포스트의 구조 유사성과 일일 발행수를 이용한 스플로그 탐지

(Splog Detection Using Post Structure Similarity and Daily Posting Count)

백 지 현 † 조 정 식 †† 김 성 권 †††
(Jee-Hyun Beak) (Jung-Sik Cho) (Sung Kwon Kim)

요 약 블로그는 웹과 로그의 합성어로, 개개인의 생각이나 관심사 등을 일기처럼 기록할 수 있는 웹 서비스이다. 블로그에는 문자 외에, 그림이나 비디오 파일 등 다양한 컨텐츠를 올릴 수 있다. 일반적으로 블로그의 포스트는 시간상의 역순으로 정렬되어 표현된다. 블로그 검색 엔진은 웹 검색 엔진처럼 블로그를 대상으로 사용자의 질의에 따라 정보를 찾아주는 서비스이다. 블로그 검색 엔진은 때때로 만족스럽지 못한 결과를 내곤 하는데, 이것은 스플로그라고 불리는 블로그 스팸에 의해 발생한다. 스플로그는 다른 블로그나 웹 페이지를 무단 도용하거나 자동으로 생성된 컨텐츠로 구성된 스팸 포스트를 가지고 있다. 스플로그는 검색 엔진의 검색 순위를 높이거나, 회원 가입 사이트로 보다 많은 사람들을 유치하기 위해 사용된다. 본 논문은 스플로그 탐지를 목적으로 한다. 본 논문에서 제안하는 스플로그 탐지 기법은 블로그 포스트의 구조 유사성과 일일 포스트 발행수에 따른 분석으로 토대로 이루어진다. 본 논문에서 제안하는 기법을 바탕으로 한 실험의 결과, 스플로그 탐지에 있어 90% 이상의 높은 정확도를 가지며, 만족할만한 수준을 보여준다.

키워드 : 웹, 블로그, 스플로그, 웹 스팸

Abstract A blog is a website, usually maintained by an individual, with regular entries of commentary, descriptions of events, or other material such as graphics or video. Entries are commonly displayed in reverse chronological order. Blog search engines, like web search engines, seek information for searchers on blogs. Blog search engines sometimes output unsatisfactory results, mainly due to spam blogs or splogs. Splogs are blogs hosting spam posts, plagiarized or auto-generated contents for the sole purpose of hosting advertizements or raising the search rankings of target sites. This thesis focuses on splog detection. This thesis proposes a new splog detection method, which is based on blog post structure similarity and posting count per day. Experiments based on methods proposed a day show excellent result on splog detection tasks with over 90% accuracy.

Key words : Web, Blog, Splog, Web Spam

† 비 회 원 : 중앙대학교 컴퓨터공학과
jhbaek@alg.cse.cau.ac.kr
†† 학생회원 : 중앙대학교 컴퓨터공학과
mfg@alg.cse.cau.ac.kr
††† 종신회원 : 중앙대학교 컴퓨터공학과 교수
skkim@cau.ac.kr
논문접수 : 2009년 2월 27일
심사완료 : 2009년 11월 9일

정보과학회논문지: 소프트웨어 및 응용 제37권 제2호(2010.2)

1. 서 론

인터넷의 정보가 크게 증가함에 따라 인터넷에서 필요한 정보를 쉽게 찾는 다는 것은 중요한 문제이며, 이를 위해 검색 엔진이 개발되게 되었다.

검색 엔진의 사용이 증가할수록, 검색 엔진은 인터넷 사용자들에게 점점 더 많은 영향력을 끼치게 되었다. 이는 상업적인 웹 사이트를 운영하는 사람들에게 특히 중요한 이슈였다. 상업적인 웹 사이트를 운영하는 사람들은 그들의 회사의 발전을 위해 많은 사람들에게 자신들의 웹 사이트를 보여야만 했고, 검색 엔진 리스트에서 자신들의 웹 사이트가 좀 더 높은 순위에 오르는 것이

중요하게 되었다.

이러한 이유로 인해, 전문적으로 검색 엔진의 순위를 올리려는 사람들이 생겨나게 되었는데, 이들을 SEO (Search Engine Optimizer)라 부른다. SEO들은 웹 사이트 개발자들이 잘 구조화되고, 주제에 맞게 키워드가 달린 컨텐츠를 개발할 수 있도록 도움을 주는 합법적인 방법으로 검색 엔진의 순위를 높인다.

하지만, 모든 SEO들이 합법적인 방법을 사용하는 것은 아닌데, 합법적인 방법을 사용하는 SEO를 White-hat SEO라 부르는 것에 반하여 그렇지 않은 SEO들을 가리켜 Black-hat SEO라 부른다. Black-hat SEO들은 웹 스팸(Web Spam)이라 부르는 악의적인 방법을 사용하여 검색 엔진의 순위를 높이려고 한다[1]. 이런 악의적인 정보 검색 활동, 즉 웹 스팸은 검색 엔진 순위의 신뢰성을 떨어뜨리고 불필요한 트래픽을 조장하여 검색 엔진 사용자들에게 정상적인 정보 검색 활동을 방해한다.

초창기의 웹 스팸은 웹 사이트에만 국한되어 발생하였다. 이 후, 소셜 네트워크(Social Network)의 발달로 블로그의 사용이 크게 증가함에 따라 블로그에서도 웹 스팸이 나타나게 되었고, 현재 가장 심각한 웹을 통한 문제점들 중의 하나가 되고 있다.

본 연구는 블로그 스팸의 탐지를 목적으로 하고 있다. 블로그 스팸은 블로그 검색시 검색 결과의 신뢰성을 떨어드리고, 불필요한 트래픽을 야기하여 효율적인 검색 활동을 방해한다. 이에 블로그 검색 엔진에서 적용 가능한 블로그 스팸 탐지 기법을 개발하여 블로그 검색시 보다 양질의 검색 결과를 구하려고 한다.

본 논문은 다음과 같이 구성되어 있다. 1장은 논문의 개요로써, 연구의 배경이 되는 인터넷과 연구 동기를 기술하고 있다. 2장은 웹 스팸에 대한 관련 연구를 제공한다. 2장에서 다루는 내용은 웹 스팸, 블로그, 블로그 스팸 등이며, 기존의 블로그 스팸 탐지 기법의 설명도 포함한다. 3장은 본 논문에서 제안하는 기법에 대해 설명한다. 본 논문에서는 블로그 스팸 탐지에 대한 두 가지 척도를 제안하며, 각각의 대한 계산 방법을 포함하고 있다. 4장은 논문에서 제안하는 기법에 대한 실험 결과를 나타낸다. 5장에서 본 논문의 성과를 요약하며 결론을 맺고자 한다.

2. 관련 연구

2.1 악의적인 정보 검색(Adversarial Information Retrieval)

정보검색은 정보를 수집, 분류, 축적하여 필요에 따라서 빼내는 작업이다. 정보검색의 기본적인 문제는 검색 결과가 사용자의 요구에 부합하지만은 않다는 것이다. 또한 경우에 따라 악의적으로 조작된 검색결과가 제시되기도 한다. 악의적인 정보검색 활동이란 검색 엔진을 속여 공격자가 원하는 검색 순위를 만들어 내거나 불필요한 정보를 사용자에게 배포하는 것을 말한다. 악의적인 정보검색 활동은 사용자에게 잘못된 정보를 제공하고 검색엔진에게 쓸모없는 페이지 처리비용을 늘린다. 결국 검색 엔진의 검색 결과의 신뢰성을 떨어뜨리고, 불필요한 네트워크 트래픽을 야기한다. 일반적으로 검색 엔진 스팸(Search Engine Spam)이나 웹 스팸(Web spam)이라고 불리는데 대부분 금전적인 이익을 위해 이와 같은 활동을 하고 있다.

2.2 블로그(Web log, blog)

블로그(Blog 혹은 Web log)는 웹(Web)과 로그(Log)를 합성어로, 스스로의 느낌이나 생각, 알리고 싶은 견해나 주장 같은 것을 웹에 일기 형식으로 적어 다른 사람들이 볼 수 있도록 게시하는 글 모음이다. 대개 시간 순으로 가장 최근의 글부터 보이게 되며, 여러 사람이 글을 게시할 수 있는 게시판과는 달리, 개인이나 소수의 사람에 의해서만 작성되어 진다. 이런 이유로 인해 블로그를 '1인 미디어'라고 부른다. 또한, 블로그를 소유한 사람을 일컬어 블로거라고 한다[2]. 블로그는 웹 문서 작성과 배포의 편리함으로 인해 빠른 속도로 사용자 수 증가하고 있다.

처음에 나온 블로그에는 단순히 텍스트만 게시할 수 있었으나, 이후 사진, 음악, 플래시 무비(Flash Movie), 동영상 등을 포함할 수 있도록 발전하였다. 또한 블로그에 댓글(Reply)과 트랙백(Trackback)을 달 수 있게 함으로써 블로그 독자들과의 의사소통이 가능하도록 발전되었으며, RSS(Really Simple syndication)나 Atom으로 손쉽게 블로그 포스트를 구독할 수 있도록 하는 서비스를 제공하기도 한다.

2.3 블로그 스팸(Blog Spam)

2.3.1 블로그 스팸의 정의

블로그 스팸은 블로그 내에서 발생하는 악의적인 정보검색 기법을 말한다. 블로그 스팸 역시 웹 스팸의 발생 원인인 검색 엔진의 순위를 높이거나 회원 가입을 조장하기 위해 만들어졌는데, 블로그의 중요성이 증가함에 따라 그 문제점 역시 증대되고 있다.

근래 들어, 블로그 스팸은 구글의 애드센스로 대표되는 PPC(pay-per-click) 광고의 활성화로, 애드센스의 광고 수익을 얻기 위한 목적으로도 사용되고 있다.

2.3.2 블로그 스팸의 분류

블로그 스팸은 크게 2가지로 나뉘어지는데, 하나는 스패머들에 의해 검색 엔진의 순위를 높이기 위해 만들어진 블로그이며, 다른 하나는 이미 만들어진 일반 사용자의 블로그에 댓글을 달거나 트랙백을 연결하는 방식으로 악의적인 정보검색 활동을 하는 것이다. 첫 번째 경

우를 '스팸 블로그(Spam Blog)'라고 하고, 두 번째 경우를 '스팸 인 블로그(Spam in Blogs)'라고 부른다. 스팸 블로그는 '스플로그(Splog)'로 불려진다. 이하 스팸 블로그의 명칭을 스플로그로 통일한다.

스플로그와 스팸 인 블로그의 가장 큰 차이점은 블로그의 소유자에 있다. 스플로그의 경우 해당 블로그의 소유자가 스패머이지만, 스팸 인 블로그의 경우 블로그의 소유자는 스패머가 아닐 수 있다. 그렇기 때문에 2가지 경우에서의 블로그 스팸의 처리 방식이 다르다.

스플로그는 대개 다른 웹 사이트나 블로그의 컨텐츠를 그대로 가져와 자신의 사이트나 블로그에 게재하는 스크래퍼 사이트(Scraper Site)의 형태를 취하며, 때로는 불확실하거나 의미 없는 내용을 포함하기도 한다[3]. 그림 1은 스플로그의 한 예이다.

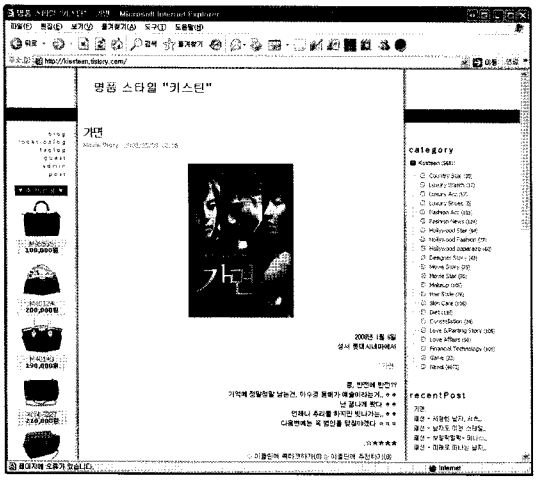

그림 1 스플로그의 예

스팸 인 블로그는 검색 엔진의 순위를 높이기 위해 웹 사이트의 방명록에 하이퍼링크를 반복해서 기록하는 것에서부터 시작되었다. 이후 초기의 블로그에서 쉽게 댓글을 달 수 있었던 특성 때문에, 웹 사이트의 방명록에 스팸 메시지를 작성하던 스패머들은 블로그에 스팸 메시지를 작성하게 되었다[4].

2.4 기존 스플로그 탐지 기법

2.4.1 SVM을 이용한 스플로그 분류

Pranam Kolari 등은 그들의 2006년 논문에서 블로그 포스트 컨텐츠의 여러 가지 특성을 추출하여 SVM 이용해 스플로그를 탐지하는 방법을 제시하였다. 그들이 SVM에 사용한 특성들은 Bag-of-Words, N-Grams, URL Tokens, Anchor Text등이다[5].

스플로그 탐지에 사용되는 Bag-of-Words 기법은 해당 블로그 포스트의 텍스트에 포함된 단어들을 이용해 집합을 만든 후, 이 집합에 포함된 단어들이 스플로그와 정상 블로그 어느 쪽에 더 많이 나타나는지를 평가한다. 이와 유사한 방법으로 N-gram 기법을 들 수 있는데, 이는 문장의 단어들을 n개씩 나누어 분석한다. E-mail의 스팸 메일 필터링에도 사용되고 있다.

URL Tokens와 Anchor Text 특성은 블로그 포스트 중 앵커 태그만을 추출하여 평가한다. 앵커 태그는 화면에 보여지는 텍스트와 하이퍼링크를 포함하는데, 이 중 하이퍼링크에 해당하는 부분이 URL Tokens이고 텍스트에 해당하는 부분이 Anchor Text이다. 블로그 포스트의 앵크 태그의 텍스트와 하이퍼링크를 단어 단위로 구분 추출하여 앞서 보인 Bag-of-Words 기법을 적용한다.

SVM(Support Vector Machines)은 1960년대에 Vapnik 등이 제안한 통계 분류와 회귀 분석에 사용되는 지도 학습 방법을 가리키는데, 커널 함수라 불리우는 트릭을 사용해 비선형 분류 문제를 선형 분류 문제로 변환하여 분류 작업을 수행한다. SVM은 현재 알려져 있는 많은 수법 중에서 가장 인식 성능이 뛰어난 학습 모델 중의 하나이다[6].

2.4.2 시간적 자기 유사성 분석을 이용한 탐지

Yu-Ru Lin 등은 그들의 2007년 논문에서 스플로그 탐지를 위해 시간적 자기 유사성(Self-Similarity) 분석의 사용을 제안하였다. 그들이 제시한 자기 유사 분석은 블로그 내의 포스트들 간의 관계를 분석한 것이다. 블로그의 포스트들은 기본적으로 시간적 순서를 지니는데, 이러한 특성을 바탕으로 기준 포스트에서 이후에 게시된 포스트로의 게시 시각, 포스트 컨텐츠 내용 그리고 포스트 내의 아웃 링크 등의 속성들이 얼마나 차이를 보이는지 비교한다. 또한, 저자들은 컨텐츠와 링크의 변화가 동시에 일어남을 발견하고, 두 특성에 대해 결합 분석도 실험하였다[7].

2.4.3 메타-핑 서버를 이용한 접근

Pranam Kolari 등은 핑 스팸을 효과적을 탐지할 있는 메타-핑 서버 시스템을 제안하였다. 그들에 따르면 핑 서버에서 75% 이상이 스팸이며, 검색 엔진 결과의 20% 정도가 스팸으로 판단되었다고 한다[8].

Pranam Kolari 등이 제안한 메타 핑 서버 시스템에서는 일련의 과정을 거쳐 스플로그 여부를 평가한다.

우선 핑 서버가 핑을 받으면 받아온 핑을 Meta-Ping Detection Module으로 보내 스팸 여부를 확인하는데, 이 때 거치게 되는 단계는 URL 기반 필터링(URL based filtering), 블랙리스트 기반 필터링(Blacklist based filtering), 블로그 홈페이지 기반 필터링(Blog home-page based filtering), 배포 피드 기반 필터링(Syndication feed based filtering)이다.

첫 단계인 URL 기반 필터링에서는 URL의 길이나

특수 문자 사용 여부, 혹은 URL 블랙리스트를 사용하여 스플로그를 걸러내게 된다. 두 번째 단계인 블랙리스트 기반 필터링에서는 스팸 도메인이나 IP 주소의 목록을 만들어 적용한다. 세 번째 단계는 블로그 홈페이지 기반 필터링으로 이 단계에서는 블로그의 홈페이지의 내용을 분석하여 스플로그를 탐지하게 되는데, 이 단계에서 핑을 내보내는 곳이 블로그인지 아닌지의 여부와 블로그의 언어 등을 알 수 있다. 마지막 단계는 배포 피드 기반 필터링이다. 마지막 단계에서 블로그 포스트 간의 텍스트와 하이퍼링크의 시간적 상관관계를 분석하게 된다.

3. 제 안

일반적인 스플로그의 포스트들은 특정 상업적 의도로 프로그램에 의해 자동 생성된 것들이 대부분이다. 프로그램에 의해 자동 생성된 블로그 포스트들은 대체로 유사한 구조를 가진다. 또한 스플로그는 매우 짧은 시간 동안에 대량의 포스트를 생성한다. 이는 블로그 검색 엔진의 검색 순위 결정 방식에 기인한 것으로, 일반적으로 블로그 검색 엔진은 웹 검색 엔진과는 다르게 검색어의 적합성과 포스트의 최신성을 바탕으로 검색 순위를 결정한다.

이와 같은 스플로그의 두 가지 특성을 이용하여 본 논문에서는 스플로그를 탐지하려 한다.

3.1 블로그 포스트의 구조 유사성 비교

일반적으로 스플로그의 포스트는 프로그램에 의해 자동 생성된 경우가 대부분이며, 스플로그 포스트들 간에는 구조 유사성이 크다. 그림 2는 스플로그의 한 예이다. 그림 2와 같이 스플로그 내 포스트들은 유사한 구조로 구성되어 있고, 대개는 플래쉬 파일이나 동영상, 혹은 그림 파일을 포함하고 있다.

```
<div>
    <center>
        <script>
            //<![CDATA[writeCode2("<object classid='clsid:d27cdb6e-ae6d-11cf-96
        </script>
    </center>
    <br>
    <div>
        <img>
    </div>
    <br>
</div>
```

그림 3 그림 2의 스플로그 포스트의 HTML 코드

그림 3은 그림 2의 각각의 포스트 중 본문만을 추출하여 HTML 구조를 나타낸 것이다. 그림 2의 경우 각각의 포스트가 다르게 보이지만, 그 구조를 그림 3의 하나의 HTML 구조를 가진다.

3.1.1 블로그 포스트 구조 비교 절차

본 논문에서는 블로그 포스트의 구조를 비교한다. 블로그 포스트는 HTML 구조로 표현 가능하다. 프로그램에 의해 자동 생성된 블로거의 경우 포스트의 HTML 구조가 매우 유사하다.

HTML은 깊이 우선 탐색 트리로 표현 가능하다. 그림 4는 그림 3을 트리 구조로 표현한 그림이다.

트리의 루트를 제외한 루트 노드에서 각각의 자식 노드까지의 링크 정보를 깊이우선순위 탐색 방식으로 구한다. 예를 들어, 그림 4에서 'center 노드'의 정보는 'div-center'가 되고, 'img 노드'의 정보는 'div-div-img'가 된다. 또한, 각각의 노드의 순서는 'center -> script

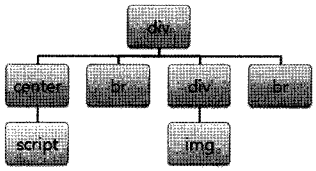

그림 4 그림 3의 스플로그의 트리 구조

그림 2 스플로그 포스트의 유사성

표 1 블로그 포스트 구조 비교 의사코드

```
function GetNodeList() {
    for ( 전체 트리 노드에 대해 ) {
        GetNodeInfo();
    }
}

function GetNodeInfo( ID ) {
    NodeInfo <- NodeInfo + 노드명;

    if ( 부모 ID != -1 ) {
        GetNodeInfo( 부모 ID );
    }
}
```

-> br -> div -> img -> br' 순이다. 포스트의 본문에 대한 모든 노드의 정보를 구한 후, 다른 포스트의 본문에 대한 노드 정보들과 각각 비교한다.

표 1은 이에 대한 의사코드이다. 트리의 각 노드들에 대한 ID, 노드명, 부모 ID에 대한 정보는 미리 알고 있다고 가정한다. 또한 루트 노드의 부모 ID는 -1이라 가정한다.

블로그 내의 포스트들은 하나의 포스트를 패턴으로, 나머지 포스트들을 테스트 대상로 설정한 후 각각의 테스트 대상들이 패턴과 얼마만큼 일치하는지를 검사한다.

이 때 브라우저를 통해 보여지는 HTML 구조는 깊이우선순위 탐색 방식으로 보여지기 때문에, 포스트 본문의 각각의 노드 비교 역시 노드 정보의 순서를 따른다. 즉, 그림 4의 깊이우선순위 탐색 방식으로 세 번째로 탐색되는 'br 노드'에 대해 일치한다는 판단이 되었다면, 'br 노드' 이전에 나온 'center 노드'와 'script 노드'는 더 이상 비교하지 않는다.

예를 들어, 비교하고자 하는 두 포스트의 HTML 구조가 그림 5와 같다고 가정하자. 이 경우 그림 5의 왼쪽의 트리를 패턴으로 설정하고, 오른쪽의 트리를 테스트 대상이라고 설정한다.

그림 5의 양쪽 트리에서 각각의 노드 정보를 구하면 표 2와 같다. 표 3과 표 4는 양 트리의 각 노드들을 비교한 것이다. 표 2에서 왼쪽 트리의 ID 1인 'div-center' 노드 정보는 오른쪽 트리의 ID 1에 존재하기 때문에 각 트리의 일치 노드에 값이 입력한다. 왼쪽 트리의 ID 2인 'div-center-script' 노드 정보 역시 오른쪽

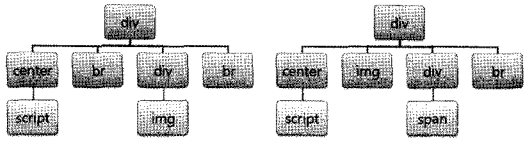

그림 5 두 트리 구조의 비교

표 2 두 트리 구조의 비교

ID	왼쪽 트리	오른쪽 트리
1	div-center	div-center
2	div-center-script	div-center-script
3	div-br	div-img
4	div-div	div-div
5	div-div-img	div-div-span
6	div-br	div-br

표 3 패턴 트리에 대한 테스트 트리 비교 (1)

왼쪽 트리			오른쪽 트리		
ID	노드	일치 노드	ID	노드	일치 노드
1	div-center	1	1	div-center	1
2	div-center-script	2	2	div-center-script	2
3	div-br	6	3	div-img	
4	div-div		4	div-div	
5	div-div-img		5	div-div-span	
6	div-br		6	div-br	3

표 4 패턴 트리에 대한 테스트 트리 비교 (2)

왼쪽 트리			오른쪽 트리		
ID	노드	일치 노드	ID	노드	일치 노드
1	div-center	1	1	div-center	1
2	div-center-script	2	2	div-center-script	2
3	div-br		3	div-img	
4	div-div	4	4	div-div	4
5	div-div-img		5	div-div-span	
6	div-br	6	6	div-br	6

트리의 ID 2에 존재하기 때문에 각각의 트리의 일치 노드에 값이 입력한다. 왼쪽 노드의 ID 3인 'div-br' 노드 정보는 오른쪽 트리의 ID 6에 동일한 노드가 존재한다. 그러나 왼쪽 트리의 ID 4인 'div-div' 노드 정보는 오른쪽 트리의 ID 4에 존재하지만 일치 노드에 값을 입력하지 않는다. 오른쪽 트리의 모든 노드 정보가 비교되었기 때문에, 더 이상 비교할 대상이 없다고 판단한다. 따라서 각 트리의 비교는 여기에서 중단된다.

표 2에서 왼쪽 트리의 ID 1인 'div-center' 노드 정보는 오른쪽 트리의 ID 1에 존재하기 때문에 각 트리의 일치 노드에 값이 입력한다. 왼쪽 트리의 ID 2인 'div-center-script' 노드 정보 역시 오른쪽 트리의 ID 2에 존재하기 때문에 각각의 트리의 일치 노드에 값이 입력한다. 왼쪽 노드의 ID 3인 'div-br' 노드 정보는 오른쪽 트리의 ID 6에 동일한 노드가 존재한다. 그러나 왼쪽 트리의 ID 4인 'div-div' 노드 정보는 오른쪽 트리의 ID

4에 존재하지만 일치 노드에 값을 입력하지 않는다. 오른쪽 트리의 모든 노드 정보가 비교되었기 때문에, 더 이상 비교할 대상이 없다고 판단한다. 따라서 각 트리의 비교는 여기에서 중단된다.

이에 대한 의사코드는 표 5와 같다.

표 5 유사도 측정 의사코드

```
function Similarity() {
   for ( 패턴에 대해 ) {
      for ( 테스트 대상에 대해 ) {
         if ( 패턴.노드 = 테스트.노드 ) {
            if ( 테스트.일치노드 = null ) {
               패턴.일치노드 <- 테스트 대상의 ID;
               테스트.일치노드 <- 패턴의 ID;
               continue;
            }
         }
      }
   }
}
```

비록 표 3의 비교가 이루어졌다고 할지라도, 만족할 만큼 비교가 되었다고 볼 수 없다. 표 3과 표 4를 비교해보면, 어떤 트리를 패턴으로 설정하느냐에 따라 결과값이 달라짐을 알 수 있다. 이런 이유로, 패턴과 테스트 대상을 바꿔 다시 비교한다. 즉, 표 3의 경우와 반대로 오른쪽 트리를 패턴으로, 왼쪽 트리를 테스트로 설정하고 비교한다.

표 3과 표 4를 바탕으로 두 트리의 유사도를 계산한다. 두 트리의 유사도는 전체 노드 정보 중 일치하는 노드 정보의 수의 백분율로 구한다. 예의 경우 전체 노드 정보의 수는 24개가 되고 일치하는 노드의 수는 14개가 되므로, 두 트리의 유사도는 58%가 된다.

이에 대한 의사코드는 표 6과 같다.

표 6 유사도 계산 의사코드

```
// 패턴에 대한 처리
for ( i:=0; i<Pattern.length; i++ ) {
   if ( Pattern[i].AgreeNode != null ) {
      AgreeNode++;
   }
}
// 테스트 대상에 대한 처리
for ( i:=0; i<Test.length; i++ ) {
   if ( Test[i].AgreeNode != null ) {
      AgreeNode++;
   }
}
// 블로그 내 포스트 간의 유사도
PostSimilarity := AgreeNode / ( Pattern.length +
Test.length );
```

3.2 블로그 포스트의 일일 발행수 분석

블로그에서 하루 동안 과도하게 많은 포스트를 발행하는 것은 비정상적이며, 이는 해당 블로그가 스플로그일 가능성을 높인다.

이미 핑 서버에서의 75%, 블로그 검색 엔진의 검색 결과의 20%가 스플로그라고 알려져 있으며, 이는 실제 스플로그에서 많은 수의 포스트가 발행됨을 짐작하게 한다.

블로그는 새로 포스트를 생성하거나 기존 포스트를 수정할 경우 핑 서버에 핑을 보내게 되는데, 블로그 검색 엔진은 핑 서버로 들어오는 핑을 분석하여 블로그 검색 순위에 반영한다. 블로그 검색 엔진들은 사용자들이 입력하는 검색어와의 정확도와 포스트의 최신성을 기반으로 검색 순위를 결정한다.

일반적으로 정상 블로그의 경우 하루 동안 3~4개 이하의 포스트를 발행한다. 하지만, 스플로그의 경우 한 번에 대량의 포스트를 발행하는데, 하루 동안 대략 수십~수백개의 포스트를 생성한다.

본 논문에서는 블로그 포스트의 일일 발행수를 분석하여 스플로그를 탐지하는데 적용하고자 한다.

3.2.1 블로그 포스트의 일일 발행수 분석 절차

본 논문에서 제안하는 블로그 포스트의 일일 발행수 분석은 하루 동안 10개 이상의 블로그를 발행하는 일을 대상으로 한다. 이는 비정상적으로 발행된 포스트들을 나타내며, 이렇게 발행된 포스트가 전체 포스트 중 얼마나 차지하는 지 계산한다. 즉, 하루 동안 10개 이상의 포스트를 발행한 일들에 대해 해당 일에 발행된 포스트 전체를 합한 후, 블로그 내의 전체 포스트에 대한 백분율을 구한다.

블로그의 전체 포스트 수와 하루 동안의 포스트 발행수는 블로그의 아카이브 서비스를 통해 알 수 있다.

이에 대한 의사코드는 표 7과 같다. 각 일에 대해, 그 날에 발행된 포스트이 수를 알고 있고, 해당 블로그에서 발행된 전체 포스트의 개수를 알고 있다고 가정한다.

표 7 블로그 포스트 일일 발행수 분석 의사코드

```
function GetPostingInfo() {
   TotalCnt <- 블로그에 발행된 전체 포스트의 수

   for ( 블로그 개설일부터 특정 일까지 ) {
      PostingCnt <- 하루 동안 발행된 포스트의 수

      if ( PostingCnt > 10 ) {
         SumPosting <- SumPosting + PostingCnt;
      }
   }
   PostingInfo <- SumPosting / TotalCnt;
}
```

4. 실 험

이번 장에서는 스플로그 탐지에 대해 본 논문에서 주장하는 기법을 실험하고자 한다.

4.1 데이터셋에 대한 설명

본 논문에서 주장하는 기법에 대한 실험은 2008년 2월 14일부터 2008년 5월 10일까지 웹에서 추출한 실제 데이터를 바탕으로 진행하였다.

실험에 사용된 데이터는 300개의 블로그에서 포스트의 구조 유사도를 실험하기 위해 56,753개의 포스트를, 블로그의 일일 발행수에 대한 실험을 위해 6만여개의 일일 아카이브 데이터를 수집하였다.

추출한 데이터로부터 실험에 사용하기 위한 정상 블로그와 스플로그 각각 100개씩을 추출하고, 각각의 블로그에서 30개씩의 포스트를 추출해 교차 비교 실험을 실시하였다.

4.2 블로그 포스트 구조 비교

그림 6은 블로그 포스트 구조 유사도를 나타낸 것이다. 그림 6에서 정상 블로그의 경우 유사도의 분포가 다양하며, 약 50%의 유사성이 나타나지만, 스플로그의 경우 그래프의 상단에 주로 분포하며, 약 80% 정도의 유사성을 보인다.

그림 6에 나타난 각각의 블로그들은 블로그 내의 두 포스트들 간의 구조 유사도 검사 결과의 평균적인 수치를 나타낸다. 즉, 각각의 점들의 위치는 블로그 당 30개의 포스트에 대해 총 870번의 교차 비교 실험 결과에 대한 평균치이다.

이러한 블로그 포스트 구조 유사도의 평균치는 SVM을 이용한 분류 작업에 특성치로 사용된다.

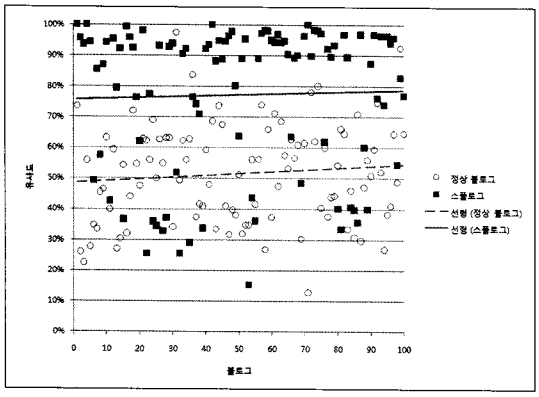

그림 6 블로그 포스트의 구조 유사성 비교

4.3 블로그 포스트의 일일 발행수 비교

그림 7은 블로그 포스트의 일일 발행수를 분석한 것이다. 일반적으로 정상 블로그의 경우 하루 동안 3~4개

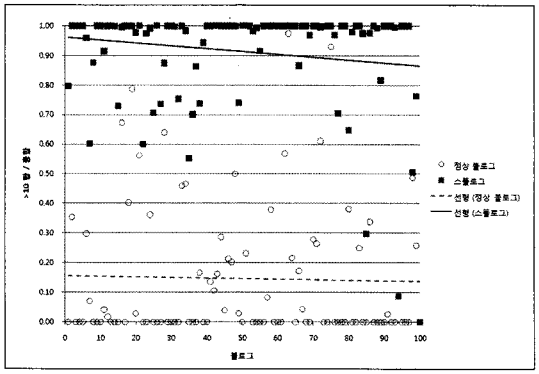

그림 7 블로그 포스트의 일일 발행수 분석

이하의 포스트가 생성된다. 하지만, 스플로그의 경우 한 번에 대량의 포스트를 생성하는데, 하루 동안 대략 수십~수백개의 포스트를 생성한다.

그림 7에서 정상 블로그의 경우 대부분이 0의 값을 가지며, 추세선 역시 0.15 정도에 위치한다. 그에 반해 스플로그는 대부분이 1.0에 위치하며, 추세선 역시 그래프의 상단에 위치한다.

그림 7은 스플로그의 경우 포스트의 대부분이 비정상적으로 발행된 것을 나타내며, 그림 7을 통해 블로그 포스트의 일일 발행수가 스플로그를 탐지하는데 유용한 척도가 될 수 있음을 알 수 있다.

4.4 SVM을 이용한 스플로그 탐지

본 실험은 앞서 설명한 블로그의 두 가지 특성을 이용해서 스플로그를 탐지한다. 본 실험의 스플로그 탐지는 SVM을 통해 데이터셋을 정상 블로그와 스플로그로 분류하는 방법으로 이루어졌다. 실험은 다항 커널함수(Polynomial Kernel Function)를 적용한 SVMlight를 사용하여, 5-묶음 교차 검증법(5-Fold Cross-Validation)을 통해 실시되었다[6,9].

5-묶음 교차 검증법은 K-묶음 교차 검증법 중의 하나이다. K-묶음 교차 검증법은 원본 데이터를 K개의 부분들로 나누는 것으로 시작한다. 나뉜 K개의 부분들 중 K-1개 이용해 모델을 학습시키고, 나머지 1개를 이용해 모델을 테스트한다. 이러한 일련의 과정을 모든 K개의 부분들에 대해 적용시킨다. 모든 테스트가 끝난 후, 단일 추정치는 각 결과 값에 대한 평균으로 구할 수 있다.

표 8은 100개와 200개의 데이터셋을 바탕으로 5-묶음 교차 검증법을 실시한 결과이다. 또한, 표 9는 표 8에 대한 최종 실험 결과를 나타낸다.

표 8에서 Accuracy는 정확도를 나타낸다. 이는 스플로그를 스플로그로, 정상 블로그를 정상 블로그로 얼마나 잘 분류하였는지의 척도가 된다. Precision은 스플로

표 8 제안기법의 5-묶음 교차 검증법 결과

	200개의 데이터셋			100개의 데이터셋		
	Accuracy Correct	Precision Incorrect	Recall Total	Accuracy Correct	Precision Incorrect	Recall Total
1	0.9500 38	1.0000 2	0.9000 40	0.9500 19	0.9091 1	1.0000 20
2	0.9000 36	0.8636 4	0.9500 40	0.9000 18	1.0000 2	0.8000 20
3	0.9750 39	1.0000 1	0.9500 40	0.9000 18	0.9000 2	0.9000 20
4	0.9250 37	1.0000 3	0.8500 40	0.9500 19	0.9091 1	1.0000 20
5	0.9000 36	0.8333 4	1.0000 40	1.0000 20	1.0000 0	1.0000 20

표 9 제안 기법 실험 결과 분포

		Prediction			
		Positive (Splog)		Negative (Authentic Blog)	
Actual	True (Splog)	20	20	0	0
		17	16	3	4
		20	93	0	7
	False (Authentic Blog)	2	3	18	17
		1	0	19	20
		1	7	19	93

그와 정상 블로그로 판단한 것 중 실제 스플로그와 정상 블로그인 것의 비율을 나타내고, Recall은 전체 스플로그와 정상 블로그 중 스플로그와 정상 블로그로 판단한 비율을 나타낸다.

본 논문의 제안 기법은 표 8에서 알 수 있듯이 상당히 만족스러운 결과를 나타내고 있다. 데이터의 수를 달리한 실험에서도 90% 이상의 성능을 보이는데, 이는 본 논문의 제안 기법이 데이터의 수와 무관하게 스플로그 탐지에 효과적임을 나타낸다.

표 9는 제안 기법에 대한 양성 오류(False Positive)와 음성 오류(False Negative)를 나타낸 것이다. 양성 오류는 정상적인 것을 비정상적인 것으로 판단하는 오류이고, 음성 오류는 비정상적인 것을 정상적인 것으로 판단하는 오류를 나타낸다. 표 9에서 음영 처리되지 않은 각 항목은 5-묶음 교차 검증법의 각 회에 해당하며, 음영 처리된 부분은 전체 데이터에 해당한다.

오판한 정상 블로그와 스플로그의 비율이 비슷하지 못한 것은 기계 학습에 사용된 데이터셋 중 올바른 결정 경계선을 찾는데 방해하는 요소들이 있는 것으로 보여진다. 많지 않은 데이터셋은 이러한 요소들에 민감하게 반응하기 때문에 보다 많은 데이터셋을 대상으로한 반복 실험이 요구된다.

정상 블로그를 오인한 경우는 대체로 그림의 사용 빈도가 문자의 사용 빈도보다 높았으며, 몇 개의 그림만으로 구성된 다수의 포스트를 포함하고 있었다. 스플로그를 오인한 경우는 대체로 스크래퍼 사이트의 모습을 보이고 있었다.

표 10 제안 기법 실험 결과 민감도와 특이도

횟수	1	2	3	4	5	전체
민감도	0.91	0.94	0.95	0.87	1.0	0.93
특이도	1.0	0.86	1.0	1.0	0.83	0.93

이를 바탕으로 표 10에는 본 논문의 제안 기법에 대한 민감도(Sensitivity)와 특이도(Specificity)를 계산한 결과이다. 여기서 민감도는 스플로그로 판단한 것 중 실제 스플로그의 비율이며, 특이도는 정상 블로그로 판단한 것 중 실제 정상 블로그의 비율을 나타낸다.

이를 바탕으로 그림 9에서는 ROC-curve(Receiver Operation Characteristic curve)를 구하였다.

ROC curve는 모델의 양성 오류와 음성 오류의 예측율을 시각적으로 표현한 것이다. Y축은 모델의 민감도를 나타내고, X축은 모델의 특이성을 나타내는데, (1-특이도)의 값으로 계산된다. 일반적으로 민감도가 높을수록, 특이도가 낮을수록 좋은 모델이라 할 수 있다.

4.5 Bag-of-Words 기법과의 비교

가장 일반적으로 사용되는 스플로그 탐지 기법인 Bag-of-Words 기법을 사용하여 본 장에서 주장하는 기법과 비교 실험을 실시하였다. 정상 블로그와 스플로그의 판단에 사용된 단어들은 정상 블로그와 스플로그 두 영역에서 사용 빈도가 높은 단어들을 추출한 것이다. 또한 실험은 본 논문에서 제안하는 기법과 동일한 5-묶음 교차 검증법으로 다항 커널함수를 적용한 SVMlight를 통해 분류하였다. 이에 대하여 표 12의 결과를 얻었다.

표 11과 같이 Bag-of-Words 기법의 성능이 떨어지는 이유는 스플로그의 한 형태인 스크래퍼 사이트 때문이다. 기존의 스플로그와는 다르게 스크래퍼 사이트의

표 11 Bag-of-Words 기법에 대한 5-묶음 교차 검증법 결과

	200개의 데이터셋			100개의 데이터셋		
	Accuracy Correct	Precision Incorrect	Recall Total	Accuracy Correct	Precision Incorrect	Recall Total
1	0.6500 26	0.8000 14	0.4000 40	0.7500 15	0.8571 5	0.6000 20
2	0.7500 30	0.8571 10	0.6000 40	0.6500 13	1.0000 7	0.3000 20
3	0.7250 29	1.0000 11	0.4500 40	0.8000 18	0.8750 2	0.7000 20
4	0.6000 24	0.8333 16	0.2500 40	0.6500 13	0.8000 7	0.4000 20
5	0.7000 28	0.8333 12	0.5000 40	0.7500 15	1.0000 5	0.5000 20

표 12 Bag-of-Words 기법 실험 결과 분포

		Prediction			
		Positive (Splog)		Negative (Authentic Blog)	
Actual	True (Splog)	18	19	2	1
		18	18	2	2
		20	93	0	7
	False (Authentic Blog)	12	15	8	5
		8	10	12	10
		11	56	9	44

표 13 Bag-of-Words 기법 실험 결과 민감도와 특이도

횟수	1	2	3	4	5	전체
민감도	0.6	0.69	0.65	0.56	0.64	0.62
특이도	0.8	0.86	1.0	0.83	0.83	0.86

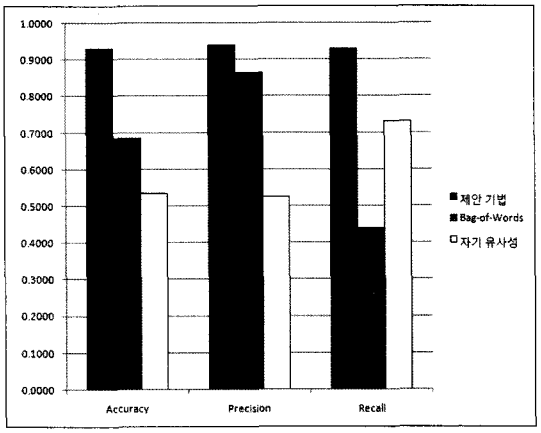

그림 8 제안기법과 비교 실험

포스트는 스팸성 키워드를 거의 담고 있지 않다. 대신 신문기사나 인기 블로그의 포스트와 같은 일상적인 내용을 담고 있다. 이렇기 때문에, Bag-of-Words 기법만으로는 스플로그를 제대로 탐지해 내지 못한다.

표 12는 Bag-of-Words 기법에 대한 양성 오류와 음성 오류를 나타낸 것이다. 또한 표 13은 Bag-of-Words 기법에 대한 민감도와 특이도를 계산한 것이다. 이 값은 그림 9에서 그래프로 표현하였다.

Bag-of-words의 경우에 Precision 값보다 Recall 값이 상대적으로 낮게 나타나는데, 이는 Bag-of-words 기법이 정상 블로그와 스플로그로 판단한 것 중 맞게 판단한 경우는 많으나, 전체 데이터에 대해서는 잘 못 판단하고 있음을 나타낸다.

그림 9는 본 논문의 제안 기법과 Bag-of-Words에 대한 ROC curve를 나타낸 것이다. 그림 9에서 제안 기법에 대한 그래프 아래의 영역은 Bag-of-Words의 아래 영역보다 상당히 큼을 알 수 있다. ROC curve에서 그래프의 아래 영역의 넓이가 1에 가까울수록 정확도가 높은 것을 나타낸다.

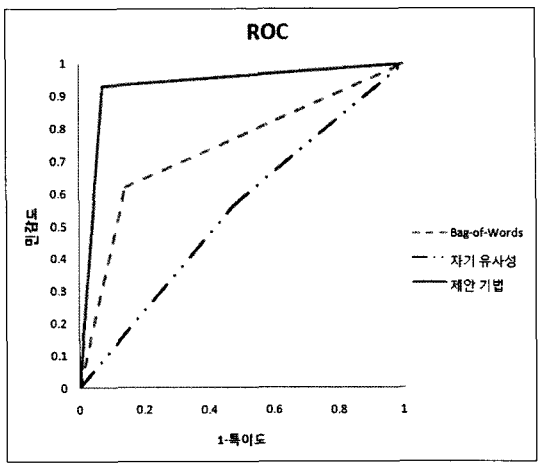

그림 9 제안기법과 비교 실험 - ROC

이는 본 논문의 제안 기법이 Bag-of-Words 보다 나은 성능을 보임을 증명한다.

표 14 시간적 자기 유사성 분석에 대한 5-묶음 교차 검증법 결과

	200개의 데이터셋			100개의 데이터셋		
	Accuracy Correct	Precision Incorrect	Recall Total	Accuracy Correct	Precision Incorrect	Recall Total
1	0.5250 21	0.5172 19	0.7500 40	0.5500 11	0.5000 9	0.7778 20
2	0.6000 24	0.5714 16	0.8000 40	0.6500 13	0.6000 7	0.9000 20
3	0.5750 23	0.5556 17	0.7500 40	0.5000 10	0.5000 10	0.7000 20
4	0.4250 17	0.4400 23	0.5500 40	0.6500 13	0.714 7	0.5000 20
5	0.5500 22	0.5333 18	0.8000 40	0.5500 11	0.5333 9	0.8000 20

4.6 시간적 자기-유사성 분석과의 비교

시간적 자기-유사성 분석과의 비교 실험을 실시하였다. 이 비교 실험은 하루 동안 포스트의 발행 시간을 비교한 것으로, 이 기법은 자동 생성된 스플로그의 포스트를 탐지하는데 효과적이라고 알려져 있다.

또한 실험은 본 논문에서 제안하는 기법과 동일한 5-묶음 교차 검증법으로 선형 커널함수를 적용한 SVM-light를 통해 분류하였다. 이에 대하여 표 14의 결과를 얻었다.

표 14와 같이 시간적 자기-유사성 분석의 성능이 떨어지는 이유는 스플로그의 목적이 과거와 달라졌기 때문이다. 과거의 스플로그는 주로 제휴 사이트로 많은 사람들의 끌여들이는 데에 목적이 있었다. 하지만, 현재의 스플로그는 애드센스로 대표되는 PPC(Pay Per Click, 클릭당지불), PPI(Pay Per Impression, 노출당지불) 광고로부터 더욱 많은 수익을 얻는 데 목적이 있다. 이에 스플로그의 방식도 기존의 프로그램에 의한 자동 생성 방식에서 일반인의 관심이 높은 기사를 다수 게시하는 방식으로 바뀌었다.

표 15는 시간적 자기 유사성 분석에 대한 양성 오류와 음성 오류를 나타낸 것이다. 또한 표 16은 시간적 자기 유사성 분석에 대한 민감도와 특이도를 계산한 것이다. 이 값은 제안기법, Bag-of-words 기법과 함께 그림 9에서 그래프로 표현하였다.

그림 8은 본 논문의 제안 기법과 Bag-of-Words, 그리고 시간적 자기 유사성 분석에 대한 Accuray, Precision, Recall을 그래프화한 것이다.

5. 결 론

과거 인터넷은 몇몇 사람들의 전유물로 보였다. 하지만, 시간이 흘러 인터넷이 널리 퍼지게 되었고, 매우 많은 정보를 인터넷에서 구하는 세상이 되었다. 웹에 대한 사람들의 관심이 커질수록, 웹을 기반으로한 다양한 서

표 15 시간적 자기 유사성 분석 실험 결과 분포

		Prediction			
		Positive (Splog)		Negative (Authentic Blog)	
Actual	True (Splog)	6	6	14	14
		8	6	12	14
		8	34	12	66
	False (Authentic Blog)	5	9	15	11
		4	4	16	16
		5	27	15	73

표 16 시간적 자기 유사성 분석 결과 민감도와 특이도

횟수	1	2	3	4	5	전체
민감도	0.55	0.67	0.2	0.4	0.6	0.56
특이도	0.52	0.57	0.56	0.44	0.53	0.53

비스들이 생겨나고 있으며, 블로그 또한 이러한 부류 중의 하나이다.

블로그는 새 글 작성의 용이함과 발행의 편리함으로 인해 많은 사람들의 관심의 대상이 되었으며, 웹을 통한 서비스 중 근래에 가장 성공한 사례이다.

이렇게 블로그에 대한 사람들의 관심이 커질수록, 블로그를 악의적인 목적으로 사용하려는 사람들이 늘어나고 있다. 이들은 웹 스팸의 한 부류인 스플로그를 통해, 그들의 목적을 달성하고자 한다. 스패머들에 의한 불법적인 활동은 정상적인 블로그 사용자의 활동을 방해해 블로그 발전에 저해 요인이 된다.

스플로그의 문제점이 대두된 뒤, 그에 따른 피해가 점점 커지고 있지만, 이를 막고자 하는 연구는 아직 미흡하기만하다. 이에 본 논문은 스플로그 탐지에 대한 연구를 통해 만족스러운 웹 활동을 하는데 도움을 주고자 한다.

본 논문에서는 블로그의 두 가지 특성을 이용하여 스플로그를 탐지하고자 하였다. 본 논문에서 제안한 기법은 블로그 내 포스트들 간의 구조 유사도와 블로그의

일일 포스트 발행수를 분석한 것이다.

본 논문에서 사용한 블로그 내 포스트들 간의 구조 유사도의 척도는 포스트의 본문에 대한 유사도이다. 본 논문에서는 포스트들 간의 구조 유사도를 측정하기 위해, 포스트 중 본문에 대해 트리 구조를 구성한 뒤 각각의 트리 구조에 대한 유사도를 측정하는 방법을 제시하였다.

또한, 본 논문에서는 스플로그 탐지를 위한 블로그의 일일 포스트 발행수를 측정하는 방법을 제시하였다.

이상의 제시한 두 특성을 바탕으로 SVM을 통해 실제 분류 작업을 수행 결과 스플로그 탐지에 있어 90% 이상의 높은 정확률을 보였으며, 이는 만족할만한 수준이다.

참 고 문 헌

[1] Dennis Fetterly, Mark Manasse, Marc Najork, "Spam, Damn Spam, and Statistics," Seventh International Workshop on the Web and Databases (WebDB 2004), 2004.

[2] Wikipedia, "blog," Online at http://en.wikipedia.org/wiki/Blog

[3] Wikipedia, "Spam Blog," Online at http://en.wikipedia.org/wiki/Spam_blog

[4] Wikipedia, "Spam in Blogs," Online at http://en.wikipedia.org/wiki/Spam_in_blogs

[5] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, Anupam Joshi, "Detecting Spam Blogs: A Machine Learning Approach," Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006), 2006.

[6] Thorsten Joachims, "SVMlight," http://svmlight.joachims.org/, 2004.

[7] Yu-Ru Lin, Hari Sundaram, Yun Chi, Junichi Tatemura, Belle Tseng, "Splog Detection Using Self-similarity Analysis on Blog Temporal Dynamics," AIRWeb 2007, 2007.

[8] Pranam Kolari, Tim Finin, Akshay Java, Anupam Joshi, "Towards Spam Detection at Ping Servers," ICWSM 2007, 2007.

[9] Wikipedia, "K-fold cross-validation," Online at http://en.wikipedia.org/wiki/Cross_validation#K-fold_cross-validation

[10] Wikipedia, "Spamdexing," Online at http://en.wikipedia.org/wiki/Spamdexing

[11] Zoltan Gyongyi, Hector Garcia-Molina, "Web Spam Taxonomy," 30th International Conference on Very Large Data Bases (VLDB 2004), 2004.

[12] Pranam Kolari and Akshay Java and Tim Finin, "Characterizing the Splogosphere," In WWW 2006, 3rd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 2006.

백 지 현

2004년 2월 경상대학교 컴퓨터공학과 졸업(학사). 2009년 2월 중앙대학교 컴퓨터공학과 졸업(석사). 관심분야는 웹 스팸, 웹 보안, SNS, 집단지성, 검색 엔진

조 정 식

2003년 2월 강남대학교 전자계산학과 졸업(학사). 2005년 2월 중앙대학교 컴퓨터공학과 졸업(석사). 2005년 3월~현재 중앙대학교 컴퓨터공학과 박사과정. 관심분야는 암호응용 및 정보보호, RFID 보안

김 성 권

1981년 2월 서울대학교 계산통계학과 졸업(학사). 1983년 2월 한국과학기술원 전산학과 졸업(석사). 1983년~1985년 목포대학교 자연과학대학 전산통계학과 전임강사. 1990년 8월 미국 University of Washington Computer Science & Engineering(박사). 1991년 3월~1996년 2월 경성대학교 이과대학 전산통계학과 조교수. 1996년 3월~현재 중앙대학교 컴퓨터공학부 정교수. 관심분야는 생물정보학, 암호응용 및 정보보호, 계산기하학 및 응용