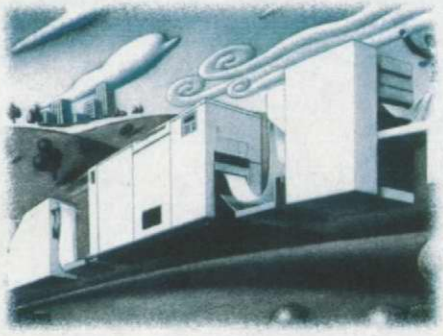
건초더미에서 바늘찾던 시대와 '결별' 선언

자연언어 질의 · 역-인덱싱 · 인터넷 탐색 엔진 · DB 텍스트 탐색 기술 관심



데이터베이스에 저장되는 데이터의 양은 급격히 증가되어 왔고, 사용자들은 필요한 데이터를 찾기 위해 서말이 넘는 땀방울을 흘려야 했다. 그러나 새로운 탐색 기술들이 이러한 문제에 대한 해답을 제시하고 있다. 자연언어 질의, 역-인덱싱, 인터넷 탐색 엔진, 데이터베이스 텍스트 탐색 같은 기술들은 사용자의 오랜 숙원을 해결해줄 기대주로 관심을 모으고 있다. 자유 형식 정보 소스들로부터 정보의 특별한 탐색과 추출을 수행하기 위해 사용 가능한 보다 새로운 탐색 기술들과 기법들을 소개한다.
〈편집자〉

데이터의 대형 쓰레기통, 데이터의 공동묘지. 각 기업의 데이터베이스 안에 저장된 정보를 이런 식으로 표현하는 것을 들어본 적이 있을 것이다.

데이터베이스에 저장되는 데이터의 양은 급격히 증가되어 왔고, 사용자를 쩔쩔매게 하고 있다. 게다가 인터넷, 인트라넷, 엑스트라넷 등 새로운 소스로부터 사용 가능한 엄청난 분량의 정보가 있고, 거기에서도 또한 찾고자 하는 정보를 얻는 일은 마찬가지로 어려운 일이다.

마이크로소프트 워드와 로터스 워드 프로 문서들은 시간이 지날수록 비공식적인 기업 정보 교환의 표준으로 되어가고 있다. 이 모든 정보 소스 가운데에서 데이터베이스 필드, 웹 페이지, 혹은 문서들에서 자유 형식(free-form) 정보를 찾을 때, 다시 말해 '건초더미에서 바늘찾기'를 할 때 사용자에게 도움을 주기 위해서는 어떻게 해야 할까?

새로운 탐색 기술들이 이러한 문제에 대한 한가지 대답이다. 이 기술들은 가장 중요한 정보를 추출하기 위한 특수한 탐색기준을 사용하면서도 탐색에서 걸러진 질의와 관계없는 정보의 양을 최소화함으로써 사용자들이 데이터베이스, 웹 페이지, 그리고 텍스트 문서들에 대해 질의를 할 수 있도록 한다.

이 글은 자유 형식 정보 소스들로부터 정보의 특별한 탐색과 추출을 수행하기 위해 사용 가능한 보다 새로운 탐색 기술들과 기법들을 소개할 것이다.

사용자 요구 폭증

폭발적인 대규모 텍스트 데이터 가운데서 탐색을 하는 작업은 갈수록 어려운 일로 되어왔으며, 이는 특별한 질의 및 보고 툴들의 도움을 받는 경우에도 마찬가지이다. 다음의 요소들이 문제를 야기하고 있다.

- 각자 특유의 탐색기법을 사용하는 많은 소스들이 있다. 데이터베이스는 SQL을 아는 사용자들이나 엔드유저 질의 툴을 필요로 하고, 웹 페이지는 인터넷 탐색 엔진을 필요로 하며, 문서들은 전용의 명령어나 인터페이스를 필요로 한다. 각각의 기법은 서로 다른 구문을 필요로 한다. 각 매체에서 가장 효율적인 사용자가 되기 위해서는 교육이 필요하다.
- 컨텐트의 복잡성은 어떤 정보가

사용 가능한 것인가에 대한 이해를 할 수 없도록 만든다. 그것은 데이터베이스에 있는 정보의 대량성 혹은 수많은 테이블과 필드 간의 복잡한 관계 때문에 야기될 수 있다.

● 질의의 복잡성은 테이블 조인과 부질의들을 필요로 하는 복잡한 SQL로 번역되는 '단순한' 기업질문 때문에 일어난다.

● 정확한 키워드 매칭을 사용하는 탐색은 데이터 소스들로부터 사용 가능한 모든 정보를 제공하지 않는 그러한 검색들로 될 수 있다. 동의어들과 유사의미 단어들이 그 검색 집합에 가치를 부가시킬 수 있지만, 포함되지 않게 된다.

● SQL은 LIKE를 사용하는 능력이 있긴 하지만 제한적이며, '*' 혹은 '_' 등과 같은 와일드카드 기호들을 포함하고 있다. SQL은 데이터의 왼쪽에서 오른쪽의 순서로 텍스트 스트링의 정확한 문자 매칭을 수행할 때 (와일드카드의 사용을 포함하더라도) 최상의 능력을 보인다. 자유 형식 텍스트 필드의 탐색은 종종 테이블 스캔을 필요로 한다.

탐색 기술 개요

몇가지 유형의 탐색용 제품들이 이러한 문제들의 해결을 시도하고 있다. 이 글에서는 다음의 카테고리에 해당하는 제품들에 초점을 맞출 것이다.

● 자연언어 질의: 이 유형의 시스템은 사용자들이 SQL 구문이 아니라 영어(혹은 다른 언어들)를 사용하여 질의를 입력하게 한다. 그러면 이 시스템은 사용자의 영어구문을 SQL로 번역한다. 번역 단계는 SQL의 일부 한계점을 해결할 수 있다.

● 역-인덱싱(inverted indexsing) : 관계형 데이터베이스 열(column)들 위에 역-인덱싱을 창조함으로써 현저한 성능향상과 탐색기능을 구현할 수 있다. 그러면 탐색은 매치하는 것을 찾기 위해서 SQL을 필요로 하는 대신에 인덱스를 이용한다.

● 인터넷 탐색 엔진 : 이 제품들은 문맥에 민감한 탐색을 제공할 수 있으며, 정확한 키워드 탐색 대신에 개념 탐색을 제공할 수 있다. 인터넷 탐색 엔진들은 SQL을 훨씬 능가하는 탐색 및 검색능력을 제공한다.

● 데이터베이스 텍스트 탐색 : 첨단 탐색은 관계형 데이터베이스에 저장된 텍스트에 대해서도 사용이 가능하다. 맞춤형 프로그램은 인터넷 탐색 엔진에서 사용 가능한 것과 유사한 엔드유저 능력을 제공할 수 있다.

이상의 카테고리에 속하는 기술들은 SQL과 정확한 키워드 탐색의 한계를 극복하는 탐색능력을 가지고 있다. 이들은 또한 문맥에 민감한 탐색, 동의어, 어근 규칙들, 그리고 채점 기준을 사용함으로써 탐색의 효율성을 향상시키고 있다. 게다가 이들의 탐색 능력은 리스트 상에 가장 가깝게 매치된 것을 맨 처음에 올려놓고 관련도에 따라 순위를 매김으로써 탐색결과물의 효과를 증가시키고 있다. 이 기술들은 데이터베이스, 웹 페이지, 그리고 PC 텍스트 파일 문서에 있는 정보를 액세스한다.

자연언어 질의

SQL의 복잡성을 통제하려는 사용자의 요구를 해결하기 위한 한가지 접근방법은 자연언어 인터페이스를 SQL 언어에 제공하는 것이다. 링귀스틱 테크놀러지의 잉글리시 위자드는 이러한 제품 유형의 한 예이다. 링귀스틱 테크놀러지의 회장인 래리 해리스 박사는 자신이 개발한 인텔렉트라는 AI사의 메인프레임 자연언어 제품의 후속 제품으로서 잉글리시 위자드를 개발했다.

잉글리시 위자드는 영어 질문을 SQL 질의로 번역하기 위하여 의미론 계층을 사용하고 있다. 의미론 계층은 데이터베이스에 있는 정보에 영어 단어들의 관계를 정의하기 위하여 데이터베이스의 메타데이터로부터 구축된 전용의 사전을 사용하고 있다.

기본적인 영어 어휘들 뿐만 아니라 잉글리시 위자드의 사전은 어휘사전, 특수 날짜 지원, 데이터베이스 정의, 그리고 결합논리들을 포함하며, 이들을 목표 데이터베이스로부터 결정한다. 예를 들면, 기업 특유의 문구를 추가하거나 데이터베이스 뷰를 제한함으로써 딕셔너리 에디터(Dictionary Editor)를 사용하여 이 사전을 수정할 수 있다.

사전들은 계층적으로 존재하기 때문에 서로 다른 주제별로 된 사전이나 개인 사용자를 위한 사전 뿐만 아니라 관리자 사전도 있을 수가 있다. 관리자 사전에서의 변경은 계층에서 낮은 단계에 있는 다른 사전들로 자동적으로 반영이 된다.

사용자들은 데이터베이스에 질의를 영어로 표현하고, 그 질의는 사전을 이용하여 SQL로 번역된 후 데이터베이스로 넘겨진다. 사전이 질의를

번역할 수 없는 경우 잉글리시 위자드는 사용자에게 설명을 하라는 메시지를 표시한다. 번역이 되면 그 설명은 자동적으로 사전에 추가된다.

잉글리시 위자드의 사용자 인터페이스는 사용자가 직접 액세스하거나 마이크로소프트의 액세스, 씨게이트 소프트웨어 인포메이션 매니지먼트 그룹의 크리스탈 리포트, 플라티늄 테크놀러지의 포레스트&트리즈, 그리고 사이베이스의 인포메이커와 같은 ODBC 준수의 데스크탑 보고서 작성 툴들에 애드-인으로 설치될 수 있다.

결과는 스프레드시트 워크북이나 ODBC 준수 데스크탑 보고서 작성 툴들에 디스플레이될 수 있다. 다른 방법으로는 잉글리시 위자드 기술을 맞춤형 애플리케이션에 내장시키기 위한 한가지 소프트웨어 개발자 킷이 나와 있다. API들은 마이크로소프트 액세스, 오라클, 인포믹스 소프트웨어, 또는 애플리케이션 처리를 위한 SQL의 ODBC(IBM DB2와 그밖의 관계형 데이터베이스) 버전 안으로 사용자나 애플리케이션이 개발한 자연언어 요청을 번역하여 보낼 수 있다.

잉글리시 위자드는 OCX, VBX, 파워소프트(사이베이스의 자회사)의 파워빌더 유저 오브젝트, 볼랜드 인터내셔널의 델파이 컨트롤, 그리고 인포믹스의 뉴에라 클래스 라이브러리들을 지원한다.

역-인덱싱

텍스트 필드들을 탐색할 때 역-인덱스를 사용하여 관계형 데이터베이스의 B-트리 인덱싱의 일부 전통적인 한계를 해결할 수 있다. 관계형 데이터베이스 B-트리 인덱스 탐색은 왼쪽에서 오른쪽의 순서로 정확한 문자 매칭을 수행할 때 가장 효율적이다.

종종 성능상의 문제를 일으키는 B-트리 인덱싱의 단점은 다수의 자유 형식 텍스트 필드들에 대하여 일반화된 키워드 탐색을 수행하며, 데이터 필드 안에서 임의의 위치에 있는 탐색목표를 찾을 때 발견된다. 예를 들면, B-트리 인덱스를 사용하여 이름들을 찾는 것은 이름이 '성'과 '명' 열들에서 발견될 때 잘 작동한다.

B-트리 인덱싱은 특히 주소나 코멘트 필드와 같은 자유 형식 필드를 포함하여 이름이 필드 안에 있는 어떤 위치에서도 발견될 수 있을 때에는 잘 작동되지 않는다. 정확한 텍스트가 알려진 경우에도 종종 테이블 스캔이 필요하다. 비트맵 인덱스가 비록 B-트리 인덱스에 비해서 개선된 성능을 제공하긴 하지만 높은 기수성의 텍스트 데이터를 위해서는 일반적으로 적합하지 않다는 점을 유의해야 한다.

역-인덱스는 열 값(또는 텍스트의 경우에는 열에 있는 각 개별 단어)의 각 경우를 관련된 데이터베이스 행들에 대한 포인터와 함께 그 인덱스에 저장한다. 탐색은 인덱스에 대해서 행해지며, 앞에서 기술한 질의형식에 대해서 B-트리 인덱스나 베이스-테이블 스캔보다 월등하게 나은 성능을 가져다 준다.

다이나믹 인포메이션 시스템즈의 옴니덱스(Omnidex)는 키워드 탐색에 필요한 시간을 줄이기 위하여 역-인덱스를 사용한 제품의 예이다. 옴니덱스는 선택된 관계형 데이터베이스 열(들)에서의 각 단어를 색인에 넣어 역-인덱스로 만든다. 주제와 관계없는 단어들, 예를 들면 'a'나 'the'와 같은 단어들은 보통 배제된다.

사용자의 관점에서 볼 때 옴니덱스의 역-인덱싱 스킴은 표준 SQL을 넘어서는 탐색능력을 갖는다.

이러한 능력에는 테이블 스캔이 필요없는 임의 위치의 키워드 탐색, 사례 무감성, 그리고 음성 탐색의 경우 정확한 철자를 요구하는 대신에 사운덱스(Soundex)를 사용하는 능력 등이 포함된다.

'Robin'과 'Robyn'의 경우와 같이 비슷한 철자를 가진 이름들이 한번의 질의로 검색될 수 있다. 옴니덱스는 또한 색인의 유연성을 제공한다. 예를 들면, 여러 열들이 하나의 역-인덱스 그룹으로 만들어질 수 있다. 이러한 접근방법은 모두 비슷한 거리 주소 정보를 가지고 있기 쉬운 ADDRESS_LINE1, ADDRESS_LINE2, ADDRESS_LINE3과 같은 열들을 위해서 이득이 된다. 다른 방법으로는 한 열의 일부분만 인덱스에 포함시키는 방법도 있다.

데이터에 변경이 있으면 인덱스들은 실시간으로 업데이트되거나 애플리케이션 트랜잭션 로그 레코드들을 사용하여 백그라운드 모드에서 비동기적으로 업데이트되고, 간헐적인 배치 업데이트/재구축 프로세스를 이용하여 업데이트될 수 있다.

OAINSERT, OAUPDATE, OADELETE, OAINSERTION, OAUPDATEINDEX, OADELETEINDEX 등과 같은 애플리케이션 API 루틴들은 인덱스를 조작하기 위해 사용된다.

역-인덱스는 대규모 데이터베이스에 있는 구조화되지 않고 자유 형식을 가진 텍스트 데이터 열들을 위하여 가장 적합하다. 역-인덱스는 질의들이 레코드들의 많은 부분을 반환하지 않을 때 가장 효율적이다. 그렇지 않은 경우라면 테이블 스캔도 마찬가지로 효율적일 것이다. 역-인덱스는 엔드 유저 질의 능력에 상당한 개선효과를 가져다 주지만 대량 여분의 스토리지, 시스템 자원, 그리고 인덱스 업데이트 시간을 필요로 할 가능성이 있다.

인터넷 탐색 엔진

인터넷 상에 있는 방대한 데이터를 탐색할 수 있게 해주는 수많은 인터넷 탐색 엔진들에 관해서 이 책의 독자들이라면 아마 하나 이상은 알고 있을 것이다. 그러한 엔진들은 탐색능력을 제공하기 위하여 인덱스를 사용하는 제품들의 예이다.

이들 제품의 대부분은 '웹 스파이더(spiders)' 탐색 인터넷 웹 페이지들과 토의 그룹들을 가지고 작동하며, 각 페이지에서 키워드, 문구 혹은 개념을 찾아내고, 그 후에 단어들/문구들/개념들, 그리고 웹 페이지 URL을 하나의 대형 가상 인덱스에 자동적으로 추가한다. 그 인덱스는 '가상'이라고 불리우는데, 한 서버에 존재할 수도 있고 많은 서버에 나누어져 존재할 수도 있기 때문이다.

비록 그 최종 결과는 보통 같긴 하지만, 야후(www.yahoo.com)는 웹 페이지들을 검토하고 범주화하며 색인을 만들기 위하여 웹 스파이더를 사용하기 보다는 사람을 사용하는 탐색 엔진의 한 예이다. 개발과정에 관계없이 대형 인덱스들은 사용자가 키워드를 그 제품의 탐색 데이터-엔트리 필드에 입력할 때 탐색된다.

탐색이라는 관점에서 볼 때, 인터넷 탐색 엔진들은 개념과 내용 탐색을 이용하여 일반적으로 키워드 탐색 능력을 확장시키고 있다. 개념 탐색은 특정 탐색의 범위를 확대하기 위하여 탐색 기준에 있는 언어규칙들을 포함한다.

예를 들면, 'pants'라는 키워드를 사용하는 탐색은 '의류'와 '청바지'라는 개념을 포함할 수 있도록 확장될 수 있다. 다른 단어들은 관련된 개념이며 동의어가 아니다. 만약 'pants'가 숨을 헐떡이다라는 식으로 명사가 아닌 동사로 사용된다면 그러한 개념들이 '바지'에 연관되지 않도록 하기 위하여 문맥도 또한 고려될 필요가 있다.

인터넷 탐색 엔진들은 종종 관련도에 따라 정보의 순위를 매김으로써 검색 결과를 개선한다. 관련도는 검색된 데이터에 있는 단어의 빈도수, 상호간에 키워드와의 근사성, 그리고 데이터 안에서 그 키워드들의 위치 등과 같은 통계적 기법을 사용하여 결과들을 분석함으로써 일반적으로 구해진다.

많은 인터넷 탐색 엔진 벤더들은 인트라넷, 데이터베이스, 그리고 텍스트 문서들을 통합하기 위하여 서비스를 확장하고 있다. 탐색 엔진 기술은 다른 벤더들에 의하여 개발된 애플리케이션에 내장될 수 있는 경우가 종종 있다.

여기서는 베리티의 서치 '97과 엑스칼리버 테크놀러지의 리트리벌웨어(RetrievalWare)를 인터넷 탐색 엔진 제품의 예로 설명하였다.

베리티의 서치 '97은 웹 페이지, 토의그룹, 전자우편, 관계형 데이터베이스, 그리고 다른 데이터 소스들로부터 정보의 인덱싱과 추적을 할 수 있도록 설계되었다. 서치 '97은 정보 분배의 '풀'과 '푸시' 모델 모두를 지원한다.

서치 '97 아키텍처

베리티의 서치 '97 아키텍처는 몇 가지 핵심 컴포넌트들로 구성된다.

- 웹 브라우저 혹은 서치 '97 정보 서버 카탈로그들에 저장된 베리티의 퍼스날 '97 액세스 데이터
- 인포메이션 서버: 서치 '97의 정보 카탈로그들을 창조하고 관리하는 탐색 엔진. 한 기업 안에서 인포메이션 서버의 웹 스파이더는 새로운 혹은 변경된 정보를 카탈로그화하고, 그것을 서치 '97의 카탈로그에 추가한다.
- 에이전트 서버: 정보를 모니터하고 적합한 기준이 충족되면 적합한 명령어를 실행한다.

베리티는 보다 개선된 기능과 성능

으로 관계형 데이타베이스를 액세스할 수 있도록 64k(베리티가 최근 인수함)의 데이터베이스 액세스 기술을 서치 '97의 에이전트-서버 계층에 통합하고 있는 중이다. 인터넷/인트라넷 웹 페이지와 텍스트 액세스는 계속 베리티의 기존 제품에 의해 처리될 것이다. 64k의 관계형 데이터베이스 관련도 순위화는 이에 해당되는 베리티의 텍스트 능력으로 결합될 것이다. ODBC는 데이터베이스들을 액세스하기 위해 사용할 예정이다.

64k의 DB가이드 제품에 있는 핵심적인 특징은 반복적인 탐색 과정의 시작으로 전체 데이터베이스 내용들의 개요 혹은 부집합을 보여주는 능력이다. 사용자들은 데이터베이스에 무엇이 들어있는지를 발견하기 위하여 많은 질의를 하지 않아도 된다. 퍼지 질의 예문(fuzzy query-by-example) 기능은 사용자가 한 레코드를 선택할 수 있도록 해주고, DB가이드에게 '내가 선택한 레코드와 비슷한 레코드들을 보여라'라고 말하면 된다.

DB가이드는 특허출원된 '인텔리전트 인덱스' 기술을 사용한 것이다. 이 인덱스는 관계형 데이터베이스 스키마로부터 구축된다. 시스템 관리자는 이 인덱스를 창조하기 위하여 데이터베이스 위치와 인덱스 업데이트 빈도만 지정하면 된다. 전화번호, 우편번호, 그리고 URL과 같은 첨단 데이터 타입들도 제공될 것이다.

리트리벌웨어 아키텍처

엑스칼리버의 리트리벌웨어 아키텍처는 텍스트 검색 및 프로파일링 서버, 비주얼 미디어 서버, 그리고 하나의 웹 서버로 구성된다. 엑스칼리버는 데이터 갈무리, 인덱싱, 탐색, 그리고 검색을 수행하기 위하여 자사의 APRP(Adaptive Patern Recognition Processing)와 의미론 네트웍 기술들을 사용한다.

APRP 탐색 기능의 사용은 특정 키워드가 아니라 신경망과 같은 모델들에 기반을 두고 있다. 인덱스들은 디지탈 텍스트 데이터에 있는 이진 패턴들을 이용하여 창조된다. 이 기법의 한가지 장점은 수작업을 통해 키워드를 정의하지 않아도 되게 해준다는 점이다. 또 다른 장점은 이것이 퍼지 탐색을 지원하는 것이다.

엑스칼리버의 의미론 네트웍은 사전과 어휘사전들에 있는 단어들의 의미와 관련성에 근거하여 개념 탐색을 수행한다. 사용자는 질의에 사용된 단어의 의미를 선택할 수 있고, 관련도를 결정하기 위하여 사용된 단어의 의미를 디스플레이할 수 있다. 여기에는 40만개의 개념과 160만개의 단어가 포함되어 있다고 한다.

데이터베이스 텍스트 탐색

객체-관계형 데이터베이스의 애플리케이션 확장성과 유연성은 데이터베이스에 저장된 텍스트에 대한 탐색 능력을 현저하게 향상시킨다. PLS(Personal Library Software) 퍼스날 라이브러리언 제품의 핵심적인 기능성은 인포믹스의 유니버설 서버를 위해서 인포믹스와 PLS 텍스트 데이터블레이드로 통합되어 왔다.

IBM의 텍스트 익스텐더는 DB2 데이터베이스에 있는 정보를 대상으로 탐색한다. 썬더스톤 소프트웨어의 관계형 데이터베이스는 인터넷/인트라넷 텍스트 스토리지와 검색 애플리케이션들을 위해 구축되었다.

PLS 텍스트 데이터블레이드는 CPL(Callable Personal Librarian) 모듈과 객체 기반의 C 라이브러리 API들의 집합에 기반을 둔 것이다. CPL은 텍스트 처리, 질의, 관련도 순위화, 그리고 텍스트 검색을 수행한다. 개념 탐색은 그 질의에 있는 용어들과 관련된 용어를 포함하도록 질의들을 동적으로 확장함으로써 수행된다.

PLS 텍스트 데이터블레이드 아키텍처에 기본이 되는 것은 인덱스의 용어들이 나오는 데이터베이스의 열들과 함께 인덱스의 용어를 교차참조하는 역-인덱스이다. PLS 인덱스에서는 사전을 사용하여 인덱스 구조를 유지한다.

사전은 다음과 같은 방식으로 창조된다.

- 텍스트 데이터의 토큰화(Tokenizing Textual Data) : 데이터 토큰들은 단어, 단어가 아닌 텍스트 줄, 수치, 그리고 스페이스와 같은 구분문자들을 참작하는 의미있는 문자들의 그룹이다.
- 법전화(Canonizing) : 모든 용어를 일관된 형식안에 넣는다. 예를 들면, 모든 단어들은 더 낮은 쪽의 케이스로 변환된다.
- 정지단어 필터링(Stopword Filtering) : 예를 들면 'a'와 'the' 같은 거의 내용이 없는 단어들을 제거한다.
- 가지떼기(Stemming) : 단어의 접미사를 떼내어 탐색중 매칭작업을

촉진해준다.

인덱스들은 열들과 연관되며 텍스트의 용어들을 인덱스에 추가하는 CREATE INDEX...USING pls(열)라는 명령을 통해서 창조된다. 인덱스가 창조된 후에는 모든 유니버설 서버 명령에 의한 인덱스된 데이터베이스 열로의 변경은 인덱스에 반영되게 된다. 다음과 같은 데이터 타입들이 지원된다.

- Text. 길이가 약 8K 정도까지의 가변적인 문자 스트링
- Large_text. 약 8K보다 큰 텍스트 스트링
- Large_object. 그 위에 또 쓰거나 추가해서 쓸 수 없는 '한번만 쓰는' 객체들
- External File. 일러스트라나 다른 애플리케이션들에 의해 액세스할 수 있는 파일들

관련도 순위화를 제공하기 위하여 PlsOidRank_t 데이터 타입이 PLS 텍스트 데이터블레이드 설치 중에 CREATE TEYP PlsOidRank_t (pls_oid oid, pls_rank integer) 명령에 의해서 창조된다.

IBM DB2 텍스트 익스텐더는 DB2에 저장되어 있거나 DB2가 관리하지만 일반적인 파일이나 PC 문서 형식으로 존재하는 텍스트 데이터를 대상으로 언어적 탐색을 수행한다. 와일드카드 탐색, 근사성 탐색, 언어적 탐색, 그리고 가지 탐색 등은 사용가능한 탐색기능들의 예이다.

SQL은 텍스트 탐색을 수행하기 위하여 사용될 수 있다. 'TextTable'이라는 DB2 테이블의 DB2COL 열에 색인된 텍스트의 동일한 단락에서 'internet', 'text', 그리고 'search (혹은 이와 동의어)' 등과 같은 단어를 찾기 위하여 다음과 같은 SQL이 사용될 수 있다.

```
SELECT * FROM TextTable WHERE
DB2TX.CONTAINS (DB2COL,
"internet" IN SAME PARAGRAPH AS
"text" AND SYNONYM FORM OF
"search") = 1
```

(DB2TX.CONTAINS는 DB2 텍스트 익스텐더의 탐색 기능의 하나이다.)

한 텍스트 열에 대하여 세가지 유형의 인덱스를 창조할 수 있다.

- 프리사이즈 : 문서에 있는 각 용어가 그대로 인덱스에 저장된다.
- 링귀스틱 : 색인되는 텍스트를 분석하는 동안 언어적 처리가 수행된다.
- 듀얼 : 위의 두가지 타입의 탐색을 결합한 것으로 양자 모두를 지원한다. STEMMED FORM OF와 PRECISE FORM OF의 옵션은 원하는 탐색의 타입을 확인하기 위해 사용될 수 있다.

프리사이즈 인덱스는 정확한 매치를 원할 때 사용하기 가장 좋은 것이다. 이것은 일반적으로 가장 빠른 탐색 타입이다. 'leaf'라는 단어에 대한 프리사이즈 탐색은 이 단어와 똑같은 결과만을 반환할 것이며, 'leaves'와 같은 변형은 결과로 나오지 않는다. 마스크와 와일드카드의 사용이 가능하다. 링귀스틱 인덱스의 가장 핵심적인 장점은 탐색 단어의 변형들이 자동적으로 매치된다는 것이다. 'leaf'의 탐색은 'leaf'와 'leaves' 모두를 포함하는 문서들을 반환할 것이다. 듀얼 인덱스는 스토리지를 가장 많이 필요로 한다. 인덱스들은 열별로 창조될 수도 있고, 한 데이터베이스에 있는 모든 텍스트 열들에 대한 전역의 것으로 창조될 수도 있다.

DB2 텍스트 익스텐더는 최대의 성능을 발휘하기 위하여 DB2와 공유 주소공간을 가지도록 설치될 수도 있고, 자신만의 주소공간을 갖도록 설치될 수도 있다. 텍스트 익스텐더는 IBM의 DB2 유니버설 데이터베이스로 병합될 코드 베이스의 일부이다. 이 제품은 17종의 언어를 지원한다.

썬더스톤의 텍시스 관계형 데이터베이스는 그 데이터베이스나 외부 파일들(따라서 INDIRECT 데이터 타입을 사용함)에 저장된 텍스트 정보를 탐색할 수 있도록 하기 위하여 소위 '거친 그레인의 역-인덱스'를 사용하고 있다.

인덱스는 SQL CREATE INDEX 명령어로 창조되며, 일반적으로 10-15%의 추가적인 스토리지를 필요로 한다. 성능을 향상시키기 위하여 인덱스를 기초가 되는 데이터와 별도의 스토리지 디바이스에 저장할 수 있다.

사용자가 직접 텍시스를 액세스할 수 있기는 하지만 이 데이터베이스는 대개의 경우 몇개의 전용 키워드를 가진 SQL을 사용하는 프로그램들에 의해서 액세스된다. 절단, 와일드카드, 근사성 퍼지 매칭, 그리고 개념 탐색 기능 등의 사용이 가능하다.

결과는 관련도에 따라 순서가 매겨질 수 있다. 질의 결과는 반복적인 탐색을 위하여 임시적인 테이블에 저장될 수 있다.

핵심적인 SQL LIKE 문장 확장은 메타모프(Metamorph) 구문을 지원함으로써 썬더스톤의 메타모프 탐색 엔진의 기능을 포함한다. 따라서 LIKE 'text internet w/para'는 'text'와 'internet'이라는 단어가 BODY_TEXT 열내에서 동일한 단락에 있는 모든 기사의 저자를 검색하게 된다. '/w'는 메타모프 근사성 키워드이다. 다른 근사성 키워드 파라미터들은 근사성을 한 문장 안이나 특정 수 이내의 단어로 한정하는데 사용될 수 있다. 동의어 능력도 또한 사용할 수 있다.

LIKER SQL 확장 키워드는 요청된 용어의 출현빈도에 근거한 관련도의 순서에 따라 데이터를 반환하게 된다. LIKEP는 근사성 탐색을 사용하게 된다.

텍시스 데이터베이스 텍스트 탐색 기반구조의 상당부분은 맞춤화할 수 있다. 25만 단어의 어휘사전, 정지-단어 리스트(인덱스의 필요가 없는 잡음 같은 단어들), 그리고 이 제품과 함께 배포되는 '개념' 데이터베이스 등 모두가 사용자에 의해 업데이트될 수 있다. 인터넷과 인트라넷 애플리케이션은 텍시스의 웹스크립트 언어를 사용하여 창조할 수 있다.

다른 데이터베이스 탐색

오브젝트 디자인사는 키워드 매칭, 개념 탐색, 관련도 순위화, 그리고 그 밖의 베리티 텍스트 탐색 능력들을 제공하기 위하여 베러티의 토픽 서치 엔진과 토픽 디벨로퍼 킷에 기반한 오브젝트스토어 텍스트 오브젝트 매니저를 개발했다. 텍스트 문서는 카탈로그화되고 각자의 원래 형식 그대로 저장되며, 저자, 수정일자, 토픽, 그리고 컨텐트 타입 등과 같은 애트리뷰트들에 따라 캡슐화할 수 있다. 그 밖의 애트리뷰트들은 필요한 경우 추가할 수 있다.

데이터웨어 테크놀러지사의 인포매그넷((InfoMagnet) 기술은 로터스 노츠를 포함하는 수많은 문서들과 다른 형식들로부터 정보를 걸러낸다. 매그넷리더는 데이터를 동적으로 읽고, 인덱스를 만들고 사용자에게 분배하기 위해 실시간으로 데이터를 걸러낸다. 매그넷스위퍼는 정적인 웹 또는 파일 서버 데이터를 읽고, 인덱스를 만들고 걸러낸다. 인포매그넷은 HTML과 자바스크립트를 가지고 맞춤화할 수 있다. 자연언어 사용자 인터페이스도 또한 포함되어 있다.

인포매그넷은 키워드 가중치 부여, 개념 탐색, 그리고 관련도 순위화 기법을 지원한다. 인포매그넷이 사용자의 프로파일 안으로 더욱 세밀한 정도의 판별자료를 제공할 수 있도록 사용자는 검색된 문서 중 어느 것이 관련된 것이고, 어느 것이 관계없는 것인지를 지시할 수 있다.

데이터베이스들은 텍스트와 숫자보다도 탐색과 검색이 훨씬 더 복잡한 디지탈 데이터를 저장하기 시작했다. 데이터베이스 벤더와 써드파티들이 이제는 각 컨텐트에 관한 텍스트 탐색 용어들에 제한되는 것이 아니라 디지탈 컨텐트 그 자체의 탐색을 지원할 수 있다.

인포믹스와 비라지(Virage)의 비주얼 인텔리전스 리트리벌 데이터블레이드는 인포믹스 유니버설 데이터베이스에 대한 비주얼 및 텍스트 탐색을 수행한다. 비주얼 인텔리전스 뷰어(VIV)는 그래픽 뷰어 데이터블레이드이다. VIV를 이용하여 탐색을 하려면 탐색할 그래픽의 템플레이트를 지정한 후에 탐색에서의 그래픽 특징의 중요성을 지시하기 위하여 슬라이더 컨트롤을 사용한다. 그래픽 이미지들과 이와 관련된 텍스트 필드를 함께 탐색할 수 있다. 일반적으로 사용되는 이미지 포맷들을 지원하며, 텍스트, 오디오, 비디오, 그리고 다른 탐색 데이터블레이드의 사용이 가능한다.

IBM의 DB2 이미지 익스텐더는 대형의 객체들을 위해 DB2 버전 2의 최대 크기인 2기가바이트까지의 이미지를 지원하는 이미지 데이터타입과 기능을 제공한다. QBIC(Query by Image Content) 능력은 특정한 색과 질감을 가진 이미지를 탐색할 수 있게 해준다. 질의는 이미지와 관련된 이미지 및 텍스트 정보 모두 포함할 수 있기 때문에, 예를 들면 '빨간색'을 포함하며 1996년 이전에 찍은 사진이라고 선택할 수 있다. GIF, JPEG, BMP, TIFF와 같은 이미지 포맷들이 지원된다. DB2 텍스트, 오

디오, 비디오, 그밖의 다른 탐색 확장자의 사용이 가능하다.

비주얼 리트리벌웨어는 엑스칼리버의 디지탈 이미지 탐색 및 검색 제품이다. 이미지에 관한 형상, 색깔, 그리고 질감 정보를 얻고 그것은 일반적으로 원래 이미지 크기의 1-10%인 탐색가능한 인덱스에 추가된다.

한 소프트웨어 개발회사의 킷이 리터럴(literal) 이미지 컨텐트 매칭을 사용, 인덱스를 작성하고 탐색을 수행할 수 있는 소위 그림 벡터(feature vector)로 이미지를 압축하는 기능을 포함하여 비주얼 리트리벌웨어에 대한 'C' API 프로그램 인터페이스를 제공하고 있다.

이와 관련하여 그림 벡터를 가진 인덱스와 실제의 이미지를 가진 데이터베이스라는 두가지의 데이터베이스가 있다. VBX들과 관계형 데이터베이스 DLL들은 윈도우 95와 NT 플랫폼 상에서 제공된다. 엑스칼리버는 얼굴 인식의 매칭을 수행하는 얼굴 인식 인포믹스 유니버설 서버 데이터블레이드를 개발했다. SQL 질의 인터페이스를 이용하여 이미지에 대한 질의를 할 수 있다.

탐색 기술의 절충

자연언어 질의, 역-인덱싱, 인터넷 탐색 엔진, 그리고 데이터베이스 텍스트 탐색 등의 기능은 수많은 상황속에서 유용성을 발휘하고 있지만, 또한 이와 함께 고려해야 할 사항들이 있다.

우선, SQL 질의를 배제해서는 안된다는 점이다. 최종 사용자에게 포인트-앤-클릭의 SQL 프론트엔드 툴을 제공하는 것은 사용자의 보고서 작성 요구를 만족시킬 수도 있기 때문이다. 이곳에서 검토한 탐색 기술들은 대규모의 자유 형식 데이터 안에서 탐색을 수행할 때에는 최상의 것이다. 대상 데이터가 구조화된 것이라면 첨단 탐색기능이 전혀 불필요할 수도 있을 것이다.

비록 자연언어 질의가 개념적으로는 매력적인 것이긴 하지만 실제로 채용된 적은 없는데, 이는 아마 영어의 복잡성으로 인해 이 기술이 100% 정확성을 가진 질의를 구현할 수 없었기 때문인 것으로 보인다. 게다가 관리를 위한 셋업과 데이터베이스 용어의 영어 번역에 필요한 지식과 시간 또한 과소평가될 수 없다. 잉글리시 위자드와 같은 최신의 자연언어 질의툴들은 이러한 제한요인을 극복할 수 있도록 설계되었다.

역-인덱싱은 질의 성능을 현저하게 향상시킬 수 있다. 그러나 이 개선된 성능은 그 인덱스를 저장하는데 필요한 디스크 비용과 인덱스를 실시간이나 배치모드에서 업데이트하는데 필요한 추가적인 처리시간을 대가로 얻어지는 것이다.

인터넷 탐색 엔진은 현재 인터넷이나 인트라넷 웹 페이지와 토의그룹들에 대하여 가장 잘 작동하고 있다. 벤더들이 기업 데이터베이스의 탐색도 지원하기 시작했지만 인터넷 탐색 엔진을 관계형 데이터베이스에 질의하기 위하여 사용할 경우에는 사용자 인터페이스, 인덱싱, 그리고 탐색 수행 경험 등 많은 것들이 부족한 상태이다.

이와 유사하게 텍스트 탐색의 역사나 데이터베이스에서 사용 가능한 다양한 디지탈 컨텐트가 풍부한 것이 아니다. 그러나 보편적인 데이터베이스 구조와 데이터블레이드나 익스텐더를 창조하는 능력은 특히 텍스트 객체들에 대한 탐색을 포함하여 탐색 능력을 향상시켜가는데 적합하게 설계된 것으로 보인다.

SQL을 넘어서

데이터 소스의 수와 규모가 증가함에 따라 사용자들이 효과적인 특수한 질의와 탐색을 수행하는 작업이 갈수록 어려워지고 있다. 몇가지 기술들이 SQL 키워드 및 문자 스트링 탐색을 훨씬 능가하는 능력을 제공하고 있다. 자연언어 질의, 역-인덱싱, 인터넷 탐색 엔진, 그리고 보편적인 데이터베이스 탐색 등이 바로 그것이다.

이런 기술을 사용하는 제품들은 일반적으로 SQL이 제공하지 않는 사전, 어휘사전, 의미론 규칙 개념 탐색, 고성능 인덱싱, 그리고 관련도 순위화 기능 등을 사용하고 있다. 웹 페이지와 마이크로소프트 워드 문서 등과 같은 데이터베이스가 아닌 소스들도 지원하고 있다.

엔드유저들이 사내에 있는 정보를 효과적으로 검색할 수 없을 때, 어떤 형태의 탐색 기술을 구현할 것인가를 생각해야 할 것이다. 효과적인 탐색 기술은 데이터베이스 관리자 및 사용자의 시간을 절약해줄 것이며, 다른 방법으로는 찾아낼 수 없었던 귀중한 정보를 제공할 것이다. DC