論文
2-10-9

웹 게시판 비속어 처리 프로그램의 설계 및 구현 (Design and Implementation of a Swearing Remover Program on Web board)

조 아 영*
(Ah-Young Cho)

요 약

기존의 웹 게시판 비속어 처리 프로그램들은 입력 차단형이라는 한계성 때문에 비속어의 모양이 조금만 변형이 되어도 비속어를 차단하지 못하는 문제점이 있었다. 이러한 한계성을 극복하기 위하여 본 논문에서는 인터넷의 열려있는 게시판에 대해 분석 및 삭제/치환하는 감시형 프로그램을 개발하였다. 이를 위해 먼저 웹 상의 비속어 패턴을 분류하였고 이를 바탕으로 이러한 패턴들을 분석할 수 있는 토크나이저를 개발하였다. 그리고 각 게시판에 대한 토크나이징 및 비속어 삭제/치환 모듈을 스레드로 구현하여 각각 병렬적으로 작업제어가 되도록 구현하였다.

이 프로그램을 웹 게시판의 데이터에 실험적으로 작동시킨 결과 대부분의 비속어를 검출하여 91.9%의 재현율을 보였으나 형태소적 변형 비속어[1]와 문맥적 비속어의 검출에는 다소 미흡함을 나타내었다. 이 프로그램은 품사적, 의미적 중의어와 문맥적 비속어의 처리에 있어서 이 프로그램의 수동모드의 확장을 통해 앞으로 더욱 보완되어야 할 것이다. 이 프로그램은 게시판 사용자들을 적절한 단어사용에로 유도하며, 공공기관, 학교, 방송국 등의 게시판 관리자의 수작업을 대체해 줄 수 있을 것으로 기대된다.

ABSTRACT

The existing swearing remover programs could not have blocked even slightly transformed swearings because of their input blocking properties. To overcome these defects, this paper implemented a supervising program which analysize and remove/replace swearings on web board. For this purpose this paper first classified the patterns of swearings on web board and then implemented a tokenizer which can analysize those patterns. The module tokenizing and removing/replacing swearings on each web board was implemented as a thread so that it could be parallely controlled.

As a result of running this program on some web boards , we found out it had detected almost of the swearings as 91.9% of recall but it could not meet our purpose sufficiently on morphological transformed swearings and swearings in context. So the studies will be continued about processing on morphological ambiguous words, ambiguous words in meaning and sweaings in context by extracting this program's manual mode. We expect this program could induce the users to proper usage of words and replace the manual works of web board managers in schools, public bodies, broadcasting stations etc.

[1] 형태소적 변형 비속어 : 본 논문의 3.3절의 "타"항목에서 정의함.

* 정회원 : 울산대학교 정보통신대학원, (주)자룩스

논문접수 : 2001. 10. 11.
심사완료 : 2001. 10. 22.

1. 서론

국내 인터넷 사용인구는 이제 2000만이나 되고 있다. 그러나 인터넷 사용 인구가 늘어남에 따라 비속어 및 폭력적 언어의 사용도 늘어나고 있으며, 이러한 언어사용은 어린이와 청소년들에게도 자연스럽게 수용되고 있다. 이에 따라 공공기관, 학교 등에서는 비속어 삭제 관리의 어려움을 호소하고 있으며 정부는 네티켓에 대한 교육 및 욕설에 의한 사이버 폭력 예방책 등을 시행하고 있다. 또한 산업계에서는 비속어 차단 소프트웨어들을 내 놓고 있다. 이에 대해 이러한 규제방식이 타율적이고 무분별히 나타날 때에는 네티즌은 강력히 반대하여 왔다[1,2,3]. 이러한 배경 하에서 비속어를 처리하는 소프트웨어는 네티즌의 자율성을 최대한 고려하면서도 비속어 사용을 규제할 수 있도록 구현해야 할 필요성이 있다.

기존의 비속어 차단 소프트웨어에서는 두 가지 문제점을 찾아 볼 수 있다. 첫째는 기존의 소프트웨어는 입력 차단형 이어서 사용자의 반발심을 불러일으킬 수 있다는 것이다. 둘째는 입력 시에만 차단함으로써 비속어의 모양을 변형시켜 입력하는 경우에 한 번 차단하지 못하면 이미 입력된 글에 대해서는 처리할 방안이 없다는 점이다. 이 논문에서는 이러한 한계성을 극복할 수 있는 감시형 프로그램을 개발하였다 이 프로그램은 이미 입력된 글에 대해 관리자의 설정모드에 따라 비속어를 추출한 후 처리하기 때문에 게시판의 글 입력을 차단하지 않는다. 또한 이 프로그램은 동적 관리 로봇 방식으로 구현하여서 비속어의 변형어가 입력된 후에라도 이를 삭제, 치환 처리될 수 있게 하였다. 또한 Java 및 JDBC 기술을 사용하여 운영 플랫폼과 게시판 데이터베이스에 독립적으로 작동할 수 있게 하였으며, 웹 어플리케이션인 JSP(Java Server Pages)로 구현하여서 언제 어디서나 관리가 용이하도록 배려하였다. 앞으로 이 프로그램의 명칭은 Webcleaner로 기술한다.

2. 관련연구

2.1 JSP/JDBC를 이용한 게시판 RDBMS와의 연동

일반적으로 웹 게시판 프로그램은 게시물 정보를 데이터베이스 또는 파일에 저장해 두고 이러한 게시물의 정보를 CGI(Common Gateway Interface)에 의해 HTML(Hyper Text Markup Language)형태로 인터넷 상에서 볼 수 있도록 짜여져 있다. 이렇게 이미 웹 게시판 프로그램이 구현되어 있는 상태에서 Webcleaner는 이 게시판의 게시물에 접근을 하여 비속어를 추출 및 처리한다. JSP로 구현함으로써 단일 프로세스에서 다중 스레드로 작동하는 Servlet의 효율성이 적용된다. 프로그래밍 모델은 JSP-Servlet-저장구조의 3계층 구조로 프로그래밍한다. 이는 JSP는 화면에 결과를 보여주는 프리젠테이션의 역할만 하고 Servlet은 JSP의 요청을 받아 저장구조를 억세스하며 정보처리를 수행한다. 이러한 프로그래밍 모델은 JSP에서 직접 저장구조를 억세스하는 프로그래밍 모델보다 조금 복잡하긴 하지만 모듈화가 잘 이루어져 프로그래밍의 재사용성, 유지보수성을 높여 생산성을 높여주는 모델이다.

웹 게시판의 데이터는 데이터베이스에 저장되는 경우, 파일에 저장되는 경우 등이 있으나 이 논문에서는 데이터베이스에 저장되는 웹 게시판을 대상으로하며 데이터베이스에 접근은 자바소프트에서 제공하는 JDBC(Java Database Connectivity) API를 사용한다. JDBC는 자바 프로그램 내에서 SQL문을 실행하기 위한 자바 API로 JDBC를 사용하면, 어떠한 RDBMS(Relational Database Management System)로도 SQL문을 전송하기 쉽다. 즉, JDBC를 사용하면 JDBC를 지원하는 드라이버가 제공되면 그것을 통하여 그 데이터베이스의 내장 프로시저를 호출하여 데이터베이스 URL, 데이터베이스 접근 사용자, 접근 패스워드를 가지고 데이터베이스에 접근한다. 또한 테이블의 각 필드에의 접근은 JDBC의 ResultSet객체를 이용한다.

2.2 다중 쓰레드 기법과 싱글톤 관리 기법

Webcleaner는 여러 개의 게시판을 병렬적으로 로봇 방식으로 감시하기 위해 다중 스레드 기법과 싱글톤(Singleton) 관리 기법을 사용한다. 다중 스레드 기법은 병렬 처리나 동시 처리가 요구되는 시스템에서 많이 사용되는 기술로, 다수의 프로세스를 생성하여 처리하는 기법에 비하여 스레드 생성시 적은 자원만을 필요로 하며, 시스템 자원을 비교적 효율적으로 사용할 수 있다는 장점을 가지고 있다. 반면, 자원 공유의 특성 때문에 프로그램 상에서 정교한 타이밍 실패나 의도하지 않은 변수의 공유가 발생하는 경우 원하는 프로그램을 구현하는 것이 실패하게 되는 위험이 있다[4].

Webcleaner에서 각 게시판에 대한 감시 모듈은 각각 하나의 스레드로 구현이 되어 있고, 각 스레드는 비속어 리스트를 공유함으로써 자원을 효율적으로 사용하게 된다. 또한 각 비속어 리스트를 동기화를 지원하는 해시테이블로 생성함으로써 비속어 리스트 공유로 인한 동기화 문제를 해결하였다.

싱글톤 패턴이란 어떠한 클래스를 생성 시 단 하나의 인스턴스만을 가지도록 생성하는 것이다. 싱글톤 관리는 한 개 이상의 싱글톤 클래스들을 한 장소에 편리하게 저장해 놓기 위하여 모든 싱글톤 클래스들의 registry를 생성하여 그 registry가 어디서나 사용 가능하도록 해 놓는 것이다. 이 때 하나의 싱글톤이 그 자신을 생성할 때 마다 그 인스턴스는 그 registry에 알려진다. 이렇게 하면 그 프로그램의 어느 부분에서든 그 중 어떤 싱글톤 인스턴스 변수든 사용하고 다시 반납하고 할 수 있다. 그리고 물론 registry 자신은 싱글톤이어야 하며 프로그램의 모든 부분에 생성자나 또 다른 메서드들을 통해 넘겨지게 한다[9].

Webcleaner에서는 각 게시판 스레드를 임의로 시작시키고, 중단시키며, 업데이트 할 수 있게 하기 위해 게시판 스레드에 대한 registry를 싱글톤 관리자로 구현해 놓고 프로그램의 어느 부분에서든 접근하여 제어할 수 있게 하였다. 이렇게 함으로써 공유자원인 게시판 각각에 대하여 단 하나의 감시 스레드만 생성하며 제어를 용이하게 할 수 있다.

2.3 패턴매칭

일반적으로 정보 검색 시스템의 성능은 정확성과 신속성에 달려 있다. 검색의 정확성을 높이는 데에는 검색어로서 색인어의 역할이 아주 중요하며, 신속한 색인어 탐색을 위해서는 주로 이진 트리와 해시 기법 등을 이용한다.

색인어의 추출은 일반적으로 형태소 분석 단계에서 문자 상태를 구분하여 문자 상태별로 형태소 분석을 수행하고 그리고 나서 자동색인 시스템에서 색인어를 추출하게 된다. 또한 색인어 추출 과정에서는 원형을 복원하거나 기본형으로 변환하기도 하며, 추출된 색인어는 등록단계에서 불용어 검사와 복합명사 확장 처리를 해야 한다[5].

Webcleaner는 정보 검색 및 정보 처리 프로그램으로, 정보 검색 시 게시판을 감시하기 위해 반복적으로 수행할 수 있어야 하므로 시스템에 상당한 부하를 줄 수 있는 형태소 분석 대신 패턴분석으로 색인어를 추출한다. 또한 색인의 목적이 일반 정보 검색 시스템과는 다른데, 사용자의 질의에 대한 정보 검색을 해 주기 위해 색인을 하는 것이 아니라, 생성된 색인어들을 비속어 리스트와의 패턴매칭에 사용하며, 색인정보를 게시물 치환 시 사용하기 위해서 이다.

Webcleaner의 색인모듈은 정확한 패턴매칭을 위하여 웹 게시판의 비속어 패턴을 조사하여 비속어의 다양한 패턴을 최대한 추출할 수 있도록 고안하였다. 즉, 색인 모듈은 한국어로 이루어진 게시물 본문으로부터 구분자(delimeters)에 의한 토크나이징 과정을 거쳐 색인어로 적합하지 않은 불용어를 제거한 후 음절별로 이루어진 단어를 결합하고, 음소별로 이루어진 단어를 음절로 결합하여 기본형으로 복원한다. 여기서 한국어는 자바에서 코드변환 과정없이 처리할 수 있는 유니코드 문자 세트를 이용하였으며 컨트롤 문자와 키보드의 특수문자 세트를 불용어로 처리하였다. 패턴매칭은 토크나이저에 만들어진 각각의 색인어토큰에 대해 서브워드를 취해서 비속어 리스트를 검색함으로써 이루어진다. Webcleaner에서는 비속어 리스트를 비속어의 첫 음소별로 해시테이블의 특정 키에 대한 리스트로 저장하여 두고, 리스트를 정렬해 두었다. 그리고 나서 패턴매칭 시 각 색인어의 첫 음소별로 특정 리스트를 찾아가서 이진탐

색(binary search) 기법으로 검색함으로 신속히 검색 될 수 있도록 하였다.

3 . 설계 및 구현

3.1 정의

3.1.1 요구사항

Webcleaner가 갖추어야 할 요구사항들은 다음과 같다.

첫째, 의사소통의 자유를 차단하지 않는 프로그램을 구현하기 위해 게시판 프로그램과는 따로 작동하는 게시판 관리 프로그램으로 하며, 게시판의 성격에 따라 비속어 등급별 추출정도를 조정하여 삭제 또는 치환 또는 수동으로 처리할 수 있도록 1차 검색 후 2차 삭제 또는 치환기능을 가진다.

둘째, 비속어 추출도를 높이기 위해 특수문자세트에 의한 단어구분을 하며, 패턴매칭을 통해 띄어쓰기가 된 비속어, 문장 중간의 비속어도 검출해 내도록 한다. 그리고 새로운 비속어와 비속어 레벨을 등록할 수 있도록 하며 기존 비속어를 수정 가능하게 한다.

셋째, 편리한 운영을 위해 동시에 여러 개의 게시판에 대해 추출작업을 수행할 수 있게 하며 예약 반복수행 기능과 날짜별 처리결과 로그 기록을 한다.

넷째, 비속어 추출 및 처리 모듈은 자료구조 및 알고리즘 설계 시 효율성을 중심으로 하여 시스템에 부하를 덜 주도록 한다.

3.1.2 비속어 추출/처리 모드

Webcleaner는 비속어 등급별 추출정도를 조정하여 처리할 수 있도록 다음의 <표 1>에서와 같은 비속어 등급을 정의한다. <표 1>에서 단어개수는 Webcleaner 초기에 등록되어 있는 비속어의 개수이다.

<표 1> 비속어 단어 등급의 정의

<Table 1> Leveling of swearing

등급 : 의미	단어개수
1등급: 비속어가 확실함	685
2등급: 문맥상 비속어일 확률이 50% 이상임	204
3등급: 비어에 가까움	92

위와 같이 등급이 정해진 상태에서 각 게시판 마다 검색모드와 처리모드를 <표 2>, <표 3>과 같이 설정할 수 있다

<표 2> 비속어 검색모드

<Table 2> swearing search mode

모드	의미
1	1등급만 찾음
2	1등급, 2등급 찾음
3	1등급, 2등급, 3등급 찾음

<표 3> 비속어 처리모드

<Table 3> swearing process mode

모드	의미
1	자동 삭제
2	자동 치환
3	수동 삭제/치환
4	자동 삭제 반복수행([ ]시간[ ]분 마다)
5	자동 치환 반복수행([ ]시간[ ]분 마다)

비속어를 처리하는 모드로는 크게 자동모드와 수동모드가 있는데 수동모드의 경우 1차로 검색만 하고 2 차로 검색된 결과 내에서 검색모드를 다시 취하여 재검색 해 본 후에 관리자가 삭제 또는 치환결정을 내릴 수 있게 한다. 이 모드는 문장 속에서 문맥에 따라 비속어가 될 수도 있고 아닐 수도 있는 중의적 단어에 대해서 특별히 쓰일 수 있다. 그리고 처리와 처리결과의 저장을 위해서 다음 세 가지의 텍스트 유형을 정의한다.

- 본문텍스트: 게시물 원본 텍스트
- 변환텍스트: 본문 텍스트에서 비속어를 찾은 곳을 비속어의 레벨별로 특정색으로 표시하여 놓은 텍스트

· 치환텍스트: 본문 텍스트에서 비속어가 발견된 곳을 치환문자열로 치환한 텍스트

처리모드가 삭제의 경우는 본문텍스트에서 한 개의 비속어 단어를 발견하면 비속어 검출을 멈추고 DBMS로 delete 질의를 보내어 그 게시물을 삭제하며, 치환의 경우는 본문텍스트의 끝까지 비속어 단어를 모두 찾아서 사용자가 지정한 문자로 치환한다. 예를 들어 "개새끼"는 "XXX"로 치환한다. 그리고 RDBMS로 치환텍스트로 update질의를 보낸다. 수동인 경우는 결과를 1차 파일에 저장해 두었다가 2차로 사용자의 요구시 RDBMS로 updte/delete 질의를 보내어 처리한다.

3.2 구성

Webcleaner의 개발환경은 <표 4>와 같으며, Webcleaner의 실행환경은 첫째로 서버시스템은 운영체제에 상관없고, 게시판 시스템의 데이터베이스의 종류에 상관이 없이 실행된다. 그리고 CPU는PENTIUM 500 Mhz 이상, RAM은 512M 이상을 권장한다. 둘째로 클라이언트 시스템은 IE explorer 등 IFRAME을 지원하는 브라우저가 필요하다.

<표 4> Webcleaner 개발환경

<Table 4> Developing environment of Webcleaner

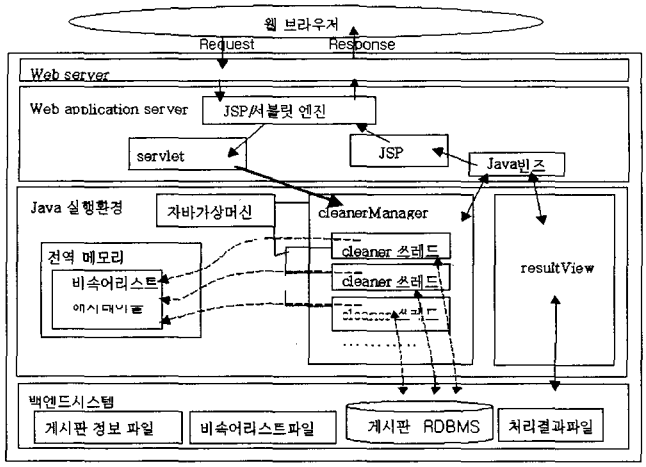
운영체제	Linux RedHat 6.2
CPU	1GHz
Memory	128M
HDD	18.2 GB(SCSI)
저장구조	파일구조, 데이터베이스 사용하지 않음
웹 서버 & JSP 서버	Jakarta Tomcat 3.2.1
프로그래밍 언어	Java, JSP
프로그램 개발 툴	JDK 1.3

Webcleaner는 프로그래밍 모델 중 MVC(Model-View- Controller) 디자인 패턴[6]을 따라 개발하였으다. 게시판 시스템에의 접근은 Webcleaner가 감시해야 할 게시판시스템이 어떤 RDBMS에서 구현되었든지 상관없이DB에 접근할 수 있도록JDBC를 사용했다. 그리고 Java실행환경과 JSP/서블릿 엔진 설치를 Webcleaner 설치 과정에 포함시켰다.

[그림 1]은 Webcleaner 시스템의 구성도 이다. 그림의 Web server와 Web application server부를 보면 Http Request를 서블릿이 받아서 자바 실행환경의 비즈니스 로직부분을 호출하며 Controller의 역할을 하고 있고, 비즈니스 로직에서는 백엔드시스템과 통신한 결과를 Model인 자바빈즈에 넘겨준다. 그러면 JSP에서는 자바빈즈의 내용을 화면에 보여주는 View의 역할을 하여 Response를 보낸다. 백엔드시스템에는 게시판DB와 비속어리스트 파일, 결과를 저장하는 파일, 비속어를 처리할 게시판리스트 정보를 가지고 있는 게시판 정보파일이 있다. 특정 게시판에 대한 비속어처리를 시작하라는 Request가 있을 때 비즈니스 로직부의 cleaner 스레드 중 그 게시판에 관한 스레드를 시작시키게 된다. cleaner 스레드는 자바가상머신에서 다중 스레드로 실행되면서 주기억장치의 비속어리스트를 공유하므로 효율적이며 각 게시판에 대한 동시수행이 가능하다.

[그림 1]에서 보이는 cleaner 스레드는 Webcleaner의 각 게시판에 대한 감시 스레드로서 게시물 본문에서 비속어를 추출해 내어 삭제 또는 치환을 수행한다. 이 스레드는 크게 토크나이저(색인 모듈)와 패턴매칭 및 처리모듈로 나뉜다.

각 모듈은 다음의 절에서 설명한다.



[그림 1] Webcleaner 구성도

[Fig. 1] Cofiguration of Webcleaner

3.3 비속어 패턴 조사 및 토크나이저 개발

<표 5>는 웹 게시판 상에서 나타나는 비속어의 패턴을 분류한 것이다.

<표 5> 웹 게시판에 나타난 비속어 패턴 분류

<Table 5> Classification of swearing pattern on Web Board

설 명	비속어 패턴의 예
가. 문장중간의 비속어	이미친년
나. 합성어	개씹쌩년
다. 한 음절씩 띄어쓰기	개 새 끼 야
라. 그림문자	凸먹어라
마. 움소불로 띄어쓰기	ㅅㅅ ㅣㅂㅂ ㅏ
바. 한 줄씩 건너 쓴 비속어 패턴	씹 년 아
사. 비속어 중간에 특수문자	바~보
아. 영문으로 시작	C8놈
자. 숫자로 시작	18넘
차. 약한음, 강한음, 사투리 비속어	이 새기야, 이 쉐끼야, 몽그라뭑은 놈[2] (막 굴러 먹은놈:제주)
카. 여러가지 패턴이 합성된 패턴	ㅎㅎㅎ 씨 ㅂ ~~~ 싸 ㅇ~~~년 !!!
타. 형태소적 변형 비속어 (어미변화)	깝쳐, 깝치지 마
파. 문맥적 비속어 (문맥 안에서만 비속어인 것)	똘끼 짤리뿐다

게시물로부터 색인어를 만들어 내는 토크나이저는 이러한 패턴들을 모두 검출 해 낼 수 있도록 색인어를 만들어 내어야 할 것이다. 이렇게 고안된 토크나이저의 흐름을 [그림 2]에 나타내었다. 이 토크나이저 에서는 첫째로 <표 5>의 "다" 와 같은 한 음절씩 띄어쓰기 된 패턴을 인식하기 위하여 1차 토크나이징 시에는 한 음절 단어를 결합한다. 둘째로 "사"와 같은 경우가 있으므로 특수문자를 제거하며 게시물 본문 텍스트 상의 인덱스로 색인을 만든다. 셋째로 "마"또는 "카" 와 같은 경우에는 자모(자음, 모음)에서부터 음절로 복원하는 과정을 거치며 이 때 색인작업도 새로 한다.

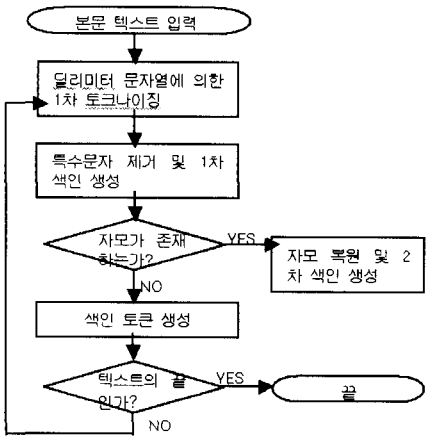
[2] http://www.dadatour.co.kr/news4.htm 를 참조하였음.

이렇게 색인 해 두면 패턴매칭 및 처리모듈에서 각 색인토큰에 대하여 색인토큰의 각 서브워드 의 첫 음소에 의해 비속어 리스트를 검색하여 매칭 워드 발견 시 1차 또는 2차 색인에 의해 원문 텍스트를 치환시키고, 변환버퍼에 발견된 비속어를 추가시킬 수 있게 된다. 이 때 각 스레드는 비속어 리스트를 공유메모리 상에서 참조한다. 이렇게 되면 비속어 리스트에 대한 사용이 집중되지만 각 스레드에서 현재 처리하고 있는 색인토큰의 서브워드의 첫 음소 별로 참조하는 비속어 리스트가 다를 확률이 높고 해시테이블과 바이너리 서치 기법을 사용하여 고속으로 검색하므로 집중도를 해소시킨다.

자모글자를 음절글자로 복원은 유니코드완성형의 한글자모코드("Hangul Compatibility Jamo:0x3130 ~ 0x318F)"와 한글 음절 코드("Hangul Syllables: 0xac00~0xd7a3)"를 상호 변환함으로써 한다[11]. 한글 음절 코드와 한글 자모 코드는 일정한 순서에 의해 배치되어 있으므로 수학적인 계산에 의해 쉽게 변환이 가능하다.

예를 들어 "ㄱ(0x3131) ㅐ(0x3150)"는 다음과 같은 알고리즘에 의하여 "개(0xac1c)"로 복원되어 진다.

```
char a = 0x3131;//ㄱ
char b = 0x3150;//ㅐ
char c =(char)(44032 + ((a-0x3130)*28*21
        + (b-0x314F)*28));
```



[그림 2] 토크나이저의 흐름

[Fig. 2] Flow of Tokenizer

3.4 패턴매칭 및 처리모듈

(1) 비속어 리스트

비속어 리스트는 기본적인 비속어 리스트는 미리 파일에 저장해 두고, 사용자의 입력에 의해 갱신되거나 추가될 수 있다. 비속어는 검색 시 효율성을 위해 초성의 첫 음소별로 구분하여 저장해 두었다. 파일의 저장구조는 <표 1>에서와 같이 비속어 단어에 등급을 부여하기 위하여 "단어$<단어등급>" 으로 표시하였다. 예를 들어 "개새끼$1", "개년$3"등이다. 이러한 파일을 차례로 읽어서 Hashtable의 형태로 메모리에 올린다. Hashtable은 해시기법을 이용한 키와 값의 쌍의 모음인데 일정한 수의 버킷을 테이블로 구성한 다음 한 개의 키를 해시테이블에 대한 인덱스로 변환시키는 해시함수를 이용하는 것이다. 해시테이블은 데이터항목의 수가 웬만큼 많아도 삽입과 탐색은 거의 O(1)이라는 일정시간에 가능하다. 해시 기법은 키를 이용하여 효과적으로 탐색할 수 있지만 키에 속한 데이터들이 순서대로 정렬되어 있지 않다는 커다란 결함이 있다[5, 10]. 그러나 Webcleaner에서는 자바의ArrayList에 sort 메소드를 결합하여 비속어 리스트의 데이터들을 순서대로 정렬해 놓았다. 이렇게 정렬되어 있기 때문에 [그림 2]에서 처럼 binary search알고리즘을 사용하여 데이터를 효과적으로 검색할 수 있다. Binary search검색은 N개의 데이터항목을 검색할 때 O(log2 N) 탐색시간의 효율을 낸다.

(2) 인코딩 방법의 선택

Java 언어의 String와 char타입은 한글을 읽어들이면 유니코드로 읽어들인다[16]. Webcleaner에서는 Java언어에서 특별한 코드 변환 없이 유니코드 완성형을 사용하여 문자적인 계산을 하였다.

비속어에는 "개쉽창년"의 "쉽" 과 같이 조합형에서만 지원되는 문자가 포함되는 경우도 있다. 그런데 각 시스템의 파일 시스템과 데이터베이스 시스템은 지원하는 한글 코드 형식이 다르다. 일반적인 시스템에서는 한글 조합형 보다는 완성형을 지원하는데 상용 웹 브라우저(인터넷 익스플로러, 네스케이프 네이게이터 등)의 경우도 그러하다. 상용 웹 브라우저에서는 웹 게시판 글 입력시 조합형 한글을 수용하지 못하거나 조합형 한글을 HTML에서 사용하는

특수문자 표기법인 "&#<유니코드 번호>;" 형식으로 입력받아서 요청에 전달해 준다. 그러므로 게시판 데이터베이스의 게시물에 저장되어 있는 형태도 게시판 프로그램에서 특정한 조치를 취하지 않는 한 그 형태로 저장되어 있다가 브라우저 상에 다시 보여질 때에는 HTML 특수문자 표기형식으로 인식되어 표시되게 된다[15]. 예를 들어 한글완성형에서는 지원되지 않는 "개쉽창년" 의 "쉽"과 같은 문자가 게시판 데이터베이스에 저장되어 있는 형태를 살펴보자. 비속어 입력 시 MS explorer와 같은 브라우저에서는"쉽"을 "쓉"로 변환하여 넘겨준다. 그러면 게시판 프로그램은 넘겨준 문자를 그대로 게시판 데이터베이스에 저장하고 있으며 그 정보가 다시 브라우저에서 보여지게 되면 "쉽"으로 보이게 된다. 그러므로 이렇게 저장되어 있는 데이터를 다루는 게시판 Webcleaner는 각 데이터베이스에 저장되어 있는 인코딩된 상황을 예측할 수 없게 된다. 그래서 Webcleaner에서는 기본적으로 유니코드 완성형으로 문자열을 비교 처리하지만 "&#<유니코드 번호>;" 형의 데이터 간에는 서로 비교가 가능하도록 프로그래밍 하였다.

(3) 추출 알고리즘

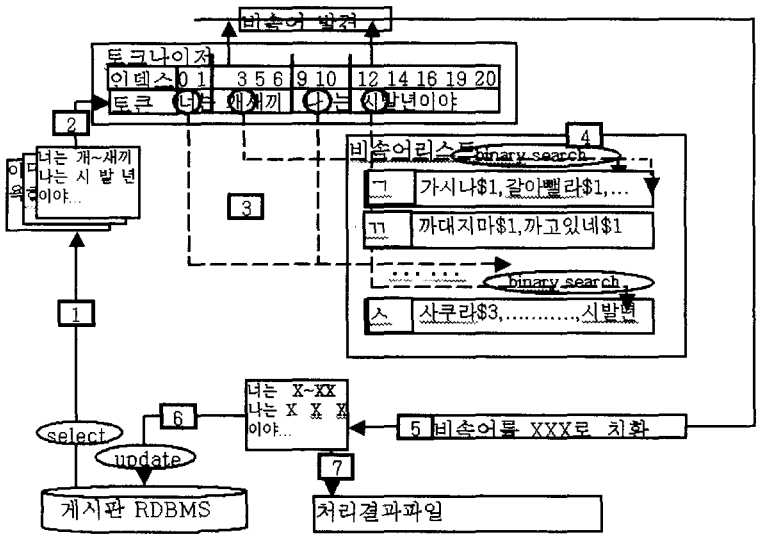
[그림 3]은 게시판 RDBMS로 부터 본문 텍스트들을 얻어서 비속어를 추출하여 치환된 텍스트를 얻는 과정인데 순서대로 설명하면 다음과 같다.

(1) 게시판RDBMS에 JDBC인터페이스를 사용하여 select 질의를 주어 가장마지막 처리한 게시물의 입력날짜 이후의 게시물 리스트를 읽어온다.

(2) 게시물의 본문테스트를 차례로 토크나이저에 넘겨준다. 그러면 토크나이저가 정의된 특수문자세트에 의해 본문텍스트를 한번 스캐닝하여 단어(토큰)별로 나누게 하고 각 토큰의 본문텍스트 상의 인덱스를 기록해 둔다.

(3) 비속어 추출모듈에서는 토크나이저에게 토큰을 차례로 넘겨줄 것을 요청하여 각 토큰의 첫음절 음소에 의해 주기억장치에 올라와 있는 비속어 리스트의 키를 결정하여 찾아간다.

(4) 찾아간 비속어 리스트 상의 특정리스트에서 토큰과 패턴이 매치되는 데이터가 있는지를 binary search기법으로 찾는다.



[그림 3] 비속어 처리모듈의 흐름

[Fig. 3] Flow of the swearing process module

(5) 이렇게 해서 본문텍스트의 끝까지 찾아서 비속어리스트와 패턴이 매치되는 토큰이 발견되었다면 토크나이저에게 비속어 부분을 치환시킨 텍스트를 요청하여 치환텍스트를 얻어낸다. 토크나이저는 토크나이징 할 때 본문텍스트 상의 토큰의 인덱스를 기록해 두었으므로 치환텍스트를 만들어 낼 수 있다.

(6) 게시판 RDBMS에 JDBC인터페이스를 사용하여 비속어가 발견된 게시물의 본문텍스트를 치환텍스트로 update하는 SQL문을 실행시켜 치환시킨다.

(7) 치환시킨 날짜별로 치환시킨 게시물에 대한 정보를 결과파일에 저장한다.

4. 평가

4.1 평가 방법

Webcleaner의 정확도와 재현율을 측정하기 위하여 다음과 같은 실험과정을 설정하였다.

(1) 웹 검색로봇을 이용해 인터넷 상의 욕게시판, 안티 게시판의 게시물 내용을 찾아서 HTML 문서를 수집 및 파싱한다. 이렇게 수집된 데이터를 게시판에 넣기 위하여 임의로 게시판 시스템을 만들어 둔다. 즉 게시물 아이디, 게시물 본문, 게시물 입력날짜 등의 게시판 형식을 갖춘 데이터베이스의 테이블을 만들어 둔다

(2) 수집된 데이터를 게시판 DB에 저장한다.

(3) 이렇게 수집된 게시판에 대하여 Webcleaner가 게시판 DB에 접속할 수 있도록 DB접속정보를 입력 해 주고 그 게시판에 대한 감시로봇을 작동시킨다. 이 때 감시로봇은 비속어를 추출하여 비속어 부분을 "XXX"의 문자열로 치환하도록 설정한다.

(4) Webcleaner의 작동이 끝나면 Webcleaner의 결과로그화면과 게시판 상의 치환된 게시물 데이터를 비교하면서 재현율과 정확도를 측정한다.

4.2 실험 결과

<표 6> 웹 게시판 데이터 수집 결과

<Table 6> The result of data collection from web boards

사이트 명	수집한 게시물 개수
http://my.netian.com/~seafar의 욕게시판	623
http://bbs.naver.com/freeboard/board.php?id=seo134_1의 정책비평 게시판	12
합계	635

<표 6>에서와 같이 두 개의 사이트의 게시판에 대하여 웹 검색로봇을 작동시켜 총 635개의 게시물 데이터를 수집하였다. 이 데이터를 임의로 만든 게시판 시스템의 데이터베이스의 "seafar 욕 게시판"과 "정책비평 욕 게시판"에 저장 하여 두었다.

<표 7> Webcleaner 작동 결과 발견된 비속어

<Table 7> The result swearings of running Webcleaner on the web boards

게시판 명	Webcleaner가 발견한 비속어
Seafar 욕 게시판	429/623건 (79.0% 비속어 발견확률)
정책비평 게시판	3/12건 (25.0% 비속어 발견확률)

<표 7>은 Webcleaner의 작동 결과이며 이 때 Webcleaner의 비속어 리스트에는 <표 1>에서와 같이 비속어가 총 981개가 등록되어 있었다. 이 두 개의 게시판에 대해 검색모드는 1등급,2등급, 3등급 모든 비속어 찾기, 처리모드는 문자 X로 자동치환으로 설정하여 Webcleaner를 작동시킨 결과 실행 경과시간은 22초가 나왔으며 <표 7>과 같이 seafar 욕 게시판에서 429개의 게시물 상에 나타난 비속어를 "XX"와 같은 문자로 치환하였으며, 정책비평 게시판에서 3개의 게시물 상에 나타난 비속어를 역시 "XX"와 같은 문자로 치환하였다.

<표 8> 관찰된 비속어 종류별 나타난 빈도수
<Table 8> The frequency of each kind of swearing by observing web boards

비속어의 종류	나타난 빈도수(게시물 개수)
품사적, 의미적 중의어	28/635(4.4%)
문맥적 비속어	38/635(5.9%)
형태소적 변형 비속어	14/635(2.2%)
미등록된 비속어	48/635(7.5%)

<표 8>은 Webcleaner 작동 후 Webcleaner의 결과 로그와 치환이 되어진 게시판을 동시에 관찰하여 얻어진 결과이다.

이 중 품사적, 의미적 중의어는 전체의 4.4%에 해당하는 28건 이었는데, 이들 모두 Webcleaner의 결과로그에서 발견할 수 있었다.이 중 비속어가 아닌데 치환한 경우는 다음과 같이 3건이 발견되었다. 이들은 전체 품사적, 의미적 중의어 28건 중에3건이므로10.7 %의 오류율을 보였다.그러나 이 수치는 대상 게시판이 달라짐에 따라 변화될 수 있는 수치이다.

(1) 'XX'를 분석해 봅시다. XX는 X+X로 이루어진 욕입니다. (졸라)
(2) 어XX구 (쩔라)
(3) 내려와XX 말라니깐 (보지)

문맥적 비속어는 각 단어로서는 비속어임을 알 수 없고 문장에서만 비속어임을 알 수 있거나 문맥 안에서만 비속어임을 알 수 있는 경우의 비속어를 말하는 데 다음은 그 중 몇 가지의 예이다.

(1) 대갈 진골 갈아서 사골탕 만들어뿐다.
(2) 쌈싸먹을 년이다
(3) 대가리 쑤시뿐다
-> XXX 쑤시뿐다

이러한 문맥적 비속어의 경우는 추출 해 내지 못하더라도 구문 상의 단어 중 비속어로 등록이 된 단어가 있을 시에는 부분적으로 치환이 되므로 게시물을 보았을 때에는 눈살을 찌푸릴 만한 비속어는 많지 않았다.

형태소 분석은 Webcleaner의 작동에 상당한 부하를 줄 수 있으므로 Webcleaner에서는 형태소 분석을 하지 않고 패턴매칭을 수행하였다.그러므로 형태소적 변형 비속어의 경우도 패턴에 있지 않으면 일일이 등록하여 검출하기가 쉽지 않다. 다음은 형태소적 변형 비속어의 경우의 예이다.

(1) 좆구린다, 좆꺼꾸로 물고, 니좆이나 빨러가

그리고 약한음, 강한음, 소리나는 대로 쓴 비속어 등 아직 Webcleaner에 등록이 되어 있지 않은 다음과 같은 비속어들이 발견되었다.

(1) 개땍끼햐
(2) 시박넘

이들은 Webcleaner의 비속어 리스트에 새로 등록 메뉴에서 추가하면 되므로 문제 시 되지 않는다.

5. 결론

실험적으로 Webcleaner를 작동시켜 본 결과를 종합하여 다음의 네 가지로 정리하였다.

첫째, 정확도는 품사적, 의미적 중의어 해석에 있어서의 오류 10.7%로 해석 해 볼 수 있으나 이것은 앞서 설명했듯이 측정 시 마다 달라질 수 있는 수치이므로 품사적, 의미적 중의어에 대한 적절한 조치가 필요하다. 이는 자주 나타나는 품사적, 의미적 중의어에 대하여 수동 삭제/치환 모드를 설정하는 방안을 생각 해 볼 수 있다.

둘째, 재현율은 형태소적 변형 비속어와 문맥적 비속어를 추출 해 내지 못했으므로 91.9%로 측정되었다. 그러나 패턴 분류에서 나타난 대부분의 경우를 검출 해 내었다. 즉, <표 4>에서 분류한 여러가지 비속어 패턴 중 형태소적 변형 비속어와 문맥적 비속어를 제외한 모든 경우에 검출 해 내었다. 그럼으로써 "개새끼"에 대한 유의어 "ㄱ ㅐ ㅅ ㅐ ㄲ ㅣ", "개 ㅅ ㅐ 끼", "개~새~~끼"를 등록하는 등 비속어의 유의어(변형어)를 일일이 등록할 필요가 없다. 한글조합형의 경우는 "미췬개쐽창눈"에서 "쐽"과 같은 한글조합형도 인식하고 있다.

셋째, 형태소적 변형 비속어의 경우는 각 형태소적 변형 비속어를 비속어 리스트에 새로 등록 하면 되지만 이러한 작업은 번거롭게 된다. 그러므로 부분적인 형태소 분석을 도입하든지 실질 형태소에 대해 검출하여 수동모드로 처리하도록 하는 방안을 생각 해 볼 수 있다. 즉, 수동삭제/치환 모드를 좀 더 세분화 하여 게시판 관리자의 판단 하에 비속어의 등급별로 적절한 처리를 할 수 있도록 만들면 좋을 것이다.

넷째, 문맥적 비속어에 대한 처리루틴은 아직 없는데 이를 처리하기 위해 구문분석이 필요하다. 이는 현실적으로는 구현이 어려우나 부분적으로 구문정보를 이용하면 효과적인 처리를 할 수 있다[5].

Webcleaner는 게시판 사용자들을 적절한 단어 사용에로 유도하며, 공공기관, 학교, 방송국 등의 게시판 관리자의 수작을 대체해 줄 수 있을 것으로 기대된다.

※ 참고문헌

[1] http://www.gamezone21.com/cover.html, 게임존 21 커버스토리, "저속어 표현에 필터링 강화, 과연 옳은 판단인가?", 2000/10/07

[2] http://freeonline.or.kr/index.html, 정보통신 검열 반대 공동행동

[3] http://mall.unicoop.co.kr/unn/campus/campus_read.asp?id=95&read=17&pagec=, 한국대학신문, "익명게시판 사용 86%가 찬성, 71%는 개선필요" , 2001/6/13

[4] 최유경, 안동언, 정성종 공저, "정보 검색용 다중 스레드 한국어 형태소 해석기", 2001 제 13회 한글 및 한국어 정보처리 학술발표 논문집

[5] 한국어 정보처리 연구소, "C로 구현한 인터넷 정보검색시스템", 도서출판 골-드, 1999, pp.29, pp.34-35, pp87-102

[6] 이동훈, 최범균 저, "JSP Professional", 가메출판사, 2001, pp 401-402

[7] 김영택외 10명 공저, "자연언어처리", ㈜교학사, 1994, pp.496-508

[8] Scott Oaks & Henrry Wong, "Java Thread", 한빛미디어, 2000, pp.22,

[9] James W. Cooper, "The Design Patterns Java Companion", Addison-Wesley Design Patterns Series, October 2, 1998

[10] Mitchell waite and Robert lafore 저, 박정호 역, "JAVA와 애니메이션 중심의 자료구조와 알고리즘", 이한출판사, 1999년 8월

[11] http://www.unicode.org/, "Code charts", Basic Latin, Hangul Compatibility Jamo, Hangul Syllables

[12] http://java.sun.com/, "APIs"

[13] http://industry.java.sun.com/products/jdbc/drivers

[14] http://jarkarta.apache.org/

[15] David Hunter 외 5인 공저, "Beginning XML", 정보문화사, 2000년 10월, pp56-65

[16] 이상로, http://trade.chonbuk.ac.kr/~leesl/code/concept.html, "문자셋의 개념 및 용어", 1997년 4월

[17] 박동혁, http://user.chollian.net/~iecrc/java/novice/3-1.htm, "자바의 문법-자바의 문자 표현"

조 아 영

2000. 2. 울산대학교 컴퓨터공학과 졸업
2000. 3.~ 현재 울산대학교 정보통신대학원 정보디자인 학과 재학
2000. 11. ~ 현재 ㈜ 자룩스(http://www.jalux.co.kr) 과장
현재 자룩스 에서는 웹 프로그래밍으로 우리농산물 지키기 참여연대 홈페이지 (http://www.kngy.org) 프로그래밍,
코리안 디지털 웨어 홈페이지(http://www.kodig.co.kr) 프로그래밍,
울산지역 검색사이트인 울산올(현재 운영중단)을 구축,
제품으로는 비속어 처리 프로그램인
Webcleaner v.1.0(데모: http://www.webcleaner.net)을 개발.