의 작업 형태를 분산시키면 전체 검색의 처리속도에서 효과를 얻을 수 있음을 시뮬레이션을 통해 비교해 보았고 좀더 전문적으로 구축된 지식베이스를 이용하여 검색 결과의 질적 향상을 얻는 방법을 제안했다. 향후과제로는 특화 된 클라이언트쪽의 시소러스를 구축 시 필요한 프로파일 정보의 포맷(format)을 구성하고 학습능력을 가진 시소러스의 구조를 연구하는 것이다. 현재 프로파일과 캐쉬의 개념만으로 단순화 시킨 시소러스 운영에 대해 좀더 자세한 역할 분담과 작업에 대한 설계가 필요하다. 또한 분산 정보 검색 시스템으로 개념을 확장하여 시스템을 설계할 때 검색서버를 프레임웍(framework)으로, 클라이언트들의 컴포넌트(component)로 간주하여 시소러스끼리 정보를 교환하고 서비스함으로 검색의 질을 높이는 방법도 생각해 볼 수 있다.

6. 참고문헌

[1] 정재헌, "정보검색을 위한 효율적인 시소러스 구조에 관한 연구",한국정보과학회 춘계 학술발표 논문집 Vol. 22 No.1, pp.949-952, 1995

[2] 맹성현, "정보검색의 의미 및 관련 시스템의 소개", 네트웍 타임즈, pp140-143, 1995.11

[3] 정영미, "정보검색론", pp.105-107, 구미무역, 1993

[4] 맹성현, "정보 여과 시스템의 프로파일과 효율성", 네트웍 타임즈, pp119-124, 1996.3

[5] Brachman R.J. and Levesque H.J., "Readings in Knowledge Representation", Morgan Kaoufmann, Los Altos, CA, 1985.

[6] Jerry Banks and John S.Carson,II and Barry L.Nelson, "Discrete-Event System Simulation", Prentice Hall, pp.203-204, 1996.

[7] 최익규, "신경망과 심볼릭 표현의 결합을 기반으로 하는 인식론적인 시소러스 시스템", 아주대학교 석사학위 논문,1995

[8] Simon Hakin, "NEURAL NETWORKS : A Comprehensive Foundation", Macmillan College Publishing Company, N.Y., 1994.

[9] 최종필, "다중 퍼셉트론에 기반한 혼합형 시소러스", 아주대학교 석사학위 논문, 1999.2

[10] 한국데이터베이스 진흥센터, "시소러스 개발표준", 데이터베이스 표준화 연구 보고서, 1996,pp.93 - 95.

[11] William B. Frakes and Ricardo Baeza-Yates , "Information Retrieval Data Structures & Algorithms", Prentice Hall, 1992.