LSA 유사도 비교를 통한 트랙백 스팸 탐지

전혁수○ 김태환 최중민
한양대학교 컴퓨터공학과
hyeksu@gmail.com kimth@islab.hanyang.ac.kr joongmin@gmail.com

Trackback Spam Detection using Similarity Analysis by LSA

Hyeksu Jun○, Taehwan Kim, Joongmin Choi
Department of Computer Science and Engineering, Hanyang University

요 약

오늘날 인터넷 사용자들은 블로그나 뉴스 등의 매체에서 트랙백을 사용해 자신의 의견을 보다 자유롭게 나타낸다. 그러나 이러한 자유로움을 악용해 트랙백 스팸을 유발하여 네트워크의 자원을 낭비하고 방문자들에게 잘못된 정보를 전달해 해당 포스트의 신뢰를 떨어뜨린다. 트랙백 스팸은 유명한 포스트와 연계하여 자신의 포스트로 사용자들을 유도하는 특징을 가지기 때문에 일반적인 웹 스팸을 탐지하는 기술을 적용하기 어렵다. 따라서 본 논문에서는 자신이 작성한 글이 다른 사람의 글과 관련이 있다고 생각하여 다른사람의 글에 자신의 글을 링크시키는 트랙백의 특성을 이용하여 원본 페이지와 트랙백 페이지 그리고 트랙백 페이지의 아웃링크 내용상의 유사도와 동시 출현(co-occurrence) 정보를 이용하여 트랙백 스팸을 처리하고자 한다.

1. 서 론

사회 연결망 서비스 중의 하나인 블로그는 개인의 의사를 표현하는 수단뿐만 아니라 기업의 홍보에까지 널리 사용되는 인터넷 미디어이다. 이러한 블로그 문화가 널리 퍼지게 된 까닭은 블로그의 주인뿐만 아니라 블로그를 방문한 사람들에게도 의견을 올릴 수 있는 덧글이나 트랙백이라는 기능이 존재하기 때문이다.

덧글은 작성된 글을 읽고 간단한 의견이나 생각을 적을 수 있고, 트랙백은 자신이 작성한 글이 다른 사람의 글과 관련이 있다고 생각하여, 다른 사람의 글에 자신의 글을 링크 시키는 것이다. 트랙백이 걸린 원문에는 트랙백을 건 사람의 포스트 페이지 주소와 그 페이지를 요약한 내용이 실리게 된다. 덧글은 긴 글이나 HTML 태그를 허용하지 않는다. 이 때문에 텍스트로만 된 글을 올려야 하는 단점이 있는데, 트랙백을 이용함으로써 자신의 의견을 보다 자유롭게 표현을 하는 장점이 있다.

하지만 덧글과 트랙백은 누구나 올릴 수 있는 자유로운 기능이어서 스패머에 의해 악용되고 있다. 스패머는 스팸 페이지를 만듦으로써 자신의 이익을 취할 뿐만 아니라, 사용자들이 원하지 않는 페이지로 유입시켜 네트워크의 자원을 낭비하고 검색엔진의 성능을 저하 시킨다. 또한 방문자들에게 잘못된 정보를 전달하고 해당 포스트의 신뢰를 떨어뜨린다.

웹 스팸은 검색엔진의 특성을 파악하여 스패머가 원하는 페이지를 해당 검색 결과 상단에 올리는 기술이다. 이러한 기술은 크게 두 가지로 구분할 수 있다. 첫째는 문장에 나타나는 용어를 이용하는 방법이고, 둘째는 페이지의 링크를 이용하는 방법이다. 전자의 방법은 용어의 빈도수(tf)에 의해 검색 순위가 결정되는 검색 시스템에서 사용된다. 다양한 질의에 대해서 검색 순위를 높이기 위해서 사전에 나오는 모든 단어를 중복해서 스팸 페이지에 넣는 방법이다. 후자는 구글 검색엔진과 같이 인링크의 개수로 검색 순위가 정해지는 시스템에서 더미 페이지를 만들어 특정 페이지로의 링크를 많이 걸어 검색 순위를 높이는 방법이다.

트랙백 스팸은 검색 엔진 결과의 상위에 랭크시키는 것이 아니라 유명한 포스트와 연계하여 자신의 포스트로 사용자들을 유도한다. 또한 자신의 포스트로 유도된 사용자들을 또 관련이 없는 광고 페이지로 유도하기 위하여 불필요한 링크를 사용한다. 이렇게 트랙백 스팸과 웹 스팸이 가지는 방법이 다르기 때문에 이전에 스팸을 처리하기 위해 연구되었던 기술들을 트랙백 스팸에 적용할 수 없다.

본 논문에서는 트랙백 스팸이 사용하는 두 가지 방법을 고려하여 스팸 문제를 해결하려 한다. 첫 번째는 목표가 된 타겟 페이지와 트랙백 페이지의 유사도를 구하여 트랙백 스팸을 찾는 방법이다. 두 번째는 트랙백 페이지가 또 다른 페이지로 유도하기 위해 사용된 링크 정보를 이용하는 방법이다. 이렇게 제안된 방법은 문서의 유사도 검색에 많이 사용되는 VSM(Vector Space Model)과 비교하여 성능 평가를 하였다.

본 논문의 구성은 다음과 같다. 2장에서 스팸 검출에 대한 기존 연구들을 살펴보고 3장에서는 본 논문에서 사용되는 유사도 측정 방법인 VSM과 LSA에 대해서 간단히 설명한다. 이어지는 4장에서는 LSA 방법을 적용한 실험 시스템에 대한 설명을 하고 5장에서 실험에 사용되는 데이터에 대해서 설명하고 결과를 보이고, 마지막으로 6장에서 결론과 향후 연구 과제를 제시하도록 하겠다.

2. 관련연구

웹 스팸은 검색엔진의 특성을 파악하여 스패머가 원하는 페이지를 해당 검색 결과 상위에 올리는 기술이다[1]. 즉, 스팸 페이지가 질의와 관련 없이 검색엔진의 상위에 올리는 방법이다. 이러한 웹 스팸을 검출하기 위한 과거의 연구를 살펴보면 두 가지 검색 엔진의 특성에 맞게 연구되었다.

첫 번째 검색엔진은 질의 단어가 문서에 나타나는 빈도수를 이용하여 문서들을 검색 결과 상위에 올린다는 특성을 이용하여 웹 스팸이 나타나게 되었고 이를 해결하려는 연구가 진행되었다.

이러한 검색 엔진을 대상으로 스패머는 다양한 질의어에도 웹 스팸 페이지를 검색 결과 상위에 나타나게 하기 위해서 사전과 같이 많은 단어를 나타내는 페이지들을 스팸 페이지에 여러 번 나타내는 방법을 취한다. 이러한 웹 스팸을 거르기 위해 A. Ntoulas[2]는 문서 내 단어의 개수, 단어의 길이, anchor text의 수, 압축률(중복)등을 구하여 햄과 스팸을 구분하는 방법을 제안하였고 약 84% 정도의 정확도를 나타낸다. P. Kolari[3]은 Supprot Vector Machines를 이용하여 블로그 스팸을 찾으려 하였다. SVM의 Feautres로 문서의 내용, anchor text의 수, URL 주소, n-gram 단어 들을 이용하였고 약 90% 정도의 정확도를 가졌다.

두 번째는 질의 단어와는 관계없이 웹 페이지가 가지는 인링크(In-link)와 아웃링크(Out-link)의 관계를 따져서 해당 페이지의 검색 순위를 결정하는 링크 기반 검색 엔진을 대상으로 높은 검색 순위를 얻으려고 한다. 링크 기반 검색엔진은 HITS[4]와 페이지랭크(PageRank)[5] 두 알고리즘을 주로 사용한다. 이러한 링크 기반 알고리즘을 악용하기 위해 스패머들은 더미 페이지를 여러 개 만들어 그 더미 페이지와 스팸 페이지의 상호 링크를 통해 더미 페이지와 스팸 페이지의 인링크 수를 높여 리스트의 순위을 높였다. 이러한 링크 기반 스팸 문제를 해결하기 위하여 Zoltán Gyöngyi[6]는 페이지랭크 알고리즘을 기반으로 하고 있는 트러스트랭크(TrustRank)알고리즘을 제안하였다. 페이지랭크 알고리즘은 모든 페이지가 신뢰할 수 있다는 특징 때문에 사용자가 인위로 만든 더미 페이지와 스팸 페이지를 구별해 낼 수 없다. 하지만 트러스트랭크 알고리즘은 신뢰할 수 있는 페이지와 신뢰할 수 없는 페이지 두 종류로 나누어서 신뢰 할 수 있는 페이지로부터의 인링크는 가중치를 부여하고, 신뢰할 수 없는 페이지로부터의 인링크는 가중치를 부여하지 않음으로서 사용자가 만든 더미 페이지와 스팸 페이지를 일반 페이지와 구분하였다.

하지만 트랙백 스팸은 검색 결과의 순위를 높여 사용자들의 유입을 증가시키는 웹 스팸과 다르게 사용자들이 즐겨 보는 블로그 포스트나 신문 기사에 트랙백을 걸어 사용자들을 스팸페이지로 유도하기 때문에 웹 스팸을 처리하는 것과 다른 방법으로 접근해야 한다.

트랙백 스팸은 타겟 페이지와는 무관한 광고 글을 게제하고 또 다른 광고 페이지로 사용자들을 유도하기 위하여 링크를 활용한다. 그렇기 때문에 타겟 페이지와 트랙백 페이지간의 유사도뿐만 아니라 타겟 페이지와 트랙백 페이지가 가지고 있는 링크 페이지간의 유사도 또한 낮다. 본 논문에서는 본문과 트랙백 및 트랙백이 가지는 링크 정보를 분석하여 유사도를 평가하고 기존의 웹 스팸에서 사용하는 방법과 비교하여 제안된 방법의 우수성을 보이려 한다.

3. 유사도 비교 모델

이 장에서는 본 논문에서 유사도 측정에 사용되는 두 가지 방법에 대해서 간단히 설명한다. 첫 번째로 정보 검색 시스템에서 쉽게 사용되는 VSM에 대해서 알아보고, 단어의 동시출현 정보를 이용하여 유사도를 측정하는 LSA에 대해서 설명한다.

3.1 VSM(Vector Space Model)

VSM은 문서를 단어의 벡터 형태로 표현하여 문서 사이의 유사도를 수학적으로 구하는 모델이다.

$$d_i = < w_{i1}, w_{i2}, ..., w_{in} > \quad (1)$$

w_{ij}는 i번째 문서에 j번째 단어가 포함 되어 있으면 1 그렇지 않으면 0으로 나타낼 수도 있고 문서 안에서 나타나는 빈도를 정수로 줄 수도 있다. 현재 정보 검색 분야에서는 대부분 식(2)의 방법인 tf-idf로 계산된 가중치 값을 사용한다[7].

$$w_{ij} = tf_{ij} \cdot \log(\frac{N}{df_j}) \quad (2)$$

w_{ij}는 i번째 문서에 j번째 단어의 가중치를 나타내고, tf_{ij}는 문서 i에서 단어 j가 나타난 횟수, df_j는 단어 j가 나타난 문서의 개수, N은 전체 문서의 수를 나타낸다.

벡터로 나타내어진 문서 d_1과 d_2의 유사도를 구하는 간단한 방법은 두 벡터 사이의 거리를 나타내는 유클리드 거리(Euclidean distance)가 있는데 이는 단순히 벡터 간의 거리만을 비교하여 실제 문서들 사이에 공통으로 나타나는 단어가 고려되지 않을 수도 있다. 따라서 본 논문에서는 단어 사이의 거리가 아닌 벡터 공간상에서 이루는 각도의 코사인 값으로 유사도가 나타나는 코사인 유사도 방법을 사용한다.

$$\text{유사도}(d_1, d_2) = \cos\theta = \frac{d_1 \bullet d_2}{|d_1| \cdot |d_2|} \quad (3)$$

3.2 LSA(Latent Semantic Analysis)

LSA[9]는 개념적으로 동시출현(co-occurrence)정보를 이용하여 단어의 형태뿐만 아니라 의미를 이용하여 문서간의 유사도를 측정하는 방법이다. 예를 들어 '사과'라는 단어는 같은 문장에 '나무' 또는 '용서'가 같이 나올 수 있는데, 두 경우 의미가 달라진다. '나무'와 같이 나오는 경우에는 '과일의 사과'를 뜻하고, '용서'와 같이 나오는 경우에는 '잘못에 대한 용서를 빎'이라는 뜻의 '사과'를 뜻하게 된다. 이론적으로는 선형대수학의 SVD(Singular Value Decomposition) 기술을 사용한다. SVD를 이용해서 높은 차원의 단어-문서 빈도 행렬을 낮은 차원의 의미 공간으로 사상시켜 단어와 문서 간에 연관관계를 구한다.

3.2.1 SVD(Singular Value Decomposition)

LSA는 차원을 줄이는데 SVD를 사용한다. SVD는 m-by-n 단어-문서 빈도 행렬 A(m: 단어 수, n: 문서 수, m > n)를 세 가지 다른 행렬로 분해한다.

$$A = T \cdot S \cdot D^T \quad (4)$$

T 행렬은 m-by-n 행렬로써 A 행렬의 행 성분(단어)을 나타낸다. D 행렬은 n-by-n 행렬로써 A 행렬의 열 성분(문서)을 나타낸다. 두 행렬은 모두 직교(orthogonal)행렬이다. S 행렬은 n-by-n 대각(diagonal)행렬로써 각 대각에는 A 행렬의 고유값(eigenvalue)이 내림차순으로 정렬되어 있다. 그림1에 단어-문서 행렬에 대한 SVD를 나타내었다.

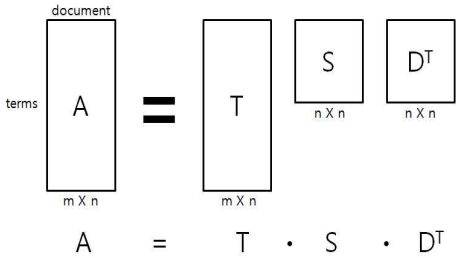

그림 1 행렬 A의 SVD

대각행렬 S에서 중요하지 않은 고유값들을 무시해 행렬 행렬 A와 유사한 행렬 A'이 나타난다. A'은 고유값의 개수 k개의 차원으로 축소 된 행렬이다.

3.2.2 LSA(Latent Semantic Analysis)

SVD를 통해서 나온 A'을 이용해서 LSA는 세 가지를 비교 할 수 있다. A'의 두 행벡터 비교를 통한 두 단어 사이의 유사도 비교, A'의 두 열벡터 비교를 통한 두 문서 사이의 유사도 비교 그리고 A' 자체로 단어와 문서와의 관계 비교를 할 수 있다.

3.2.2.1 두 단어 사이의 유사도 비교

행렬 A'의 행은 단어에 대한 특성을 나타내는 벡터 값이므로 두 행벡터를 내적 하여 두 단어의 유사도를 계산한다. 이를 위해 행렬 A'와 행렬 A'^T를 곱하면, $A' \cdot A'^T = (TS'D^T) \cdot (DS'^T T^T)$가 되고, $D^T \cdot D$는 D가 직교행렬이기 때문에 I가 된다. 따라서 두 단어의 유사도는 식(5)의 셀(i, j)에 나타난다.

$$A' \cdot A'^T = TS'^2 T^T \quad (5)$$

3.2.2.2 두 문서 사이의 유사도 비교

두 단어 사이의 유사도 비교를 수행하는 것과 비슷하다. 단지 행렬 A'의 행벡터를 내적 하는 것이 아니라 열벡터를 내적 하는 것이 다르다. 따라서 행렬 A'^T와 행렬 A'을 곱하면 $A'^T \cdot A' = (DS'^T T^T) \cdot (TS'D^T)$가 되고, $TT \cdot T$는 T가 직교행렬이기 때문에 I가 된다. 따라서 두 문서의 유사도는 식(6)의 셀(i, j)에 나타난다.

$$A'^T \cdot A' = DS'^2 D^T \quad (6)$$

3.2.2.3 단어와 문서사이의 관계 비교

열벡터나 행벡터를 내적 하는 것과는 다르게 행렬 A'의 셀(i, j)의 값이 단어 i와 문서 j의 관계를 나타낸다.

$$A' = T \cdot S' \cdot D^T \quad (7)$$

본 논문에서는 문서간의 유사도를 구하기 위해 두 번째 방법인 식(6)을 사용한다.

4. 트랙백 스팸 탐지

스팸 트랙백은 직접 만드는 데에 많은 노력이 들기 때문에 자동으로 생성한다. 그렇기 때문에 의미가 없거나 글의 순서가 없고 조리가 없다. 이러한 페이지들은 타겟 페이지와 유사도가 많이 떨어지게 된다. 또 광고를 목적으로 생성하기 때문에 스팸 트랙백의 아웃링크들 역시 타겟 페이지와의 유사도가 낮다. 따라서 타겟 페이지와 트랙백 페이지, 아웃링크 사이의 유사도를 구하여 임계값 이하의 유사도를 가지는 트랙백은 스팸 트랙백으로 간주를 한다.

본 논문에서는 앞 장에서 설명한 두 모델인 VSM을 사용하여 타겟 페이지와 스팸 페이지와의 유사도를 구하는 기준치로 삼고, LSA를 사용하여 유사도를 구해 기준치인 VSM과 비교를 할 것이다.

4.1 트랙백 스팸 필터 시스템

문서의 유사도를 구하기 위해 타겟 페이지, 트랙백 페이지 그리고 트랙백 페이지의 아웃링크들을 트랙백 페이지에 대한 트리플로 구성한다. 각 트리플을 입력으로 하여 유사도를 구한다.

문서를 대표할 수 있는 주제어를 추출하기 위해 각 페이지의 HTML 태그를 제거하고 형태소분석을 통해 문서 내의 명사를 추출한다[1]. 명사는 문장을 대표하는 주어나 목적어들로 이루어져 있다. 따라서 문서의 명사는 문서를 대표할 수 있는 주제어로 적합하다고 판단하여 추출한다. 그 후 추출 된 명사를 좀 더 일반적인 형태로 나타내기 위해 불용어를 제거하고, 스테밍[2]을 한다. 추출된 주제어의 집합을 문서의 단어 집합으로 한다.

LSA를 적용하기 위해서 앞에서 구한 문서 집합을 가지고 단어-문서 행렬 A를 구성한다. 행렬의 각 값들은 식(2)을 사용하여 가중치를 구한 값으로 한다.

그 다음 LSA를 적용하는 방법 알고리즘1과 같다.

·단계 1. 주어진 행렬 A를 SVD를 수행하여 TSD^T로 분해한다.

·단계 2. 차원을 줄여 새로운 행렬 A'을 구한다.

2.1 $\sum_{i=1}^{k} C_i \geq \theta$을 만족하는 k를 구한다.

, where $C_i = \frac{w_i^2}{\sum_{k=1}^{n} w_k^2}$ (w_i는 S의 i번째 고유값)

2.2 $w_i = 0$을 수행해 수정된 S'을 구한다.

, where i = k+1, k+2, ..., n

2.3 새로운 행렬 $A'=TS'D^T$을 구한다.

2.4 A'의 값을 0과 1사이의 값으로 정규화 한다.

·단계 3. 문서 간 유사도를 구하기 위해 식(6)을 수행한다.

알고리즘 1 LSA 적용 방법

문서 간 유사도 행렬의 셀(i, j)이 가지는 값은 문서 i와 문서 j의 유사도를 나타낸다. 각 문서들 간의 유사도를 하나의 값으로 나타내기 위해 식(8)을 사용한다.

$$\text{유사도}(o,t) = (1-\alpha)\cdot sim(o,t) + \alpha\cdot\sum_{i=1}^{n}\frac{sim(o,l_i)}{n} \quad (8)$$

o는 타겟 페이지를 나타내고 t는 트랙백 페이지, 그리고 l_i는 트랙백의 아웃링크 페이지를 나타낸다. n은 아웃링크의 수 그리고 sim(a, b)는 문서 a와 문서 b의 LSA 문서 유사도를 나타낸다. α는 댐핑팩터이다.

유사도 측정 후 유사도에 대한 평가를 한다. 유사도가 임계값 c 이하인 페이지를 스팸 페이지라 판단하여 분류한다. 이 때 임계값 c는 실험을 통해서 결정 한다. 제안하는 시스템의 구조는 그림 2에 나타내었다.

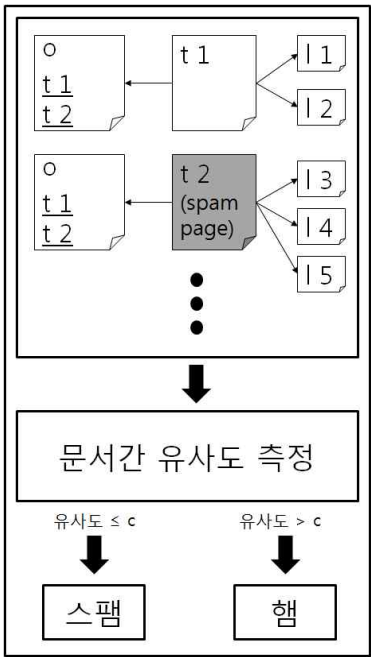

그림 2 시스템 구조

5. 실험

5.1 실험 데이터

데이터는 구글에서 임의로 검색된 블로그, 잡지 기사, 뉴스 기사 중 트랙백을 가지고 있는 338개의 타겟 페이지에서 수집을 하였다. 973개의 트랙백을 대상으로 연구실 인원 3명이 스팸과 햄에 대해서 판단하였다. 타겟 페이지와 트랙백 페이지와의 유사도에 대한 선입견을 제거하기 위해 단지 트랙백 페이지만을 보고 직관적으로 스팸과 햄에 대해서 판단하였다. 직관적으로 스팸을 구분하기 위하여 다음과 같은 판단 기준을 제시하였다. 첫째는 타겟 페이지에서 트랙백 페이지로 이동시 해당 URL의 메인 페이지로 링크된 경우는 스팸으로 판단하였다. 두 째는 링크된 페이지의 내용이 3줄 이하인 경우 스팸으로 판단하였다. 3줄 이하인 경우에는 타겟 페이지

1) http://nlp.stanford.edu/software/index.shtml
2) http://tartarus.org/~martin/PorterStemmer

내에 댓글을 이용할 수 있는데 트랙백을 사용했기 때문이다. 이는 다분히 자신의 페이지로 사용자들을 유입하려는 의도를 내포하기 때문이다. 세 째는 링크된 페이지 내에 광고와 관련된 내용이 있는 경우 스팸으로 판단하였다. 이렇게 직관적으로 판단한 결과 457개의 햄 트랙백, 244개의 스팸 트랙백, 나머지 272개의 트랙백은 영어가 아닌 언어로 되어 있거나 접속이 되지 않는 트랙백이었다. 또 457개의 햄 트랙백과 244개의 스팸 트랙백에서 14710개의 아웃링크를 추출하여 타겟 페이지와 트랙백 그리고 아웃링크들 간의 유사도를 측정하였다.

표 1 트랙백 페이지의 분포

	햄	스팸	기타	합
페이지	457	244	272	973
아웃링크	8022	6688	0	14710

5.2 평가방법

스팸 페이지와 햄 페이지의 분류 성능을 평가하기 위해 전체 햄의 문서들 중에서 얼마나 많은 햄을 찾았는지 그리고 전체 스팸의 문서들 중에서 얼마나 많은 스팸 문서를 걸러냈는지를 계산하였다. 각각은 햄 페이지에 대한 재현율(H_{recall})과 스팸 페이지에 대한 재현율(S_{recall})에 대해서 나타내는 수치이다. 이 두 값을 하나의 값으로 판단하기 위해 두 값의 조화평균(HM)을 사용하였다.

$$H_{recall} = \frac{\text{시스템이 햄이라고 구한 수}}{\text{전체 햄의 수}} \quad (9)$$

$$S_{recall} = \frac{\text{시스템이 스팸이라고 구한 수}}{\text{전체 스팸의 수}} \quad (10)$$

$$HM = \frac{2 \cdot S_{recall} \cdot H_{recall}}{S_{recall} + H_{recall}} \quad (11)$$

5.3 조화평균 측정을 통한 최적의 임계값 산출

먼저 기준치인 VSM에 대한 실험의 임계값 c를 구하기 위해 c를 0.1 단위로 0부터 1까지 조화평균값을 구하였다. 그림 3에 임계값의 변화에 따른 햄의 재현율과 스팸의 재현율 그리고 조화평균값을 나타내었다. 이 결과에서 보면 c가 0.3일 때 가장 성능이 좋은 것을 알 수 있었으며, 이 결과를 토대로 임계값을 0.3으로 정하였다.

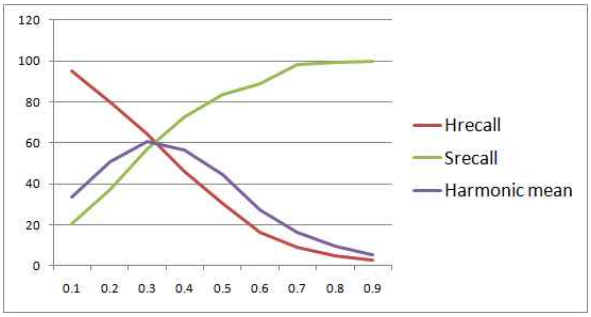

그림 3 VSM의 임계값의 변화에 따른 측정 결과

LSA에 대한 가중치 α와 임계값 c를 구하기 위해 각 α와 c를 0.1단위로 0부터 1까지 단계별로 실험을 수행하였다. 표 1에 각 α와 c에 따른 조화평균값을 나타내었다. α와 c 모두 0.4일 때 가장 성능이 좋은 것을 알 수 있었으며, 이 결과를 토대로 가중치 α와 임계값 c를 모두 0.4로 정하였다.

표 2 LSA의 α와 c값의 변화에 따른 조화평균

α \ c	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	29.3	37.8	53.8	62.4	67.8	69.3	70.4	63.8	57.6
0.1	29.3	39.8	55.3	64.4	69.8	70.6	67.7	62.5	51.9
0.2	29.8	42.7	58.6	67.8	70.1	70.4	66.0	60.0	40.2
0.3	26.8	47.1	62.0	69.7	71.2	68.9	64.1	56.1	28.2
0.4	28.0	51.3	65.2	**71.6**	70.8	67.9	61.4	50.9	22.4
0.5	29.7	57.5	67.9	70.7	70.3	66.2	58.0	44.3	15.3
0.6	35.7	62.4	70.6	70.1	68.9	63.8	53.9	37.4	11.9
0.7	45.0	68.8	70.2	69.0	67.1	60.9	50.8	33.2	8.0
0.8	57.0	68.8	70.8	68.1	65.2	58.7	47.2	30.0	6.4
0.9	65.0	68.5	68.9	67.4	62.8	56.3	44.0	28.2	4.7
1.0	66.0	69.1	68.5	66.2	61.7	53.9	40.8	24.0	2.6

5.4 실험결과

그림 4에 기준치 방법인 VSM을 사용하여 타겟 페이지와 트랙백 페이지 사이의 유사도를 측정하여 나타내었

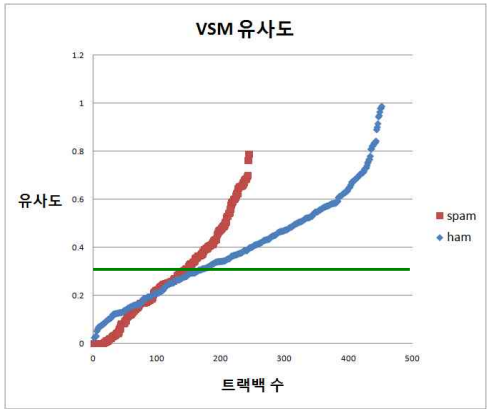

그림 4 VSM을 이용한 유사도

다. 그래프에서 파란색은 햄을 나타내고 빨간색은 스팸을 나타낸다. 그리고 녹색은 임계값인 0.3을 나타낸다. 임계값이 0.3일 때 햄은 295개를 찾고, 스팸은 139개를 걸러내었다. 그림 5에 LSA를 사용하여 타겟 페이지와 트랙백 페이지 사이의 유사도와 타겟 페이지와 아웃링크들과의 유사도를 측정한 값을 나타내었다. 이 때 가중치 α와 임계값 c의 값은 0.4로 동일하다. 이때의 결과는 햄은 379개를 찾고, 스팸은 131개를 걸러내었다.

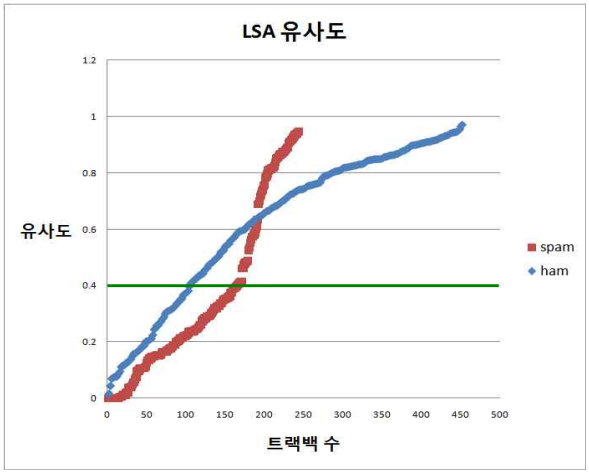

그림 5 LSA(α=0.4)를 이용한 유사도

표3에 VSM(c=0.3)과 LSA(α=c=0.4)의 결과를 비교하여 놓았다. LSA를 이용한 경우가 햄의 재현율이 약 11.6%, 스팸의 재현율이 약 10.65% 그리고 조화평균이 약 11.11% 나은 성능을 보였다.

표 3 VSM과 LSA의 결과 비교

	H_{recall}(%)	S_{recall}(%)	HM(%)
VSM	64.55	56.97	60.52
LSA	76.15	67.62	71.63

5. 결론 및 향후 연구 과제

본 논문에서는 트랙백 스팸을 탐지하기 위해서 LSA를 사용하여 타겟 페이지와 트랙백 페이지 그리고 아웃링크의 유사도 측정하는 시스템을 제안하고 구현하였다. 제안한 시스템의 성능을 VSM을 사용한 시스템을 비교 평가 하였을 때 전반적으로 스팸 재현율과 햄 재현율 그리고 조화평균의 결과가 모두 10% 이상 나은 성능을 보였다.

하지만 스팸을 햄으로 판단한 경우가 약 32.38%가 되는데 이유는 스팸 트랙백 페이지를 단순히 의미 없는 단어의 나열로 생성하는 것이 아니라 타겟 페이지의 내용을 그대로 복사하여 스팸 트랙백 페이지를 생성 할 경우 두 페이지의 유사도가 높은데 이를 고려하지 않고 단순히 유사도가 낮은 트랙백을 스팸 트랙백으로 탐지하여서이다. 향후 과제로는 이러한 유사도가 높은 스팸 트랙백 페이지도 탐지하는 방안을 고려하고 있다.

또한 햄을 스팸으로 판단할 경우 옳은 정보를 제거하여 사용자들에게 제대로 된 정보를 전달해 주지 못한다. 이런 결과가 약 109개로 전체 햄의 비율중 약 23.9%를 차지한다. 이러한 데이터를 살펴보니 트랙백 페이지 문서의 길이가 대체로 짧아 단어의 빈도수가 다른 문서보다 낮아 타겟 페이지와의 유사도 또한 낮게 나왔다. 추후 문서의 길이에 관계없이 강건(robust)한 스팸 탐지 기법에 대해서도 연구해 보고자 한다.

참고 문헌

[1] Z. Gyöngyi and H. Garcia-Molina. Web Spam Taxonomy. In 1st International Workshop on Adversarial Information Retrieval on the Web(AIRWeb 2005), page 39-47, May 2005.

[2] A. Ntoulas, M. Manasse, and D. Fetterly. Detecting Spam Web Pages through Content Analysis. In Proceedings of the 15th International Conference on the World Wide Web, May 2006.

[3] P. Kolari, T. Finin, and A. Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs, 2006.

[4] J. Kleinberg. Authoritative sources in a hyperlinked environment. In Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, page 668-677, Jan. 1998.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[6] Z. Gyöngui, H. Garcia-Molina, and J. Pedersen. Combating Web Spam with TrustRank. In Proceedings of the 30th International Conference on Very Large Databases(VLDB), 2004.

[7] Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval. McGraw-Hill. 1983.

[8] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. Journal of the American Society of Information Science, 41(6):391-407, 1990.