정보표시를 이용한 화일처리와 정보검색시스템

박 재 완, 최 윤 철, 송 만 석

연 세 대 전 산 과 학 과
한 국 어 전 자 사 전 개 발 실
서울 서대문구 신촌동 134 (우편번호 120-140)

요 약

표본자료에 있는 정보표시는 그 표본에 있는 모든 자료의 내용을 상실하지 않도록 하기위해서 필요하다.그러한 정보표시는 또한 자료들의 구분을 명확히 하여 자료의 저장과 정보검색 목적으로 사용된다. 본 연구에서는 이러한 텍스트 정보표시의 잘못 쓰여진 오류의 검출과 수정 그리고 이러한 분류표시를 이용하여 방대한 표본자료를 정보표시별로의 화일처리와 필요한 정보에 관한 검색 시스템에 관하여 기술한다.

1. 서론

현재 본 연구의 바탕인 전자 사전 개발실에서는 본 대학 한국어 사전 편찬실에서 신뢰성있고 타당한 한국어 어휘뭉치를 표본 선정 기준으로 정하기위해 설문조사를 통해 <표 1>과 같은 300만 어휘뭉치를 이미 구축해 놓았다[2][3][6]. 어휘뭉치 속에는 텍스트외의 정보를 어휘뭉치에서 유지하기 위하여 여러가지의 정보 표시(Tag)를 붙여 놓았다[5]. 이러한 정보표시 사용목적은 첫째로는 모든 자료의 내용을 상실시키지 않기 위함이고, 둘째로는 말뭉치에 들어있는 자료들을 분류하기 위한 목적이고, 셋째로는 분류를 통한 자료 저장과 자료 검색에 그 목적을 둔다[7][12].

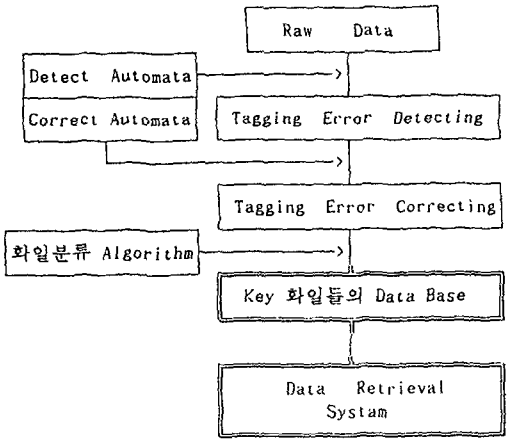
이러한 정보표시를 기입하는것도 아직은 컴퓨터가 할 수 없으므로 인간의 수작업으로 이루워진다. 이로인한 많은 정보표시에 대한 오류가 발생하게 되었다. 본 연구에서는 이러한 정보표시의 오류를 검출하고 그 검출된 오류를 정확한 정보표시로 바꾸어 주는 소프트웨어개발과 더불어 이런 완전한 정보표시를 이용하여 자료들의 화일 처리를 통해서 방대한 양의 어휘뭉치들을 정보표시로 분류하고 저장함으로 인해서 사용자가 편리하게 컴퓨터에 수록된 자료들을 원하는 유형과 목적에 맞게 원하는 자료를 검색할 수 있는 정보표시를 이용한 화일 처리와 검색시스템에 대한 연구를 제시한다.

본 연구에서는 이미 전자사전을 구축한 OED의 PAT 시스템[10]의 '질의어 방식 (Query Method)'과는 달리 풀-다운(PULL-DODW) 과 팝_업(POP-UP)방식 그리고 질의적 검색을 이용한'메뉴 유도 방식(Menu-Driven Method)'을 자료검색 시스템에서 사용한다.

전체적인 구조를 그림으로 설명하면 <그림 1>과 같다.

<표 1> 전문가 및 일반조사 결과에 의한 어휘뭉치 표본 선정 기준

① 신 문	33 %
② 잡 지	20 %
③ 소 설 및 수 필	18 %
④ 취 미 및 교 양	10 %
⑤ 수 기 및 전 기	9 %
⑥ 교 과 서	5 %
⑦ 희 곡 및 시 나 리 오	5 %
계	100 %



<그림 1> 본 연구의 전체적 흐름도

2. 표본 데이타의 정보표시 오류(Tagging Error) 처리

2-1. 정보표시의 오류 검출과 수정 자동장치(Tagging Error Detecting and Correcting Automata)

```
sample    ::= <ㅍㅂ> contents </ㅍㅂ> dummy sample ¦ eof
contents  ::= <ㅇㅎ> gettypeno </ㅇㅎ> <ㅊㅊ> chunks </ㅊㅊ>
              <ㅈㅁ> chunks </ㅈㅁ> {<ㅂㅁ> chunks </ㅂㅁ>}*
              {rests1 ¦ rests2 ¦ rests}
dummy[1]  ::= !notdummy ¦ ∈
gettypeno[1]::= !notdummy ¦ ∈
notdummy  ::= PB¦YH¦CC¦JM¦BM¦JJ¦NW¦Z_¦
              YY¦HH¦Y_¦HY¦HHY¦SPECIAL¦DONE¦INVALID
rests1    ::= <ㄴㅇ> dummynw </ㄴㅇ> {<ㅇㅇ> chunks </ㅇㅇ>} tail
dummynw[2]::= digit*
digit     ::= 0¦1¦2¦3¦4¦5¦6¦7¦8¦9
rests2    ::= {<ㅈㅈ> chunks </ㅈㅈ>} <ㄴㅇ> dummynw </ㄴㅇ>
               <ㅍ> dummy </ㅍ> {<ㅇㅇ> chunks </ㅇㅇ>} tail
rests3    ::= {<ㅍ> chunks </ㅈㅈ>} <ㄴㅇ> dummynw </ㄴㅇ>
               <ㅍ> dummy </ㅍ> tail
chunks[3] ::= <ㅎ> dummy </ㅎ> chunks ¦
              <ㅇ> dummy </ㅇ> chunks ¦
              <ㅎㅎ> dummy </ㅎㅎ> chunks ¦
              <ㅎㅇ> dummy </ㅎㅇ> chunks ¦
              <ㅎㅎㅇ> dummy </ㅎㅎㅇ> chunks ¦
              dummy chunks ¦ ∈
tail      ::= chunks {<ㅂㅁ> chunks </ㅂㅁ> chunks}*
```

(주1) dummy와 gettypeno는 tag 기호을 제외한 모든 문자를 받아들이는 규칙이다.
(주2) dummynw은 dummy와 마찬가지이나 숫자만을 골라서 출력하는 규칙이다.
(주3) chunks는 <ㅎ>, <ㅇ>, <ㅎㅎ>, <ㅎㅇ>, <ㅎㅎㅇ> 등을 받아들이는 규칙이다.

(1) 모든 표본에 있어서 <ㅇㅎ> 태그는 항상 <ㅍㅂ>뒤에 사용되었다. 때문에 이 이외에서 사용된 것은 모두 <ㅎㅇ>의 잘못된 표현이라고 간주하여 <ㅎㅇ>로 치환시켜 준다.

(2) 요약(<ㅇ ㅇ>) 테그는 외래어의 뜻인 <ㅇ>의 잘못된 표기로 사용된 예가 많았고 <ㅇ ㅇ>의 위치가 일정하지 않았다. 그래서 <ㅇ ㅇ>은 항상 선택적으로 주어지게 했다. 또한 <ㅂㅁ> ~ </ㅂㅁ>사이에 <ㅇ ㅇ>이 있거나 있어야 할 위치가 아닌 곳에 <ㅇ ㅇ>이 있을 경우는 모두 <ㅇ>으로 치환시킨다.

4. <ㅉ> 테그가 사용될 경우에는 항상 <ㄴㅇ>뒤에 사용된다. 그리고 <ㅈㅈ>는 사용될 경우 항상 <ㄴㅇ>앞에 온다. 그러므로 <ㅈㅈ>가 <ㄴㅇ>뒤에 올 경우에는 <ㅉ>로 치환시킨다.

5. 끝 테그에 slash('/')가 빠진 경우는 항상 삽입한다.

6. 테그처리에 있어서 테그의 짝짓기(pair matching)을 기본으로 해 준다.

7. 표본 데이타에서는 어느 일부분이 삭제되어 쓰여진 오류가 있다. 이런경우 앞,뒤 상황에 맞게 조절한다.

2-2. 오류 검출과 수정(Error detection and Correction) 단계

제 1 단계 <PASS 1>

시작 테그 기호 '<'와 끝 테그 기호 '>'은 오로지 테그를 위해서 사용되어야 함에도 불구하고 외국 지명이나 인명 혹은 인용문을 위해 사용되는 경우가 많았다. 이것을 방치했을 때 많은 문제점이 드러났었다. 그래서 pass1에서는 표본 화일을 읽으면서 테그 이외의 목적에 '<'나 '>'을 사용했을 경우에는 모두 '('나 ')'로 바꿔버린다.

제 2 단계<PASS 2>

① Tag의 형식을 취한 것 중에서 테그의 내용을 자소별로 분석하여 테그의 진위를 밝혀 테그가 아니라고 판명난 것은 모두 ( )로 치환시킨다.

② ①에 의해 테그라고 판명났지만 실제로 테그가 아닌 것은 그 위치에 맞는 적절한 테그로 치환시켜 준다.

③ 앞과 뒤 테그가 맞지 않을 경우 기본적으로 앞 테그에 맞추어 뒤 테그 가 따라가게 했다.

④ 완전히 틀린 테그오류는 오류 수정이 불가능하다.

3. 효율적인 자료 검색과 유지 보수를 위한 화일구조

정보표시가 존재하는 이유는 어떠한 정보도 없는 source text를 처리하기가 무척 힘들기 때문에 한국어 사전 편찬실에서 표본 선정후 처리하기 용이하게 하기 위해 위와 같은 테그들을 붙여 둔 것이다[5].

3-1. 표본의 각 테그를 이용한 화일구조 구성

다음 각 테그별로의 화일구조 구성에 관해 논한다.

(1) 유형별 화일(테그 표시 : <ㅇㅎ> ~ </ㅇㅎ>)

=> · 유형 테그는 어휘뭉치를 구분하기 위한 표본들의 유형을 나타낸다.

=> 화일구조 (6 바이트)

· 유형에 대한 화일처리는 현재 존재하는 유형 (20가지 유형)외에도 확장을 고려하여 1-99까지의 유형들이 존재한다고 가정하고서 처리를 한다.이중 99 유형은 표본의 유형부분의 삭제나 또는 정보표시의 오류들로 만들어진 화일이다. 그 이외의 유형들은 키 화일화 되어진다.

· 유형은 현재 정수 1-99까지이므로 2 바이트로 처리를 한다. 2 바이트는 어떤유형의 마지막 주 화일 인덱스를 지시한다.

· 또한 유형 화일은 그 이전에 자기 자신과 같은 유형의 화일을 지시하고 있어야 하므로 4바이트 크기의 인덱스 포인터가 필요하다(previous TYPE record number).

(2) 출처 화일(테그 표시 :<ㅊㅊ> ~ </ㅊㅊ>)

=> · 출처 테그는 각 표본 유형의 출처(펴낸 곳)를 나타내는 테그이다.

=> 화일구조 (14 바이트)

· 출처 테그는 대부분 책이름이나, 신문이름을 나타내는 테그이므로 최대 5글자로 잡고 10 바이트로 처리한다.나머지 4 바이트는 주 화일의 인덱스이다.

(3) 주제목 화일(테그 표시: <ㅈㅁ> ~ </ㅈㅁ>)

=> · 주제목 테그를 키 테그로 선택한 이유는 사용자 측면에서 찾고자 하는 내용을 화면에 보여줌으로 인해 사용자가 원하는 내용의 화일을 쉽게 선택하게 하기 위해서 주제목 화일을 만든 것이다.

=> 화일구조 (34 바이트)

· 주제목의 내용은 그 표본의 가장 핵심이 되는 내용을 나타내므로 길이가 그다지 길지 않으므로 15자(30 바이트) 정도를 최대로 한다.그 이상의 글자수에 대한 처리는 주 화일을 설명할 때 다루기로 한다. 4 바이트는 자신이 속한 주 화일의 인테스이다.

(4) 부제목 화일(테그 표시 : <ㅂㅁ> ~ </ㅂㅁ>)

=> · 부제목 테그를 이용한 화일의 생성 목적은 주제목과 동일하다.

=> 화일구조 (34 바이트)

· 단 부제목은 같은 표본내에서도 서로 내용을 달리하며 여러번 쓰일 수 있으므로 같은 표본 유형을 지시하는 주 화일 인덱스 포인터(4 바이트)와 자기 자신이 속해 있는 표본 화일의 레코드 번지를 지시하는 표본 인덱스 포인터(4 바이트)를 가지고 있다. 그 이외는 주제목과 동일하다.

(5) 저자 화일(테그 표시 : <ㅈㅈ> ~ </ㅈㅈ>)

=> · 저자 테그 또한 사용자가 같은 표본 유형에서도 어느 특정 인물이 쓴 내용을 필요로 할 때 화일 검색 시스템에서 필요한 키 테그이다.

=> 화일구조 (14 바이트)

· 최대 5자로 잡아 10 바이트 크기로 한정한다.그 이외의 4 바이트는 주 화일을 지시하는 인덱스이다.

(6) 년월 화일(테그 표시 : <ㄴㅇ> ~ </ㄴㅇ>)

=> · '년월'에 대한 화일은 단지 주 화일에 4 바이트 정수 유형으로 들어오다.

(7) 쪽(page) 화일(테그 표시: <ㅉ> ~ </ㅉ>)

=> · '쪽'에 대한 화일은 유형, 저자와 동일하다.다른 면은 '년월'과 마찬가지로 정보검색 후 어느책 몇 페이지에서 발췌했다는 내용을 사용자에게 보여 주는 역할을 한다.

=> 화일구조 (14 바이트)

· 4 바이트는 사용자가 원하는 내용이 어느 표본 유형, 어느 저자의 몇 페이지 속한 내용이라는 지시를 위해 필요로 한다.레코드 크기는 10 바이트이다.

(8) 요약 화일(테그 표시 : <ㅇㅇ> ~ </ㅇㅇ>)

=> · 요약 화일의 필요성도 역시 '년월','쪽'화일과 마찬가지로 자료검색이 다 이루워진 후에 사용자에게 이 화일에 있는 요약 내용을 보여주기 위해 필요하다[10].

=> 화일구조 (128 바이트)

· 레코드 크기를 124 바이트로 잡고 그 이상이 되는 레코드는 인덱스 포인터 필드에 연속(continue)표시를 해주어 처리하고 나머지 4바이트는 주화일의 인덱스이다.

(9) 표본 화일(그 이외의 테그들)

=> · 표본 화일의 내용은 그 앞의 모든 키 테그들과 그 내용을 제외시킨 그 이외의 표본 화일의 내용을 담아두기위한 화일이다.

=> 화일구조 (128 바이트)

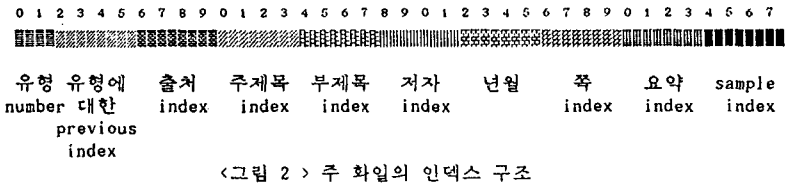
· 레코드 크기를 총 128 바이트로 하고 124 바이트에는 내용을 받아들이고 나머지 4 바이트는 그 내용을 지시하는 주 화일에 대한 인덱스 포인터를 둔다.

(10) 주(main) 화일

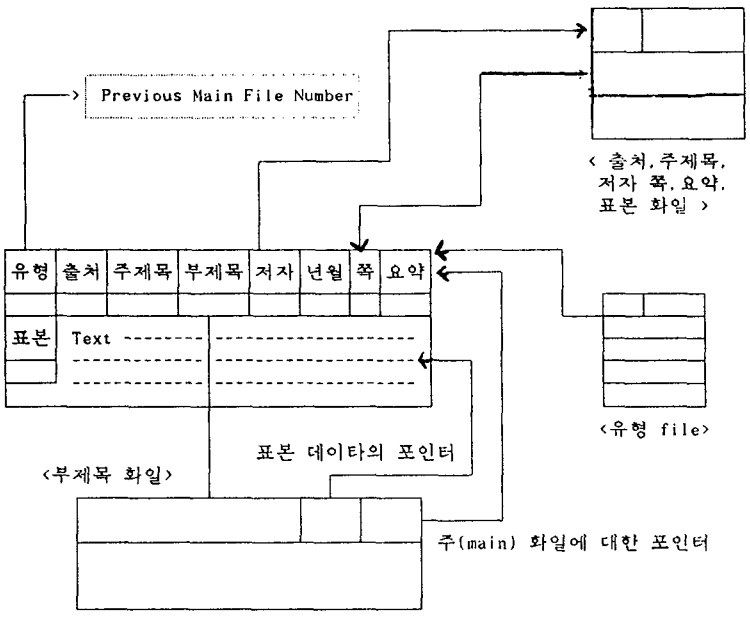
=> · 검색시 꼭 필요한 화일로서 모든 레코드 내용은 표본 화일에대한 인덱스들로만 구성되어져 있다. 다음에 검색할 때 이 주 화일의 내용을 따라가며 검색하므로 검색하는데 있어 효율성을 제시하는 화일이다.

=> 화일구조 (38 바이트)

주 화일의 38 바이트는 모두 인덱스로만 이루워져있다. 이 주 화일을 탐색함으로써 다른 화일을 쉽게 검색하게 하기위한 주 화일을 구성한다. 주 화일의 구성은 <그림 2>에 있다.<그림 3>은 모든 화일들에 대한 구성이다.



<그림 2 > 주 화일의 인덱스 구조



< 그림 3 > 전체적인 화일 구조

3-2. 화일 레코드 크기 (File Record Size)

화일 구조를 구성할때 화일 검색을 임의적으로 하기 위해서는 화일레코드 크기를 고정시켜야 한다. 레코드 크기를 고정시키면서 각 레코드 끝에 포인터 필드를 두어 하나의 레코드에 수용할 수 없는 데이타는 그 포인터 필드에 연속(continue)테그 '0'을 주어 계속 되는 데이타임을 나타내고, 모든 데이타를 저장 시킨 후에는 완료표시인 END('@') 표시를 한후 포인터 필드에 자신이 가르키는 주 화일에 대한 포인터를 기재한다.또한 부제목 화일은 인덱스를 두개를 가지고 있으므로 END표시 이외에 '^'표시를 두어 연결되어진 화일을 나타내도록 하였다. 결론적으로 주 화일을 임의 크기(random size) 화일로 구성한 이유는 탐색이후에 빠른 접근을 하기 위함이다. 또 각 화일의 레코드에는 역 연결(invert link)이 구성 되어 있기 때문에 어느 화일에서 탐색을 시작하여도 다른 모든 화일을 접근할 수 있는 방법을 제공하기 위함이다.

4. 정보 검색 시스템 (Information Retrieval System)

4-1. 정보검색시스템의 개요

정보 검색시스템은 시스템의 사용자가 필요로 하는 정보를 수집하여 내용을 분석한 뒤,찾기쉬운 형태로 조직하여 두었다가 정보에 대한 요구가 발생했을때 정확한 정보를 신속하게 제공하는 것을 말한다[4][14].

본 논문에서는 이미 분류 저장되어진 데이타베이스내에 있는 화일 구조를 이용해서 다음과 같은 정보 검색 시스템을 제시한다. 화일 구조에서 임의 접근이 가능하도록 이미 설명한 주 화일은 모든 키 화일에 대한 인덱스들을 가지고 있고 각각의 키 화일들은 주 화일에 대한 역 인덱스(invert index)를 가지고 있으므로 정보 검색시 화일 탐색을 할 때 임의적으로 처리하므로 효율성이 부여된다.

4-2. 키 화일을 이용한 정보검색시스템

표본의 유형에 대한 각각의 분야들은 정보표시중 유형 테그로 분류 되어 있고,화일 구조의 주 화일에 각각 저장되어 있다.정보 검색에 있어서의 궁극적인 목적은 모든 키 화일들을 탐색해서 사용자에게 필요한 키 화일들의 내용과 그 키 화일에 대한 표본 화일에 대한 내용을 보여주는것을 그 목적으로 한다. 정보검색시스템은 메뉴유도방식을 이용하여 순차적 접근과 질의적 접근으로 크게 두가지로 구분된다.

4-2-1. 순차적 접근

순차적 접근은 사용자가 원하는 정보에 대해 아무런 지식도 없는 상황에서 정보를 검색하는 경우에 사용하는 방식이다.

1) 유형 필드 : <표 1>에 있는 자료의 유형이 화면에 출력된다.
2) 종류 필드 : 각 유형에 대한 여러 종류들이 화면에 출력된다.

3) 출처 필드 : 선택된 종류에 대한 모든 출처가 화면에 출력된다.

4) 저자 필드 : 선택된 출처에 대한 모든 저자가 화면에 출력된다.

5) 주제목 필드 : 선택된 저자에 대한 모든 주제목들이 화면에 출력된다.

6) 부제목 필드 : 주제목에서 선택한 것이 아무것도 없을 때 출력된다.

7) 표본 필드 : 주제목이나 부제목에서 선택된 내용에 대한 모든 표본 화일들이 선택된 순서대로 화면에 출력된다.이 때 년월,요약, 쪽 내용도 함께 출력된다.

4-2-2. 질의적(Query) 접근

질의적 접근 방식은 사용자가 원하는 내용을 알고 있을때 그 내용을 직접 입력하여 화일을 검색한다. 이 방식의 필드로는 출처,저자,주제목,부제목,표본 필드가 있다.사용자가 내용을 알고있는 필드에서 직접 원하는 화일을 검색할 수 있다. 표본 필드는 순차적 접근 방식과 같다. 접근 방식은 주제목과 부제목에서 사용자가 입력한 질의어와 분류 저장된 화일을 선형탐색(linear search)방법을 통해 서로 대응시켜 추출한다. 저자와 출처는 해쉬테이블을 구성하여 질의어와 대응(matching)시켜 추출한다.또는 해쉬 테이블 구성시 메모리가 부족할때는 주제목,부제목과 동일하게 대응시킨다.

5. 결론

본 논문에서는 정보표시(Tag)를 이용한 오류 검출 및 수정작업과 정보표시를 이용한 화일 처리와 저장된 화일을 이용한 표본데이타의 정보검색시스템에 관해 기술하였다. 그런데 정보표시를 하는데 있어서 미리 사전의 약속인 정확한 규칙(exact rules)없는 상황하에서 표본 데이타를 삽입함으로 인한 작업의 비 효율성을 많이 발견하게 되었다. 이로 인하여 다음과 같은 정보표시의 규칙(rule)을 제시해 본다.

(1) 키 테그를 삭제해서는 안된다.만약 내용이 없다면 정보표시의 자료는 없는 상황에서 정보표시는 삭제해서는 안된다.

(2) 키 테그들 간에는 어느정도 순서를 부여한다. 이러한 순서내에서는 테그 삭제가 되었을때 그 부분에 그 테그를 임의로 삽입할 수 있다.

(3) 키 테그와 표본 자료 테그를 분리 시키면 화일 처리를 하는데있어서의 효율성을 증가시킨다.예를들자면 키 테그의 시작을 알리는 <ㅅㅈ>~</ㅅㅈ>과 표본 자료 테그를 알리는 <ㅈㄹ> ~ </ㅈㄹ>표시를 삽입하면 화일 처리를 하기가 한결 편리하고 유지시키기가 용이할 것이다.

(4) 테그 편집기(Editor)를 구상해본다. 위의 세가지의 규칙이 세워진다면 그러한 순서에 맞는 테그편집기를 설계할 수 있을 것이다.

이러한 규칙이 지켜준다면 정보표시 오류 검출, 수정과 화일 처리 구성에 일반성을 부여할 수 있을것이다. 본 논문에 이어 연구할 분야는 주어진 화일 구조내에서의 하이퍼텍스트화되고 키 워드를 이용한 질의어(Query)에 의한 자료 검색 시스템에 대한 연구가 필요하리라 생각된다. 질의어(Query) 와 하이퍼텍스트에 대한 내용이 메뉴 유도 방식에 결합되어 진다면 주어진 어휘뭉치에 대한 정보검색과 내용참조(content reference)는 보다 더 효율적이고 편리한 시스템을 제공하게 될것이다.

6. 참고문헌

[1] 김영택,"한국어 정보처리의 현안문제와 향후 과제",제 2회 한국정보과학회 춘계워크샵,국어 정보 처리, 1990.
[2] 이상섭,남기심외,새한국어 사전 편찬을 위한 사전편찬 연구,제 1 집,제 2 집,탑 출판사, 1988.
[3] 이상섭, "뭉치언어학 : 사전 편찬의 필수적 개념", 한글 및 한국어 정보처리 학술 발표 논문집, 1989.
[4] 정영미,"정보검색론", 정음사, 1988.
[5] 정찬섭,"한국어 어휘 뭉치의 표본 선정 기준", 한글 및 한국어 정보처리 학술 발표 노문집, 1989.
[6] 조재수,국어 사전 편찬론,과학사, 1984.
[7] 최운철,송만석,"한국어 전자사전 개발의 현황과 과제",한국정보과학회 국어정보처리 춘계워크샵 학술 발표논문집, 1990.
[8] 홍종화,"초성 테이블을 이용한 한글 문헌 정보 검색 시스템의 설계 및 구현",한양대 산업대학원 석사학위 논문, 1986.
[9] Darrell R. Raymond and Frank WM.Tompa,"hypertext and the OXFORD ENGLISH DICTIONARY", Comm. of ACM, Vol.31, 1988.
[10] Gaston H. Gonnet,"Examples of PAT applied to the Oxford English Dictionary",UW centre,Canada,July,1987.
[11] Gaston H.Gonnet,"Efficient searching of Text and Pictures Extended Abstract", UW centre, Canada, 1987.
[12] Header Fawcett,"Using Tagged Text to Support Online Views", UW centre,Canada
[13] Header fawcett,"Adopting SGML:The Implications for writers",UW centre,Canada
[14] Salton,G. and M.J.Mcgill,"Introduction to Modern Information Retrieval",McGraw-Hill,new York,1983.