퍼지관계곱 기반 퍼지정보검색시스템 구현

김 창 민† · 김 용 기††

요 약

퍼지관계 개념에 기반한 BK-FIRM(Bandler-Kohout 퍼지정보검색기법)은 형태론에 입각한 기존의 정보검색기법과는 달리 문서와 용어의 상대적 의미에 근거한 퍼지정보검색기법이다. BK-FIRM은 시소러스 자동 구축 기능, 검색 결과의 퍼지화된 우선 순위 제공과 같은 장점을 가지고 있다. 그러나, BK-퍼지정보검색기법은 높은 시간복잡도(time complexity)의 검색 연산을 내재하고 있어 다양한 분야 적용이 불가능하다. 본 논문에서는 축소용어집합을 이용하여 BK-FIRM의 시간복잡도를 낮춘 A-FIRM(개선된 Bandler-Kohout 퍼지정보검색모델)을 소개하고 이를 정보검색시스템으로 설계 및 구현한 A-FIRS(개선된 Bandler-Kohout 퍼지정보검색시스템)를 구현한다. A-FIRS는 크게 문서베이스와 시소러스를 구축하는 전처리부(preprocess unit)와 사용자의 검색요구를 처리하여 문서를 검색하는 실시간처리부(real-time process unit)로 나누어지며, 각 처리부는 기능적 특성에 따라 4개의 처리단계로 구성된다. A-FIRS는 WWW 기반 환경과 연동하도록 설계되었으며, WWW 환경의 사용자로부터 주어진 검색요구를 처리하여 검색결과를 제공한다.

Implementation of Fuzzy Information Retrieval System Based on Fuzzy Relational Products

Chang-Min Kim† · Yong-Gi Kim††

ABSTRACT

BK-FIRM (Bandler and Kohout's fuzzy information retrieval model) suggested by Bandler and Kohout, uses the concept of fuzzy relation, and it is able to retrieve documents in the way based on not morphology but semantics, dissimilar to traditional information retrieval theories. It also has merits such as building thesaurus automatically and providing fuzzy ranking system. BK-FIRM, however, is not able to apply so many domains because its operations have very high time complexity. In the paper, we introduce A-FIRM (Advanced Bandler-Kohout fuzzy information retrieval model) which improves time complexity of BK-FIRM by using reduced term set, and present A-FIRS (Advanced Bandler-Kohout fuzzy information retrieval system) which is the information retrieval system based on A-FIRM. A-FIRS consists of two parts, preprocess unit and real-time process unit, The first builds the document base and the thesaurus, the last analyzes user requests and retrieves documents. Each process unit consists of 4 procedures. A-FIRS is designed to work with WWW environment, and so it offers retrieved documents, when a user give a query through WWW.

키워드 : 정보검색시스템(Information Retrieval System), 퍼지정보검색(Fuzzy Information Retrieval), 퍼지관계곱(Fuzzy Relational Products)

1. 소 개

지구상 곳곳에서 생활하는 수십 억의 인류는 자신만의 전문화된 영역에서 새로운 정보를 쉼 없이 생산한다. 국지적으로 생산된 정보는 발달된 통신 수단으로 그 지역성이 극복되어 수많은 사람들과 공유된다. 특히 극도로 발달된 통신매체인 인터넷의 발달에 기인하여 우리는 바야흐로 다양한 분야에서 방대한 양의 정보가 생산되고 있는 정보화 시대에 살고 있다고 말할 수 있다. 그러나, 넘쳐나는 정보의 홍수로 휩싸인 현 사회는 정보를 수집, 분류, 습득하여야 하는 현대인에게 엄청난 부담으로 작용한다. 따라서 현대인에게 요구되는 능력 중 가장 중요한 것은 정보의 바다에서 자신이 원하는 정보를 정확하고 빠르게 검색하는 것이다. 그러나 쉼 없이 쏟아지는 정보로부터 원하는 정보만을 한정된 시간 내에 검색하는 것은 쉬운 작업은 아니다. 1960년대 초, 컴퓨터를 이용하여 원하는 정보를 제한된 시간 내에 검색하고자 하는 정보검색(information retrieval) [1] 이라는 학문분야가 태동하였다. 정보검색의 초기 목적은 도서관의 방대한 량의 문헌을 효과적으로 관리하는 것이었다. 그러나, 20세기 후반 등장한 인터넷의 영향으로 인

※ 본 연구는 1998년 과학재단의 핵심전문연구(과제번호 981-0919-102-2)에 의해 수행되었습니다.
† 준 회 원 : 경상대학교 대학원 컴퓨터과학과
†† 종신회원 : 경상대학교 컴퓨터과학과 교수
논문접수 : 2000년 12월 7일, 심사완료 : 2001년 2월 24일

터넷 상의 문서검색이 중요한 연구과제로 부각되었다. 정보검색은 문헌검색 뿐만 아니라 소프트웨어 공학에서도 아주 유용하며, 최근 부상하고 있는 게놈(Genome) 사업에도 적절히 적용될 수 있다.

정보검색은 오랜 역사만큼 많은 연구가 있었다. 수많은 연구자에 의해 다양한 기법이 제안되었고 각 기법들은 개념모델, 화일구조, 질의연산, 용어연산, 문헌연산, 하드웨어와 같은 세부 연구 분야로 구분되어 체계적인 모습을 갖추게 되었다[18]. 정보검색의 개념모델에서 살펴볼 때 근간을 이루는 검색모델은 불리언식으로 표현된 질의어를 이용하여 정보를 검색하는 불리언검색모델이다. 불리언 검색모델은 구현이 용이하고 질의어의 처리 시간 면에서 효율적이기 때문에 가장 널리 쓰이고 있는 검색 모델이다. 불리언 검색 모델은 적절한 질의어가 입력되면 조회율과 정확도 면에서 좋은 성능을 보인다. 그러나 질의어의 엄격한 해석, 검색 결과 우선 순위에 대한 대비책 부재, 색인 결정 시 존재하는 불확실성에 대한 대비책 부재와 같은 검색효율의 한계성을 드러낸다. 따라서 불리언 검색모델을 개선하여 탐색결과의 정확도와 조회율을 향상시키기 위한 다양한 모델이 제안되었다. 대표적인 검색모델로는 퍼지집합론에 근거한 Fox와 Sharat의 MMM(Max Min and Max) 모델[2], Paice의 Paice 모델[3], 정규화된 용어와 역문헌의 빈도수 통계치를 이용한 Fox의 P-norm 모델[4], 퍼지집합론과 퍼지관계곱을 이용한 BK-FIRM(Bandler와 Kohout의 퍼지정보검색모델, Bandler and Kohout's fuzzy information retrieval model)[5]이 있다.

Bandler와 Kohout는 Rijsbergen의 통찰력 있는 관점[6-7]에 근거하여 불리언 정보검색모델을 확장한 BK-FIRM을 제안하였다. BK-FIRM은 시소러스 자동 구축 기능, 검색 결과의 퍼지화된 우선 순위 제공, 직접 관련 없는 제3의 개체 유추 검색 등과 같은 특성을 가진다.

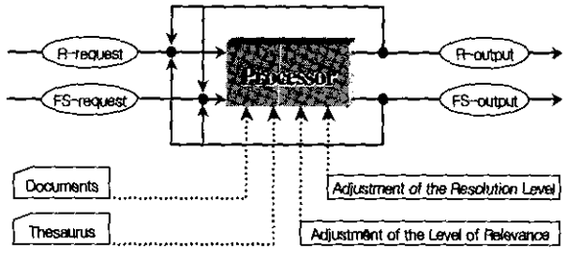
본 연구에서는 축소용어집합을 이용하여 BK-FIRM의 높은 시간복잡도를 낮추는 A-FIRM(개선된 BK-퍼지정보검색모델, Advanced Bandler and Kohout's fuzzy information retrieval model)을 WWW 환경의 문서검색시스템에 적용한 A-FIRS(개선된 Bandler-Kohout 퍼지정보검색시스템, Advanced Bandler and Kohout's fuzzy information retrieval system)의 설계와 구현에 관한 연구이다. 본 연구에서 구현하는 A-FIRS는 소규모 문서를 대상으로 한 검색하는 프로토타입 수준의 검색시스템이다.

제2절에서는 A-FIRM의 근간을 이루는 BK-FIRM을 소개하고, 제3절에서는 A-FIRM에 관하여 살펴본다. 제4절에서는 A-FIRS 구조에 관하여 살펴보고, 제5절에서는 A-FIRS의 구현과 결과에 관하여 살펴본다. 제6절에서는 연구의 결론에 관하여 살펴본다.

2. BK-FIRM

Bandler와 Kohout의 BK-퍼지정보검색모델은 형태론에 입각한 기존의 정보검색기법과는 달리 문서와 용어의 상대적 의미를 표현하는 퍼지관계와 퍼지관계곱을 이용하는 정보검색기법으로서 자동 시소러스(thesaurus) 구축기능과 검색결과의 퍼지화된 우선 순위 제공과 같은 기능을 기본적으로 가지고 있다[11].

BK-FIRM은 문서집합과 용어집합을 정의하고 문서와 용어의 상대적 의미를 문서와 용어의 퍼지관계행렬로 표현한다. BK-FIRM은 퍼지관계행렬에 퍼지관계곱 연산을 적용하여 시소러스(thesaurus)를 형성하고 사용자로부터 주어진 질의어를 해석하고 시소러스를 이용하여 확장한 후 퍼지관계를 이용하여 문서를 검색하는 퍼지검색요구(FS-request) 연산과 시소러스를 이용하여 주어진 용어의 의미를 확장하는 관계요구(R-request) 연산을 제공한다[6-11]. (그림 1)은 BK-FIRM의 모형도이다.



(그림 1) Bandler-Kohout 퍼지정보검색모델

3. A-FIRM

BK-FIRM은 높은 시간복잡도(time complexity)를 가지는 검색연산을 내재하고 있어 적용분야에 많은 제약이 따른다. A-FIRM은 축소용어집합(reduced term set)을 이용하여 BK-퍼지정보검색모델 시간복잡도를 개선한 검색모델이다[19].

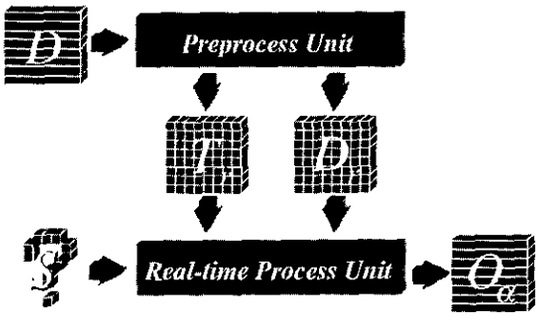
A-FIRM은 문서집합, 용어집합, 문서집합과 용어집합과의 퍼지관계를 정의하고 이로부터 축소용어집합, 문서집합과 축소용어집합과의 퍼지관계와 시소러스를 산출한다. 사용자로부터 질의어가 주어지면 질의어를 해석하고 시소러스와 합성하여 확장된 질의어를 산출한다. 이후 퍼지관계곱을 이용하여 검색결과를 얻고 $\alpha-cut$을 적용하여 최종 문서 검색결과를 구한다[19].

4. A-FIRS

A-FIRS는 A-FIRM을 기반으로 한 문서검색시스템이다. 시스템 검색성능 향상을 위하여, A-FIRS는 문서베이스와 시소러스와 같은 기초자료를 구축하는 전처리부(Preprocess

Unit)와 사용자의 검색요구를 처리하여 문서를 검색하는 실시간처리부(Real-time Process Unit)로 구분된다. 전처리부는 문서 검색을 위해 문서베이스와 시소러스를 구축한다. 전처리부는 처리시간에 거의 구애받지 않으며 처리빈도도 매우 낮아 오프라인(off-line) 형태의 작업에 적합하다. 실시간처리부는 검색식이 주어졌을 때 문서베이스와 시소러스를 이용하여 검색식에 적합한 문서를 검색 출력한다. 실시간처리부는 정해진 반환시간(turn-around time) 내에 검색결과를 출력해야하므로 온라인(on-line) 작업 형태로 이루어져야 한다.

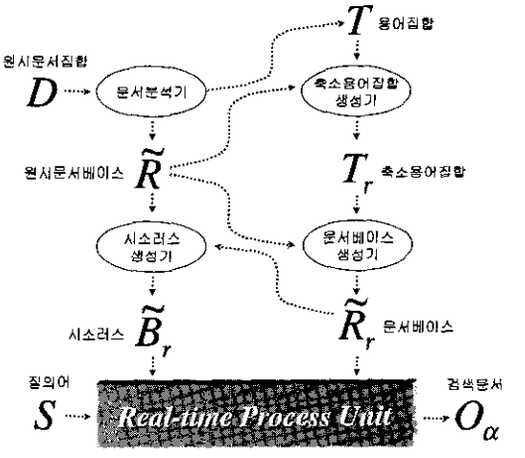
(그림 2)는 A-FIRS의 시스템모형도이다. (그림 2)에서 원시문서집합 D는 문서의 집합을 의미하고 문서베이스 $\tilde{R}_r$는 문서와 용어의 축소집합과의 퍼지관계로서 검색요구 시 문서 검색을 위한 구조체이다. 시소러스 $\tilde{B}_r$는 용어와 축소된 용어와의 관계를 유지하는 퍼지관계로서 자체에 용어의 계층구조를 내재하고 있어 문서검색 시 질의어 확장을 위한 관계요구의 구조체이다. 질의어 S는 사용자의 검색요구를 표현한 검색식이고 검색결과 O_α는 검색된 문서 목록을 의미한다.



(그림 2) A-FIRS 시스템모형도

4.1 전처리부

전처리부는 A-FIRS의 문서 검색에서 선 처리되어야하는 시소러스와 문서베이스를 생성한다. (그림 3)은 전처리부를 구성하는 4개의 처리기를 보여준다. 문서분석기(Document Analyzer)는 원시문서집합을 입력받고 용어집합 T를 추출하여 문서와 용어의 관계 원시문서베이스 $\tilde{R}$을 생성하는 처리기이다. 축소용어집합생성기(Reduced Term-set Generator)는 원시문서베이스 $\tilde{R}$을 이용하여 용어집합 T에서 축소용어집합 T_r를 생성한다. 문서베이스생성기(Document-base Generator)는 축소용어집합 T_r를 참조하고 원시문서베이스 $\tilde{R}$을 변환하여 문서베이스인 문서와 축소된 용어집합과의 관계 $\tilde{R}_r$를 생성하는 처리기이다. 시소러스생성기(Thesaurus Generator)는 원시문서베이스 $\tilde{R}$와 문서베이스 $\tilde{R}_r$에 퍼지관계곱 연산을 적용하여 용어간 계층구조를 표현하는 시소러스인 용어와 축소용어집합과의 관계 $\tilde{B}_r$을 생성한다.



(그림 3) A-FIRM의 전처리부

4.1.1 문서분석기

문서분석기는 원시문서집합을 가공하여 용어집합과 원시문서베이스를 생성한다. 문서분석기는 먼저 원시문서베이스와 용어집합을 초기화하고 원시문서집합 내에 포함된 모든 문서에 대해 반복적으로 처리한다. 문서분석기는 우선 원시문서집합으로부터 문서를 인출한 후 문서의미와 관련성이 적은 HTML 태그를 제거하고 용어를 분리한다. 그리고 스테머(stemmer)를 이용하여 용어의 어근을 추출하고 가용어 목록을 이용하여 용어를 선별한다.

```
1 : D : document base
2 : D_org : original document base
3 : T : term set
4 : d_word : terms
5 : d : document
6 :
7 : BEGIN
8 :     initialize(D_org)
9 :     initialize(T)
10 :    REPEAT FOR D_org
11 :        d = fetch_a_document_from(D_org)
12 :        remove_HTML_tag(d)
13 :        d_word = separate_terms_from(d)
14 :        stemming(d_word)
15 :        select_and_update_available_words(d_word)
16 :        sort(d_word)
17 :        calcurate_frequency_of_term(d_word)
18 :        fuzzify_the_document(d_word)
19 :        store_result_to(D)
20 :    END REPEAT
21 : END
```

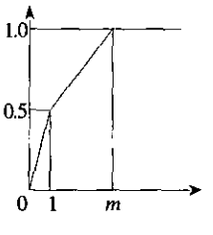
(그림 4) 문서분석기 처리절차

스테머는 스테밍을 자동으로 처리하는 도구를 일컫는데, 스테밍이라 함은 단어들을 공통 어근 형태로 감소시킴으로써 관련된 단어를 자동으로 융합(분해 및 결합)하는 것을 말한다. 스테머는 크게 접사 제거 스테머, 후속자 변형 스

테머, 테이블 탐색 스테머, n-gram 스테머[19]로 분류할 수 있다. 본 시스템에서는 접사 제거 스테머 중 최장 대응 스테머에 속하는 Porter 알고리즘[20]을 이용하였다. Porter 알고리즘은 타 기법보다 작고 간결하고 비교 검색 성능이 우수하다고 알려져 있다[2].

이후 각 용어들의 어근들을 정렬하고 이를 이용하여 어근에 대한 빈도를 추출한다. 마지막으로 빈도를 이용하여 해당 문서 용어집합에 관한 상대적 관련성을 간접적으로 추정하고 이를 원시문서베이스에 저장한다. 여기서 유의할 점은 문서를 용어집합에 대한 문서의 상대적 관련성을 퍼지화하는 어떠한 방법도 정해져 있지 않다는 것이다. 일반적으로 두 가지 방법으로 나눌 수 있는데, 첫째는 지적 능력을 가지는 사람이 직접 분석하는 방법이다. 이는 가장 신뢰할 수 있는 결과를 이끌어 낼 수 있으나 문서의 양이 많을 때는 상당한 시간과 비용이 요구되며 다수 인원의 참여 시 개인의 지적 배경의 차로 인하여 객관성을 유지하기 힘든 어려움이 따른다. 둘째는 자동화된 도구를 이용하여 분석하는 방법이다. 일반적으로 널리 쓰이는 방법은 문서 내에 등장하는 용어의 빈도수를 이용하는 방법이다. 문서 내에 등장하는 용어의 빈도수가 곧 문서 내용을 반영한다고는 규정할 수는 없지만 일반적으로 적절하여 대부분의 문서검색시스템에서 이 방법을 채택하고 있다. 본 시스템에서도 특정 문서 내에 등장하는 용어의 빈도수로써 문서의 상대적 관련성을 추출하고 이를 퍼지화한다. 그런데 또 한가지 유의할 점은 용어의 빈도수를 이용하여 특정 문서에 대한 용어의 종속성을 퍼지화하는 방법 역시 정해지지 않는다는 것이다. 본 논문에서는 용어 빈도수의 퍼지화를 위해 (그림 5)와 같은 멤버쉽 함수를 이용한다. (그림 5)에서 x는 용어의 빈도수이고 m은 빈도수에 대한 임계값이다. 본 시스템에서는 임계값 5를 m에 할당하였다.

$$\mu_{T_i}(x)=\begin{cases}\frac{1}{2}x, & 0\le x\le 1\\ \frac{0.5}{m-1}(x-1)+0.5, & 1<x\le m\\ 1, & m<x\end{cases}$$



(그림 5) 용어 빈도 퍼지화 멤버쉽 함수

4.1.2 축소용어집합생성기

축소용어집합생성기는 용어집합의 부분집합인 축소용어집합을 생성하는 처리기이다. (그림 6)은 축소용어집합생성기의 처리절차를 보여준다. 먼저 원시문서베이스로부터 레코드를 가져와서 각각의 퍼지 값에 α-cut을 적용하고 이를 ***sum***에 더하는 반복작업을 모든 문서에 대하여 반복한다. 마지막으로 ***sum***를 정렬한 후 축소용어집합을 선별하고 이를 DB에 저장한다.

```
1 : T : term set
2 : D : document Base
3 : R : reduced term set
4 : r : the size of reduced term set
5 : sum : sum of values
6 : d : document
7 : i : index
8 : level : α_level
9 :
10 : BEGIN
11 :    REPEAT FOR D
12 :       d = fetch_a_document(D)
13 :       REPEAT FOR T
14 :          sum[i] = α_cut(d[i], level)
15 :       END REPEAT
16 :    END REPEAT
17 :    sort sum
18 :    R = select_terms_and_build(sum, r)
19 : END
```

(그림 6) 축소용어집합생성기 순서도

4.1.3 문서베이스생성기

문서베이스생성기는 축소용어집합을 이용하여 원시문서베이스를 문서베이스로 변환하는 처리기이다. 문서베이스는 원시문서베이스에서 축소용어집합을 투영(projection)해서 산출할 수 있다. 문서베이스생성기의 절차는 (그림 7)과 같다.

```
1 : D : Document Base
2 : D_R : Original Document_Base
3 : R : Reduced Term Set
4 :
5 : BEGIN
6 :    D = Projection(D_R, R)
7 : END
```

(그림 7) 문서베이스생성기 순서도

4.1.4 시소러스생성기

```
1 : D : Document Base
2 : D^T : transpositive relation of Document Base
3 : D_R : Document Base on Reduced Term Set
4 : Th : Thesaurus
5 : n_T : size of Document Set
6 : n_D : size of Document Set
7 : n_R : size of Reduced Term set
8 : sum : sum of fuzzy values
9 : i, j, k : index
10 :
11 : BEGIN
12 :    D^T=Transpositive_Relation(D)
13 :    REPEAT FOR n_T BY i
14 :       REPEAT FOR n_D BY j
15 :          Initialize sum
16 :          Repeat for n_R by k
17 :             sum = sum+(D^T[i][j]→D_R[j][k])
18 :          End Repeat
19 :          Th[i][k] = Average(sum)
20 :       End REPEAT
21 :    End REPEAT
22 :    Store(Th)
23 : End
```

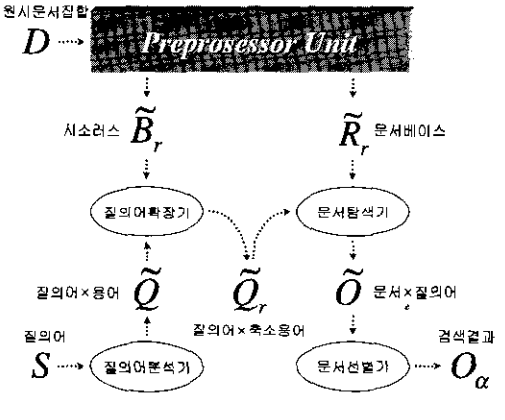
(그림 8) 시소러스생성기

시소러스생성기는 원시문서베이스의 전치행렬과 문서베이스에 퍼지관계곱 연산을 적용함으로써 시소러스를 구축하는 처리기이다. 시소러스생성기는 전처리부에서 가장 큰 처리시간을 요구한다. 시소러스생성기는 (그림 8)과 같은 처리절차를 따른다. 우선 DB에 저장되어 있는 퍼지관계행렬을 주기억장치에 적재하고 퍼지서브삼각논리곱을 적용하기 위한 반복처리를 수행한다. 이후 시소러스가 산출되며 이는 DB에 저장된다.

4.2 실시간처리부

실시간처리부는 사용자로부터 질의어가 입력되면 시소러스를 이용하여 이를 확장하고 문서베이스로부터 적합한 문서를 검색하여 제공하는 처리기이다. (그림 9)는 실시간처리부를 구성하는 4개의 처리기와 구조체를 보여준다.

질의어분석기(Query Analyzer)는 사용자로부터 임의의 질의어가 주어지면 이를 분석하여 용어와의 관계 $\tilde{Q}$를 산출한다. 질의어확장기(Query Expander)는 용어와의 관계 $\tilde{Q}$로 표현된 질의어를 시소러스를 이용하여 축소용어집합과의 관계 $\tilde{Q}_r$로 변환하는 처리기이다. 문서탐색기(Document Finder)는 문서베이스를 참조하여 질의어와 문서의 검색결과인 퍼지집합 $\tilde{O}$를 생성하는 처리기이다. 이때 퍼지집합 $\tilde{O}$는 주어진 질의어에 대한 각 문서들의 적합도를 표현한다. 그러나 $\tilde{O}$는 모든 문서에 대한 적합도를 표현하므로 사용자에게 $\tilde{O}$ 자체를 보여준다는 것은 적절치 못하다. 오히려 사용자에게 주어진 질의어에 대해 적합도가 높은 문서를 선별하여 제시하는 것이 더 바람직하다. 문서선별기(Document Selector)에서는 $\alpha-cut$을 이용하여 문서를 선별하여 제공한다.



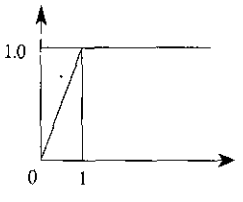
(그림 9) A-FIRM의 실시간처리부

4.2.1 질의어분석기

문서검색시스템에서 사용자의 검색요구는 질의어의 형태로 주어지지만 이를 해석할 구체적인 방법은 정해져 있지 않다. 일반적으로 다음과 같은 3가지 방법을 고려해 볼 수 있다. 첫째, 표현 형식의 제약을 두지 않는 방법이다. 이 방법은 사용자 검색요구를 용어의 단순 나열로 표현하므로 초보 검색자라도 무난하게 검색시스템을 이용할 수 있는 장점이 있다. 둘째, AND, OR, NOT 등과 같은 이진 논리 검색식을 사용하는 방법이다. 이는 주로 이진 논리를 다루는 불리언검색시스템에 이용되는 방법으로 퍼지정보검색시스템에는 적합하지 못하다. 셋째, 사용자의 검색요구를 용어, 퍼지논리연산자, 퍼지정량자의 조합으로 표현하는 방법이다. 퍼지논리연산자는 AND, OR, NOT, IMP 등이 있으며 퍼지정량자는 언어적 진리 값을 사용하거나 0과 1사이의 퍼지 값을 그대로 정량자로 사용 가능하다.

질의어분석기는 사용자의 검색요구가 표현된 질의어를 입력받아 이를 해석하여 질의에 대한 용어의 퍼지집합을 생성하는 처리기이다. 본 연구에서는 질의어의 표현 형식에 제약을 두지 않는 방법을 채택한다. 질의어는 하나의 독립된 문서로 간주되므로 전처리부의 문서분석기에 적용된 방법과 같이 포함된 용어를 분리 후 스테밍 처리를 하고 퍼지화하여 용어의 퍼지집합을 산출한다. 질의어의 퍼지화를 위해서는 질의어에 포함된 용어에 한해 1.0을 할당하는 방법을 이용한다.

$$\mu_T(x)=\begin{cases}0, & 0=x\\ 1, & 1<x\end{cases}$$



(그림 10) 용어 빈도 퍼지화 멤버쉽 함수

질의어분석기는 (그림 11)과 같은 처리절차를 가진다. 우선 사용자로부터 질의어를 입력받아 용어들을 분리하고 스테밍 처리를 가한 후 가용어만을 선별한다.

```
1 : Q_raw  : term set for query
2 : q_raw  : raw query
3 : q_terms : terms
4 :
5 : BEGIN
6 :     q_raw = get_query()
7 :     q_terms = separate_terms_from(q_raw)
8 :     stemming(q_terms)
9 :     select_and_update_available_words(q_terms)
10 :    fuzzify_the_query(q_terms)
11 :    store_result_to(Q_raw)
12 : END
```

(그림 11) 질의어분석기

4.2.2 질의어확장기

질의어확장기는 질의어와 용어의 관계로 표현된 검색요구를 시소러스를 이용하여 확장하여 질의어와 축소용어의 관계를 생성하는 처리기이다. 질의어의 확장은 용어의 퍼지집합으로 표현된 질의어와 용어와 축소용어집합의 퍼지관계로 표현된 시소러스와 퍼지합성을 이용한다.

질의어확장기의 수행절차는 (그림 12)과 같다. 우선 질의어와 용어의 관계를 메모리에 적재한다. 이후 수행되는 반복문에서는 시소러스로부터 주어진 용어들과 축소용어와의 관계를 저장한다. 마지막으로 RQ에 저장된 특정 용어와 축소용어의 관계들을 합성하여 질의어와 축소용어와의 관계를 생성한다. 여기서 눈여겨볼 것은 RQ를 합성하는 어떠한 방법도 규정되어 있지 못하다는 것이다. 최소값 선택, 최대값 선택, 산술평균 등 다양한 방법이 존재할 수 있다. 본 시스템에서는 산술평균을 이용한다.

```
1 : QR  : query on reduced term set
2 : Qraw : term set for query
3 : Th  : Thesaurus
4 : q   : raw query
5 : r   : the size of Reduced Term Set
6 : i   : index
7 :
8 : BEGIN
9 :     initialize(QR)
10 :    REPEAT FOR Qraw BY i
11 :        q = fetch_a_term(Qraw) BY
12 :        QR[i] = expand_meaning(q, Th)
13 :    END REPEAT
14 :    REPEAT FOR QR BY [i]
15 :        composition(QR[i][1...r])
16 :    END REPEAT
17 : END
```

(그림 12) 질의어확장기

4.2.3 문서탐색기

문서탐색기는 질의어와 축소용어의 관계를 입력받아 이를 문서베이스와 합성하여 문서와 질의어의 관계를 생성하는 처리기이다. 일반적으로 퍼지관계를 합성하는 방법에는 Max-Min, Max-Max, Min-Min 등 수많은 방법이 존재한다[17]. 본 시스템에서는 Max-Min 방법을 채택하였다. (그림 13)는 문서탐색기의 수행절차를 보여준다.

```
1 : DR  : document-base on reduced term set
2 : QR  : query on reduced term set
3 : Oraw : raw retrieved result
4 : q   : a query on reduced term set
5 : r   : the size of Reduced Term Set
6 : i, j : index
7 : max  : maximum fuzzy value
8 : min  : minimum fizzy value
9 :
10 : BEGIN
11 :    initialize(DR, QR)
12 :    REPEAT DR TO i BY d
13 :        max = 0
14 :        REPEAT QR TO j BY r
15 :            min = min(DR[i][j], QR[j])
16 :            max = max(max, min)
17 :        END REPEAT
18 :        O[i] = max
19 :    END REPEAT
20 : END
```

(그림 13) 질의어탐색기

4.2.4 문서선별기

문서선별기는 문서와 질의어와 관계 중 검색요구에 적절한 문서를 선별하여 출력하는 처리기이다. 본 시스템에서는 $\alpha - cut$을 이용하여 검색요구에 적절한 문서를 선별한다. α-level은 검색요구 시 사용자가 직접 설정할 수 있으며 디폴트값은 0.5이다. (그림 14)은 문서선별기의 순서도이다.

```
1 : DR  : document-base on reduced term set
2 : Oraw : raw retrieved result
3 : O   : retrieved result
4 : i, j : index
5 : alpha : value for alpha-cut
6 :
7 : BEGIN
8 :     j = 0
9 :     REPEAT Oraw TO i BY d
10 :        alpha = alpha_cut(Oraw[i])
11 :        IF alpha = 1 THEN
12 :            O[j] = i
13 :            j = j+1
14 :        END IF
15 :    END REPEAT
16 : END
```

(그림 14) 질의어분석기

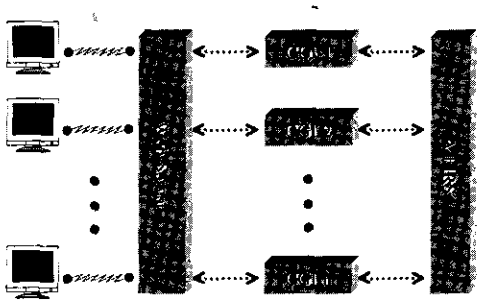
5. 시스템 구현

본 연구에서 구현한 A-FIRS의 개발 환경을 하드웨어와 소프트웨어 측면에서 살펴보면 표1과 같다. 본 시스템은 유닉스환경에서 데이터베이스와 C언어를 이용하여 개발되었다. 사용자 인터페이스는 WWW 환경에서 질의와 결과를 전달할 수 있도록 하였다.

〈표 1〉 시스템 구현 환경

Item	Content
Machine	DEC Alpha-station 255/300
Operating System	Digital Unix 4.0B
Language	C Language
Complier	Digital Unix 4.0A C Compiler
DBMS	Mini SQL 2.0.4
User Interface	WWW-based Interface

5.1 시스템 구현도



(그림 15) WWW 연동을 위한 구성도

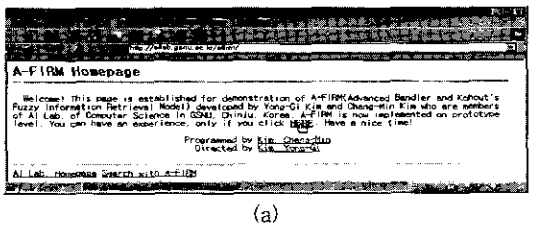
본 논문에서 A-FIRS는 WWW 기반 사용자 인터페이스를 제공한다. 사용자로부터 질의가 주어지면, CGI는 질의어를 처리하여 A-FIRS로 전달하고, 반환된 결과를 HTML형식으로 재구성하여 웹브라우저에 전달한다. 이때 A-FIRS는 서버 기능을 수행하며 CGI로부터 즉각 반응하여 해당 요청을 처리한다. (그림 15)는 WWW 연동을 위한 A-FIRS의 구성도이다.

5.2 처리한계시간

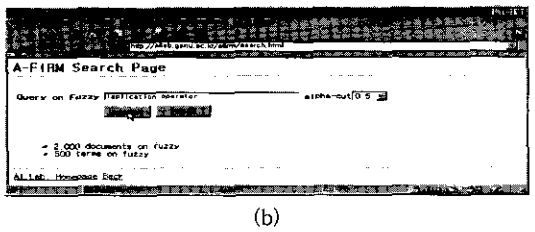
정보검색시스템에서 검색효율과 사용자와의 원활한 상호작용을 위해서 색인화(indexing)와 질의검색에 적정한 처리한계시간이 주어진다. 본 시스템의 처리한계시간은 전처리부의 처리시간과 실시간처리부의 응답시간에서 살펴볼 수 있다. 전처리부는 낮은 처리 빈도를 가지므로 오프라인 작업으로 이루어진다. 일반적으로 정보검색시스템은 짧게는 1주일, 길게는 6개월에 한 번씩 인덱싱을 수행한다. 본 시스템의 전처리부 처리한계시간을 1주일 이내로 한정한다. 실시간처리부는 사용자의 요구가 주어질 때 신속히 해당 문서를 찾아서 제시해주어야 하므로 매우 빠른 응답시간이 소요된다. 일반적으로 정보검색시스템은 10초 이하의 응답시간을 요구하며 최악의 경우에도 1분 이상을 초과하지 않는다. 본 시스템에서 실시간처리부의 기본 응답시간은 10초 내외로 정한다.

5.3 시스템 운영 예

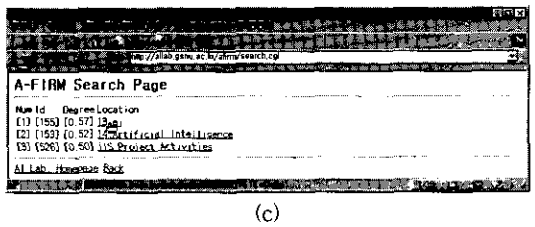
A-FIRS는 WWW를 위한 사용자 인터페이스를 갖추고 있다. (그림 16)은 본 시스템을 WWW 환경에서 운영한 예이다. (그림 16)(a)는 A-FIRS 홈페이지이고 (그림 16)(b)는 질의어와 검색결과를 선별하기 위한 α-level을 입력받는 검색요구페이지이다. (그림 16)(c)는 주어진 질의어에 대한 검색결과이다.



(a)



(b)



(c)

(그림 16) 시스템 운영 예

6. 결론 및 향후과제

BK-FIRM은 용어, 질의, 문헌과 같은 개체들의 관계를 퍼지관계행렬로 표현하고 개체연결과 같은 1차원적 단순검색 뿐만 아니라 개체간의 연관성에 근거하여 직접 관련이 없는 제3의 개체도 검색 가능하므로 단순 매칭으로 해결하기 힘든 화상 검색, 동영상 검색, 소프트웨어 재사용 등과 같은 분야에 유용한 검색기법이다.

기존의 검색이론과는 다른 BK-FIRM만의 특성에도 불구하고 실제 대용량의 문서나 용어를 다루는 검색시스템 적용이 어려웠던 것은 BK-FIRM의 연산 자체에 내재되어 있는 높은 시간복잡도 때문이었다. 본 연구에서는 BK-FIRM의 취약점을 축소한 A-FIRM 기초한 프로토타입 수준의 문서검색시스템인 A-FIRS를 구현한다.

A-FIRS는 크게 문서베이스와 시소러스를 구축하는 전처리부(preprocess unit)와 사용자의 검색요구를 처리하여 문서를 검색하는 실시간처리부(real-time process unit)로 구성된다. 전처리부는 문서 검색을 위해 문서베이스와 시소러스를 구축하는 부분이다. 전처리부는 처리시간에 거의 구애받지 않으며 처리빈도도 매우 낮아 오프라인(off-line) 형태의 작업에 적합하다. 실시간처리부는 검색식이 주어졌을 때 문서베이스와 시소러스를 이용하여 검색식에 적합한 문서를 검색 출력하는 부분이다. 실시간처리부는 정해진 반환시간(turn-around time) 내에 검색결과를 출력해야 하므로 온라인(on-line) 형태의 작업에 적합하다.

본 논문에서 A-FIRS는 WWW 기반 환경에서 구동되도록 구현되었다. A-FIRS는 서버시스템 내에서 독립적으로 수행되며, WWW 환경의 사용자로부터 검색요구를 받는다. CGI는 사용자로부터 주어진 질의어를 처리하여 A-FIRS로 전달하고, 반환된 결과를 HTML형식으로 재가공하여 웹브라우저에 전달한다.

참 고 문 헌

[1] Rijsbergen, C. J. van, Information Retrieval, 2nd edition, butterworths, 1979.

[2] Fox, E. A., and Sharat, S., "A Comparison of Two Methods for Soft Boolean Interpretation in Information Retrieval," Technical Report TR-86-1, Virginia Tech, Department of Computer Science, 1986.

[3] Paice, C. P., "Soft Evaluation of Boolean Search Queryies

in Information Retrieval Systems," Information Technology, Res. Dev. Application, 1984.

[4] Fox, E. A., "Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queryies and Multiple Concept Types," Cornell University, 1983.

[5] Bollmann, P., and Konrad, E., "Fuzzy Document Retrieval," in : Trappl R., Klir G. J. and Ricciardi L., eds., Progress in Cybernetics and Systems Research, Vol.3 (Hemisphere Publ. Comp., and John Wiley, Washington and New York) 355-363, 1976.

[6] Kohout, L. J., and Harris, M., "Computer Representation of Fuzzy and Crisp Relations by Means of Threaded Trees Using Foresets and Aftersets," Journal of Fuzzy Logic and Intelligent Systems, Vol.3, No.1, 1993.

[7] Kohout, L. J., Keravnou E. and Bandler W., "Automatic Documentary Information Retrieval by means of Fuzzy Relational Products," In Gaines, B. R., Zadeh L. A. and Zimmermann, H. J., editors Fuzzy Sets in Decision Analysis, pages 308-404, North-Holland, Amsterdam, 1984.

[8] Bandler, W., and Kohout L. J., "Fuzzy Power Sets and Fuzzy Implication Operator," Fuzzy Sets and Systems 4, 13-30, 1980.

[9] Bandler, W., and Kohout L. J., "The Identification Operators and Fuzzy Relational Products," International Journal of Man-Machine Studies 12 (1980) 89-116. Reprinted in : Mamdani E. H. and Gaines B. R., eds., Fuzzy Reasoning and Its Applications, Academic Press London, 1981.

[10] Keravnou, E. "Fuzzy Relational Products in Information Retrieval Systems," B. Tech. Dissertation, Dept. of Computer Science, Brunel University, 1982.

[11] Kohout, L. K., Bandler, W., "Fuzzy Relational Products as a Tool for Analysis and Synthesis of the Behaviour of Complex Natural and Artificial Systems," in : Wang S. K. and Chang P. P. eds., Fuzzy Sets : Theory and Application to Policy Analysis and Information Systems, Plenum Press, New York, 341-367, 1980.

[12] Kim, Yong-Gi and Kohout, L. J., "Use of Fuzzy Relational Products and Algorithms for generating Control strategies in resolution based Automated Reasoning," Proceedings of the fourth International Fuzzy System Association (IFSA) world congress, (Brussels, Belgium), July 7-12, 1991.

[13] Kim, Yong-Gi and Kohout, L. J., "Comparison of Fuzzy Implication Operators by means of Weighting Strategy in on Applied Computing (SAC'92)," Kansas City, March 1-3, 1992.

[14] Keravnou, E., "System for Experimental Verification of Deviance of Fuzzy Connectives in Information Retrieval Application," Second World Conference on Mathematics at the Service of Man. Topic 7, Measuring "Deviance in Non- Classical Logics and Modelling, Las Palmas (Canary Islands), June-July, 1982.

[15] Bandler, W., and Kohout, J. "Semantics of Implication operators and fuzzy relational products," Intl. Journal of Man-Machine Studies, 1980.

[16] Zimmermann, H. J. Fuzzy Set Theory and Its Application, Kluwer Academic Publishers, 1991.

[17] Salton, G., Automatic Text Processing, Addison-Wesley, 1989.

[18] Frakes, W. and Baeza-Yates, R., Information Retrieval, Prentice Hall, 1992.

[19] 김창민, 김용기, "퍼지관계곱을 이용한 정보검색시스템의 성능 개선", 한국퍼지 및 지능 시스템학회, 퍼지 및 지능 시스템학회 논문지, 제10권 제3호, 2000.

김 창 민

e-mail : nuno@ailab.gsnu.ac.kr

1997년 경상대학교 컴퓨터과학과(이학사)

1999년 경상대학교 컴퓨터과학과(공학석사)

1999년~현재 경상대학교 컴퓨터과학과 박사과정

관심분야 : 인공지능, 지식기반시스템, 자율무인잠수정, 지능항해시스템

김 용 기

e-mail : ygkim@nongae.gsnu.ac.kr

1978년 서울대학교 공과대학(공학사)

1987년 University of Montana(전산학석사)

1992년 Florida State University (전산학박사)

1982년~1984년 KIST시스템공학연구소 연구원

1992년~현재 경상대학교 컴퓨터과학과 부교수

관심분야 : 인공지능, 지식기반시스템, 자율무인잠수정, 지능항해시스템