2001 년 한국정보처리학회 추계 학술발표논문집 제 8 권 제 2 호

웹 게시판 비속어 처리 프로그램의 설계 및 구현

조아영*, 옥철영
*울산대학교 정보통신대학원 정보디자인학과
*e-mail:daumahyoung@hanmail.net, okcy@mail.ulsan.ac.kr

Design and Implementation of a Slang Remover Program on Web board

Ah-Young Cho*, Cheol-Young Ock
*Dept of Information Design, University of Ulsan Graduated School of Information and Communication Technology

요 약

현재까지 게시판의 비속어 처리프로그램은 비속어를 발견하면 입력을 할 수 없도록 차단하는 차단식 프로그램이었다. 이런 프로그램은 사이버 상의 의사표현의 자유를 차단한다. 또한 어떤 단어의 경우는 비속어가 아닌데도 차단되어 입력을 원천봉쇄하기도 한다. 그래서 비속어를 차단하지 않고 처리해 주며 신생 비속어도 처리를 쉽게 해 주며 검출에 유연성을 제공하는 프로그램이 필요하다. 본 논문에서는 데이터베이스 상에서 구현된 게시판을 대상으로 비차단식, 유연성이 있는 비속어 추출 프로그램을 설계하고 구현하였다

1. 서론

인터넷 상에서는 국경을 넘어서서 익명으로 다양한 상대방에게 다양한 방식으로 의사를 전달할 수 있다. 그러나 이러한 표현의 자유와 더불어 비속어, 폭력적 언어의 무분별한 사용 및 유포행위의 기회도 열려 있다. 요즈음 인터넷 사용인구의 증가와 함께 이러한 행위가 늘어나고 있으며 이런 행위는 어린이와 청소년들에게도 자연스럽게 수용되곤 한다. 이에 대한 대응방안으로 여러 가지 제도와 내용선별 소프트웨어도 생겨났다. 그러나 이런 규제방식은 타율적이고 강제적이 될 때 네티즌의 강력한 항의에 부딪혀 왔다. 우리나라에서는 많은 네티즌이 게시판을 통해 의사소통을 한다. 그리고 게시판을 통한 비속어와 상업성의 글, 폭력적 언어도 많이 유통하고 있다.이에 대해 게시판의 비속어류를 추출하는 프로그램이 등장했는데 현재까지 우리나라에서 개발된 것들*1의 작동방식은 비.속어 DB 에 있는 단어가 발견되면 게시판에 글을 올릴 수 없도록 하는 프로그램으로 우리가 자주 접했던 채팅프로그램의 비.속어 차단방식과 같다. 채팅프로그램에서 글의 입력이 차단되는 것을 보고 답답해 하는 네티즌들은 차단을 피해갈 수 있는 비.속어를 만들어 낸다.[6] 이러한 반응은 게시판의 경우도 마찬가지일 것이다.그래서 이런 작동방식과는 다른 비속어 추출 프로그램이 필요하다고 보고 본 논문에서는 웹 게시판 DB 에 접근하여 이미 입력된 글에 대해 관리자의 설정모드에 따라 비속어를 추출하여 처리하는 프로그램을 JSP(Java Server Page)와 자바언어로 설계 및 구현한다. 앞으로 이 프로그램명은 Webcleaner 로 기술할 것이다.

*1 ㈜큐컴네트워크, 아름아리 BBS, http://www.cuecom.net
㈜아이모션, Slang Cleaner, http://www.imoxion.co.kr

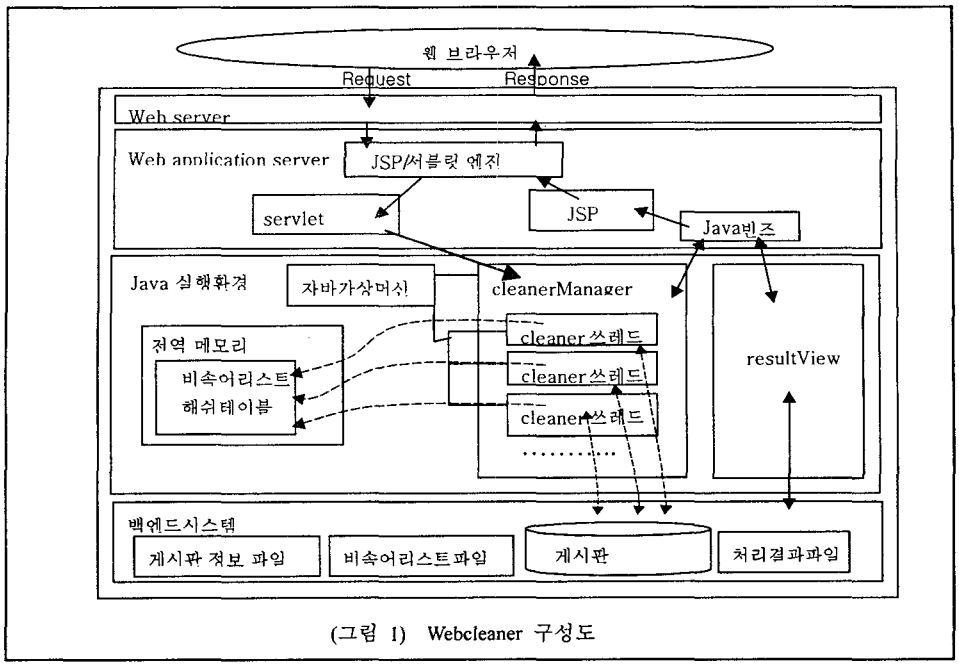
2. Webcleaner 의 요구사항 분석

1) 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 1 차 검색 후 2 차 삭제 또는 치환기능을 가진다.
2) 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.
3) 편리한 운영을 위해 동시에 여러 개의 게시판에 대해 추출작업을 수행할 수 있게 하며 예약반복수행기능과 날짜별 처리결과로그 기록을 한다.
4) 비속어 추출모듈은 자료구조 및 알고리즘 설계시 효율성을 중심으로 하여 시스템에 부하를 덜 주도록 한다.

3. Webcleaner 의 설계 및 구현

3.1 Webcleaner 구성개요

Webcleaner 는 프로그래밍 모델 중 MVC(Model-View-Controller) 디자인 패턴을 따라 개발하였으며[1] DB 에 접근방법은 DB 의 종류에 상관없는 JDBC(Java Database Connectivity)를 사용했다. Webcleaner 는 Java 언어로 작성되고 JDBC 를 사용하므로 어떤 플랫폼에서든지 실행되며 또한 어떤 RDBMS(Relational DataBase Management Systems)에서든지 JDBC 드라이버가 있다면 연결되며 Java 실행환경과 JSP/서블릿 엔진이 있으면 작동할 수 있다.



(그림 1) Webcleaner 구성도

(그림 1)은 Webcleaner 시스템의 구성도 이다. 그림의 Web server 와 Web application server 부를 보면 Http Request 를 서블릿이 받아서 자바 실행환경의 비즈니스 로직부분을 호출하며 Controller 의 역할을 하고 있고, 비즈니스 로직에서는 백엔드시스템과 통신한 결과를 Model 인 자바빈즈에 넘겨준다. 그러면 JSP 에서는 자바빈즈의 내용을 화면에 보여주는 View 의 역할을 하여 Response 를 보낸다. 백엔드시스템에는 게시판 DB 와 비속어리스트 파일, 결과를 저장하는 파일, 비속어를 처리할 게시판리스트 정보를 가지고 있는 게시판 정보파일이 있다. 특정 게시판에 대한 비속어처리를 시작하라는 Request 가 있을 때 비즈니스 로직부의 cleaner 쓰레드 중 그 게시판에 관한 쓰레드를 시작시키게 된다. cleaner 쓰레드는 자바가상머신에서 멀티쓰레드로 실행되면서 주기억장치의 비속어리스트를 공유하므로 효율적이며 각 게시판에 대한 동시수행이 가능하다.[2]

3.2. 비속어 추출/처리 모드의 정의

Webcleaner 는 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 단어의 등급

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

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

<표 3> 비속어 처리모드

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)

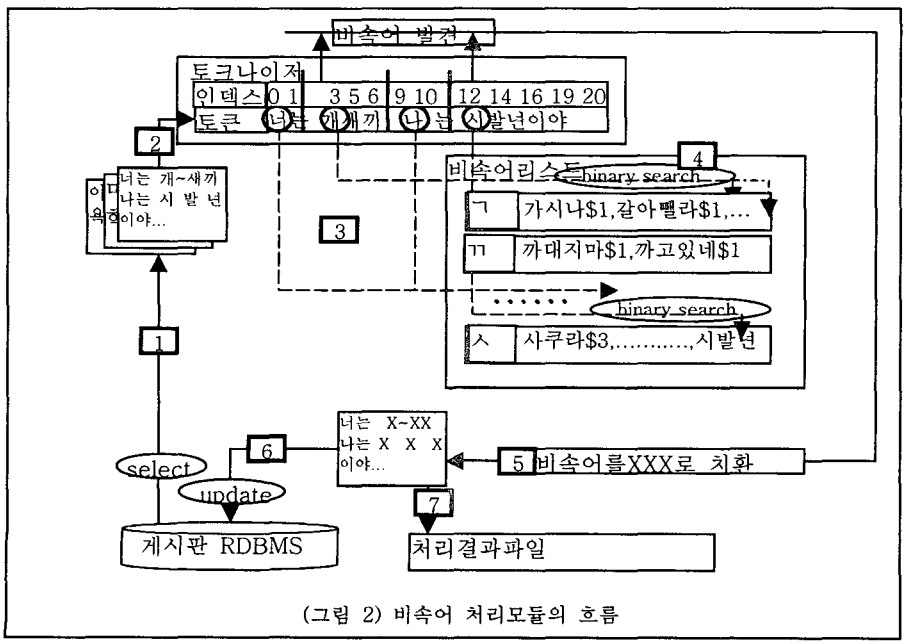
비속어를 처리하는 모드로는 크게 자동모드와 수동모드가 있는데 수동모드의 경우 1 차로 검색만 하고 2 차로 검색된 결과 내에서 검색모드를 다시 취하여 재검색 해 본 후에 관리자가 삭제 또는 치환결정을 내릴 수 있게 한다.이 모드는 문장 속에서 문맥에 따라 비속어가 될 수도 있고 아닐 수도 있는 중의적 단어에 대해서 특별히 쓰일 수 있다. 그리고 처리와 처리결과의 저장을 위해서 다음 세 가지의 텍스트 유형을 정의한다.

(1) 본문텍스트: 게시물 원본 텍스트
(2) 변환텍스트: 본문 텍스트에서 비속어를 찾은 곳을 비속어의 레벨별로 특정색으로 표시하여 놓은 텍스트
(3) 치환텍스트: 본문 텍스트에서 비속어가 발견된 곳

을 치환문자열로 치환한 텍스트
처리모드가 삭제의 경우는 본문텍스트에서 한 개의 비속어 단어를 발견하면 비속어 검출을 멈추고 DBMS 로 delete 질의를 보내어 그 게시물을 삭제하며, 치환의 경우는 본문텍스트의 끝까지 비속어 단어를 모두 찾아서 사용자가 지정한 문자로 치환한다. 예를 들어 개새끼는 XXX 로 치환한다. 그리고 RDBMS 로 치환텍스트로 update 질의를 보낸다. 수동인 경우는 결과를 1 차 파일에 저장해 두었다가 2 차로 사용자의 요구시 RDBMS 로 updte/delete 질의를 보내어 처리한다.

3.3 인코딩 방법의 선택
게시물 본문 텍스트에서 비속어를 검색하기 위해 자바에서 기본적으로 지원하는 인코딩인 KSC5601 로 인코딩 하였다. 한글완성형에서는 지원되지 않는 " 똠 방각하" 의 " 똠" 이나 " 개쉽창년" 의 " 쉽" 과 같은 문자들이 게시판 데이터베이스에 저장되어 있는 형태는 비속어 입력시 브라우저에서 넘겨주는 방식에 따라 다른데 예를 들면 MS explorer 와 같은 브라우저에서는 " 똠" 을 " 똠" 로 변환하여 넘겨준다. 그러면 게시판 프로그램은 넘겨준 문자를 그대로 게시판 데이터베이스에 저장하고 있으며 그 정보가 다시 브라우저에서 보여지게 되면 " 똠" 으로 보이게 된다.그러므로 이렇게 저장되어 있는 데이터를 다루는 게시판 비속어 추출 프로그램은 각 데이터베이스에 저장되어 있는 인코딩된 상황을 예측할 수 없다. 그러나 웹 게시판은 위의 예에서와 같이 웹상의 특수문자 표기법인 &#숫자;의 형태의 데이터를 가지고 있는 경우는 고려할 수 있다. 그래서 기본적으로 KSC5601 로 인코딩하며 &#숫자; 형의 데이터 간에는 서로 비교가 가능하도록 프로그래밍 하였다.

3.4 빠른 검색을 위한 비속어 리스트 자료구조 및 알고리즘 설계
비속어 리스트는 기본적인 비속어 리스트는 미리 파일에 저장해 두고, 사용자의 입력에 의해 갱신되거나 추가될 수 있다. 비속어는 검색시 효율성을 위해 초성의 첫 음소별로 구분하여 저장해 두었다. 파일의 저장구조는 <표 1> 에서와 같이 비속어 단어에 등급을 부여하기 위하여 단어에$<단어등급>으로 표시하였다.예를 들어 개새끼$1, 개년$3 등 이다. 이러한 파일을 차례로 읽어서 Hashtable 의 형태로 메모리에 올린다. Hashtable 은 해시기법을 이용한 키와 값의 쌍의 모음인데 일정한 수의 버킷을 테이블로 구성한 다음 한 개의 키를 해시테이블에 대한 인덱스로 변환시키는 해시함수를 이용하는 것이다. 해시탐색은 배열로 구성된 버킷 테이블(해시테이블)과 해시테이블에 데이터를 체인처럼 가리키는 링크드리스트의 구조가 필요하다. 해시 기법은 키를 이용하여 효과적으로 탐색할 수 있지만 키에 속한 데이터들이 순서대로 정렬되어 있지 않다는 커다란 결함이 있다.[4] 그래서 별도의 정렬과정을 추가하여 보완하는 방법을 사용하기도 하는데 Webcleaner 에서는 자바의 ArrayList 에 sort 메소드를 결합하여 비속어 리스트의 데이터들을 순서대로 정렬해 놓았다.이렇게 정렬되어 있기 때문에 (그림 2)에서 처럼 binary search 알고리즘을 사용하여 데이터를 효과적으로 검색할 수 있다. (그림 2)는 게시판 RDBMS 로 부터 본문 텍스트들을 얻어서 비속어를 추출하여 치환된 텍스트를 얻는 과정인데 순서대로 설명하면 다음과 같다.
(1) 게시판 RDBMS 에 JDBC 인터페이스를 사용하여 select 질의를 주어 가장마지막 처리한 게시물 의 입력날짜 이후의 게시물 리스트를 읽어온다.



(그림 2) 비속어 처리모듈의 흐름

(2) 게시물의 본문태스트를 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 search 기법으로 찾는다.
(5) 이렇게 해서 본문텍스트의 끝까지 찾아서 비속어 리스트와 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 RDBMS 에 JDBC 인터페이스를 사용하여 비속어가 발견된 게시물의 본문텍스트를 치환텍스트로 update 하는 SQL 문을 실행시켜 치환시킨다.
(7) 치환시킨 날짜별로 치환시킨 게시물에 대한 정보를 결과파일에 저장한다.

4. 평가 및 결론

<표 4> Webcleaner 실험: 비속어를 XX 로 치환한 결과

Webcleaner 가 XX 로 치환후	치환 전
X X~~ X X~!! XXX 같은년이 설치고 다니나 XX 쌈싸먹을 년이다 XXXXXXXXXXXX 뇬 2001 년 X X X 들 @@X@@X@@!!! X#X#아	ㅆ ㅣ~~ ㅂ ㅏ~!! 씹보지같은년이 설치고 다니나 보지 쌈싸먹을 년이다 미쿈개쒑창뇬 2001 년 개 새 끼 들 @@지@@랄@@!!! 개#년#아

<표 4>는 테스트용 게시판 RDBMS 에 Webcleaner 를 작동시켜 본 결과이며 검색모드는 1 등급,2 등급, 3 등급 모든 비속어 찾기, 처리모드는 문자 X 로 자동치환일 때의 예시 데이터이다. 비속어 리스트에는 <표 1> 에서와 같이 비속어가 총 981 개가 등록되어 있는 상태이다. <표 4>에서 볼 수 있듯이 특수문자 " ~", " !", " @", " #" 사이의 비속어를 찾아내며 " 미쿈개쒑창뇬" 에서 " 쒑" 도 인식하고 있다. 그러나 씹보지" 의 " 보지" 와 " 다니나 보지" 의 " 보지" 를 구분하지 못하고 모두 XX 로 치환한다. 또한 " 쌈싸먹을 년" 을 검출해 내지 못한다. 현재 비속어 리스트에 "년" 은 등록되어 있지 않다. 자지나 보지, 년 같은 문맥상 비속어가 될 수도 있고 아닐 수도 있는 비속어를 처리하려면 패턴매칭 방식으로만 검출해서는 안되고 이러한 중의적 비속어에 대해서 형태소 분석모듈을 추가시킬 것이 필요하다. 그런데 일반적으로 형태소 사전에 수록된 어휘의 수가 많을수록 형태소 분석의 처리범위가 넓어지는 반면에, 모호성이 증가하여 분석의 정확성이 낮아지기 쉽다. 왜냐하면, 어휘의 수를 늘리면 분석 가능성이 많아져서 모호성이 더 많이 발생하기 때문이다[3]. 이러한 점을 고려하여 수동삭제/치환 모드를 좀 더 세분화 하여 게시판 관리자의 판단하에 비속어의 등급별로 적절한 처리를 할 수 있도록 만들면 좋을 것이다. 그리고 문장으로 된 비속어에 대한 처리루틴은 아직 없는데 이를 처리하기 위해 구문분석이 필요하며 이는 현실적으로 구현이 어렵다. 그러나 부분적으로 구문정보를 이용하면 효과적인 처리를 할 수 있다.[4] 또한 " 씨바" 에 대한 유의어 " ㅆ ㅣ ㅂ ㅏ" 를 등록하는 등 비속어의 유의어(변형어)를 일일이 등록해야 하는 불편함이 있는데 이에 대해서는 유의어 확장사전 생성기도 고려해 볼 수 있다.

참고문헌

[1] 이동훈, 최범균 저, " JSP Professional" , pp 401-402, 가메출판사, 2001
[2] Scott Oaks & Henrry Wong, " Java Thread" , 한빛미디어, pp.22, 2000
[3] 김영택외 10 명 공저, " 자연언어처리" , ㈜교학사, pp.215, 1994
[4] 한국어 정보처리 연구소, " C 로 구현한 인터넷 정보검색시스템" , pp.29, pp.34-35 도서출판 골-드, 1999

참고사이트

[5]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
[6] http://www.gamezone21.com/cover.html, 게임존 21 커버스토리, " 저속어 표현에 필터링 강화, 과연 옳은 판단인가?" , 2000/10/07
[7] http://freeonline.or.kr/index.html, 정보통신 검열반대 공동행동