碩 士 學 位 論 文

본문과 덧글의 동시출현 자질을 이용한 역 카이제곱 기반 블로그 덧글 스팸 필터 시스템

A Comment Spam Filter System based on Inverse Chi-Square Using of Co-occurrence Feature between Comment and Blog Post

高麗大學校 컴퓨터情報通信大學院

미디어工學 專攻

田 喜 元

2007 年 12 月

林 海 彰 敎 授 指 導

碩 士 學 位 論 文

본문과 덧글의 동시출현 자질을 이용한 역 카이제곱 기반 블로그 덧글 스팸 필터 시스템

A Comment Spam Filter System based on Inverse Chi-Square Using of Co-occurrence Feature between Comment and Blog Post

이 論文을 工學 碩士學位 論文으로 提出함

2007 年 12 月

高麗大學校 컴퓨터情報通信大學院

미디어工學 專攻

田 喜 元

田喜元의 工學 碩士學位論文 審査를 完了함.

2007年 12月

委員長 林 海 彰 (印)

委 員 白 斗 權 (印)

委 員 陸 東 錫 (印)

요 약

최근 대표적인 1 인 미디어의 형태인 블로그(Blog)는 개인 기록의 수단뿐만 아니라 기업의 홍보에까지 널리 사용되는 인터넷 미디어이다. 그러나 누구나 글을 쓸 수 있다는 자유로움 이면에 이를 이용한 덧글 스팸이 성행하고 있다.

일반적인 스팸 필터의 경우 그 해당 덧글만을 가지고 스팸 필터링을 한다. 그러나 특성상 스팸인 덧글이 정상인 덧글보다 상대적으로 짧기 때문에 일반적인 덧글 자체만의 필터링 방법으로는 높은 정확도를 기대하기 힘든 단점이 있다.

본 논문에서는 정상인 덧글과 본문간의 내용상의 유사도가 있음을 가정해 이런 정보를 역 카이제곱 분류기에 동시출현(co-occurrence) 정보로 부여함으로써 스팸 필터의 정확도를 높이고자 했으며, 실제 그러한 정보를 추가함으로 단순한 확률기반 스팸 필터링 방법을 사용하는 것보다 스팸 필터의 전반적인 성능이 상승되었음을 실험 결과를 통해 알 수 있었다.

Abstract

Blog is the best media that can be used in individual purpose what is more can be used in corporate communication. Beside of free writing, there is abusing of blog comment spam.

In case of common spam filter, it only use comment feature. But it is hard to gain high accurate rate, because spam comment is shorter than ham comment that cause shortage of features using in spam filter algorithm.

This paper suggests a similarity assumption between main post and comment, and using spam filter algorithm added co-occurrence information feature with current term probability feature. Actually after adding this feature, we gain more accuracy than common filter that only use term probability feature.

목 차

그림 목차

표 목 차

1. 서 론

1.1 연구의 배경

블로그(Blog)란 사전적의미로 인터넷을 뜻하는 웹(Web)과 항해일지를 뜻하는 로그(Log)의 합성어로 웹로그(Weblog)를 의미하는 것으로 점차 대중화 되어 '블로그'라는 말로 줄여서 사용하고 있다. 로그의 의미에서 알 수 있듯이 블로그는 웹에 기록하는 개인의 일기를 뜻한다. 블로그가 대중화 되면서 점차 개인의 신변잡기적인 내용이 주된 주제가 되기 보다는 좀더 전문적인 분야로 주제가 다양화 되고 있는 상황이다. 예를 들어 칼럼, 기사 등의 글의 형태뿐만 아니라 음성, 동영상에 이르기까지 그 형태도 다양하다. [21]

또한 이런 블로그를 가지고 기업의 홍보를 하거나 특정 유명인이 자신의 생각과 여론을 알아보기 위한 그러한 마케팅적인 용도로까지 그 쓰임새가 다양해 지고 있다. 게다가 블로그의 무한한 잠재적인 가능성을 기반으로 많은 국내외 검색엔진 업체들이 그들만을 검색할 수 있는 검색 서비스를 제공하고 있는 실정이다.

실제 블로그 전문 검색서비스인 technorati 의 설립자 데이브(Dave Sifry)는 통계 결과 매일 1 만 2 천 개의 블로그가 생겨나고 40 만개의 글이 올라온다고 밝혔다. 자료에 의하면 미국의 1 억 2 천만 성인 인터넷 사용자중 27%인 3200 만 명이 블로그를 정기적으로 구독하고 있다고 한다.

이처럼 블로그는 전 인터넷 영역에 걸쳐 가장 영향력 있는 인터넷 미디어의 한 형태가 되어 가고 있다.

블로그라는게 자유롭게 글을 쓸 수 있다는 기능이 주된 기능이지만 이것을 이용한 스팸이 블로그 스피어가 발전함에 따라 성행하고 있다. 블로그에 보면 트랙백(trackback)과 덧글(comment) 기능이 있다는 이 기능은 블로그 운영자라 불리는 블로거(Blogger)를 위한 기능이라기 보다는 블로그 방문자를 위한 의견 제시 창구로 활용이 되고 있는데, 이 기능을 역이용해서 특정 업체로 트래픽을 유도하거나 상품에 대한 홍보를 하는 등 블로그와 전혀 상관없는 스팸들이 난무해서 블로거나 블로그 방문자로 하여금 정보에 대한 혼란을 유도하고 있다.

아래의 그림 1-1 은 Akismet(http://akismet.com/)에서 조사한 스팸 덧글과 정상 덧글간의 통계를 나타낸 표이다. 실제 시간이 지나면 지날수록 정상덧글(Legit comments)에 비해 스팸덧글의 비율이 늘어나는 것을 볼 수 있다.

이 문제는 실제 국내 여러 포털에서나 특정 검색엔진 업체에서 문제점을 자각해 여러 방법을 동원해서 사용자들이 불편없이 블로그를 운영하게끔 상당한 리소스를 투입하고 있는 실정이지만 스팸이 날로 지능화 됨에 따라 이에 대한 요구와 리소스를 더 늘어갈 전망이다.

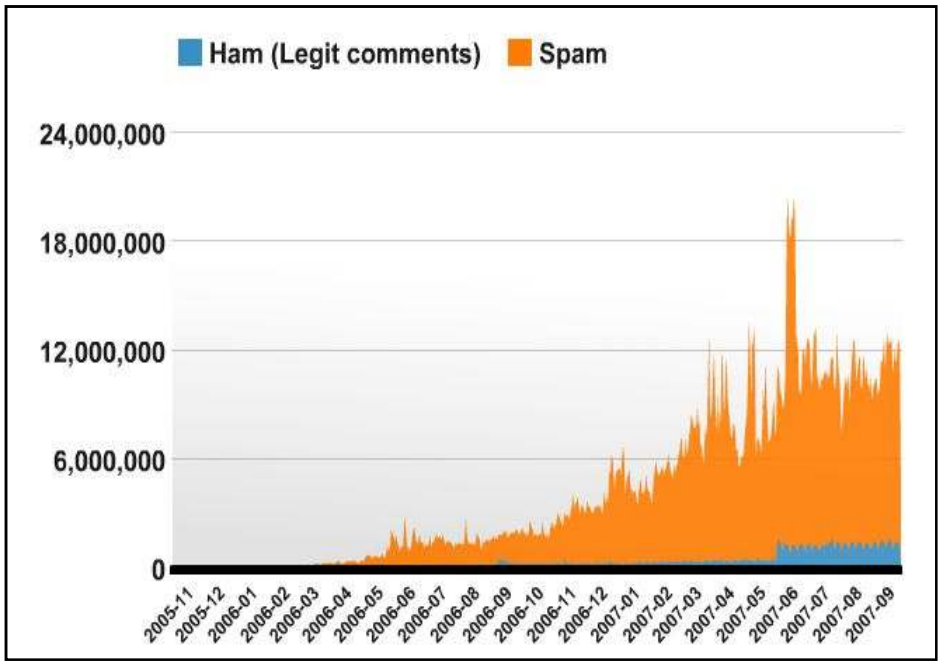

그림 1-1 스팸 덧글과 정상 덧글간의 Abuse 통계

1.2 연구의 목적

현재 덧글 스팸을 효율적으로 차단하는 여러 방법이 존재하며[4] 한가지 방법으로는 한계가 있어 여러 방법을 복합적으로 적용해 스팸 필터링을 하고 있다. 그들 방법 중에서 스팸에 존재하는 단어와 정상인 문서에 존재하는 단어간의 출현 빈도의 차이가 있다는 가정하에 제안된 베이지언 확률 기반 필터링 방법이 블로그 덧글 스팸처리에서도 쓰이고 있다[16]. 하지만 이런 확률 기반의 스팸 처리 방법을 쓰기에는 스팸인 덧글이 정상 덧글에 비해 짧아 단어 정보를 모으기에는 힘들다는 단점이 있고, 또한 실제 판정을 할 때 짧은 덧글만으로 필터링을 하면 오류율이 높아져 정확도가 떨어지는 단점이 있다. 이는 e-mail 스팸 기반으로 스팸 처리

방법론이 발전된 베이지언 확률 기반 필터링 방법이 e-mail 보다 상대적으로 짧은 덧글 스팸을 처리하는 데는 한계가 있음을 의미한다. 따라서 덧글 스팸 처리에서 이런 부족한 판정 자질에 대해 보강할 다른 자질에 대한 발굴이 필요했다.

본문과 덧글은 매우 관련성이 높은 글이다. 대부분 본문에 나온 내용을 기반으로 덧글 사용자들이 의견을 제시하기 때문에 덧글과 본문은 상당한 유사도의 가능성이 있다. 하지만 이런 연관성이 없이 무분별하게 채택된 단어로 이루어진 스팸 덧글은 이런 유사성을 보이기 힘들다. 이런 덧글과 본문의 유사도에 대한 가정을 기반으로 확률 기반의 필터링 방법을 보강하고자 한다.

본문과 덧글간 유사도에 대한 가정을 주제어에 대한 동시출현(co-occurrence) 확률 자질로 구현을 하고 이를 역 카이제곱 분류기에 적용을 하여 실험을 했다. 역 카이제곱을 이용한 이유는 베이지언 방법이 독립성 가정, 희소 단어 처리, 비대칭적 관계 등에 취약하다고 알려져 있고, 필터에서 중요한 성능 평가 요소로 판단되는 False Positive 부분에서 베이지언에 비해 나은 결과를 보여주는 것으로 알려져 있기 때문이다[17]. 이러한 장점은 본 논문의 실험 중간 결과에서도 재 증명 되고 있다.

1.3 연구의 구성

본 연구의 구성은 다음과 같다.

제 1 장에서는 본 연구를 수행하게 된 배경과 목적을 살펴보고 본 연구의 구성을 제시한다.

제 2 장에서는 관련 연구로서 기존의 기술과 관련 연구에 대한 조사를 실시하고, 역 카이제곱을 이용한 스팸필터에 대한 소개를 하겠다.

제 3 장에서는 역 카이제곱 방법과 동시출현 자질을 적용한 실험 시스템에 대한 설명과 학습 데이터와 테스트 데이터에 대한 소개를 한다.

제 4 장에서는 실험 결과를 바탕으로 분석을 실시한다.

마지막으로 제 5 장에서는 결론을 도출하고 향후 연구 과제에 대한 고민을 해보도록 하겠다.

2. 관련 연구

관련 연구의 소개는 본인이 교사(supervised) 기반의 스팸 필터링 방법을 사용했기 때문에 기존의 교사기반의 문서분류 방법들에 대한 소개와 더불어 블로그 덧글 스팸을 처리하기 위한 휴리스틱한 방법들에 대한 소개를 나눠서 하도록 하겠다.

2.1 문서 분류

문서 분류란 많은 양의 서로 다른 문서들을 미리 정의된 여러 가지 카테고리 중 하나에 속하도록 분류하는 것을 말한다. 이런 의미에서 스팸 분류도 문서분류에서 하나의 중요한 카테고리에 속한다고 말할 수 있겠다.

문서 분류의 과정은 기 분류된 다양한 카테고리의 문서셋을 가지고 학습기를 통해 학습을 시킨 다음에 그 학습 정보를 기반으로 미 분류된 입력 문서들에 대한 카테고리를 판단하게 된다.

이러한 문서 분류의 과정에서 필수적인 과정을 꼽으라 하면 바로 문서를 수치로 표현하는 것이 되겠다. 일반적으로 문서의 수치화는 문서에 포함된 단어를 기반으로 하게 된다. 그 단어들을 기반으로 벡터를 만들고 그 문서 벡터를 기반으로 문서를 학습 및 분류하게 된다.

학습 기반 문서 분류 방법은 여러 가지 방법론이 존재한다. Naive Bayesian, Support Vector Machine(SVM), Decision Tree, Boosting Tree, k-nearest neighbor(K-NN) 등 많은 학습 방법들이 채택되고 현재도 연구중에 있다. 이

중에서 나이브 베이지언(Naive Bayesian)과 SVM, 카이제곱(chi-square)방법에 대해서 설명하도록 하겠다.

2.1.1 나이브 베이지언(Naive Bayesian)

나이브 베이지언(Naive Bayesian) 모델은 문서 분류에서 가장 보편적으로 사용되는 방법이다. 베이즈 정리를 이용하여 개발된 이 알고리즘은 텍스트 분류에서 신경망이나 결정트리 학습에 비교되는 성능을 보여주며, 자료량이 많아질수록 정확도가 높다.[8]
기본적인 아이디어는 주어진 문서를 입력 받은 뒤 그것이 각 카테고리에 할당될 확률을 계산하는 방법으로 분류한다. 문서가 특정 카테고리에 속하는 확률을 계산하기 위하여 식 (1)과 (2)의 베이즈 정리를 이용한다.

$$P(c \mid x) = \frac{P(c)P(x \mid c)}{P(x)} \qquad (1)$$

$$P(x) = \sum_{c \in C} P(c)P(x \mid c) \qquad (2)$$

여기에서 x는 임의의 문서를 의미하고 c는 임의의 카테고리를 의미한다. 식(1)의 P(x)는 전확률 공식(total probability formula)에 의해 식 (2)와 같이 정의된다. 그런데 P(x)는 모든 카테고리에 대하여 같은 값을 가지므로 확률을 계산하는데 고려하지 않아도 된다. 따라서 식 (1)의 분모에 위치한 P(c)와

P(x|c)만 추정하면 문서 x가 카테고리 c에 할당될 확률을 계산할 수 있다. P(c)는 모든 카테고리 중 카테고리 c가 뽑힐 확률이다. 따라서 이는 모든 학습 문서들의 수인 $|X_L|$와 카테고리 c에 속하는 학습 문서들의 수인 $|X_{L,C}|$의 비율로 추정할 수 있다. 따라서 다음과 같은 식 (3)이 성립한다.

$$P(c) = \frac{|X_{L,c}|}{X_L} \qquad (3)$$

P(c|x)를 계산하기 위해서는 P(x|c)를 계산해야 한다. 문서 x는 단어들의 벡터인 $<w_1, w_2, \ldots, w_{|x|}>$ 로 나타낼 수 있다. 따라서 P(x|c)는 다시 $P(<w_1, w_2, \ldots, w_{|x|}>/c)$로 나타낼 수 있다. 나이브 베이지언 알고리즘 은 $P(<w_1, w_2, \ldots, w_{|x|}>/c)$의 계산을 좀 더 쉽게 하기 위해, '베이즈 독립성' 을 적용한다. 베이즈 독립성 가정은 문서 내에 존재하는 모든 단어들인 $w_1, w_2, \ldots, w_{|x|}$가 서로 독립이고, 문서 내의 단어 위치와 그 단어의 출현확률 사이에도 독립성이 존재한다는 것이다. 단어의 결합을 사용하지 않기 때문에 지수 복잡도의 다른 방법들보다 나이브 베이지언 분류를 더 효과적이게 한다. 이 가정에 따르면 P(x|c)는 다음과 같은 식으로 표현된다.

$$P(x \mid c) = \prod_{k=1}^{|x|} P(w_k \mid c) \qquad (4)$$

n_c를 카테고리 c에 출현하는 모든 단어들의 빈도수의 합이라고 하고, $n_{c,w}$를 카테고리 c에 출현하는 단어 w의 빈도수라 할 때, P(w|c)의 추정치를

$\frac{n_{c,w}}{n_c}$ 라 한다. 그러나 이 추정치를 식 (4)에 그대로 적용하면, 이 전체 식의 값을 0으로 만들 확률이 높다. 왜냐하면, 분류하려는 문서 내에 존재하는 단어가 확률을 계산하려는 카테고리 내에 존재하지 않을 수도 있기 때문이다. 이러한 문제를 해결하기 위해서 일반적으로 식 (5)와 같이 m-estimate 개념을 응용한 기법을 이용한다. 여기에서 |vocabulary|는 모든 학습문서 내에 포함되어 있는 서로 다른 단어의 개수이다.

$$P(w \mid c) = \frac{n_{c,w} + 1}{n_c + |vocabulary|} \qquad (5)$$

LEARN_NAIVE_BAYES_TEXT(Examples,V)

1. Example에 나타난 모든 단어와 토큰을 모은다.
   - Vocabulary : Example에 나타난 모든 상이한 단어 및 토큰들
2. 필요한 확률 $P(v_j)$와 $P(w_k|v_j)$를 계산
   - For 각 타겟값 V에서 v_j do
     - $docs_j$: 타겟값이 v_j인 Example의 부분집합
     - $P(v_j)$: $\frac{docs_j}{Examples}$
     - $Text_j$: $docs_j$의 모든 요소를 나열하여 만든 하나의 문서
     - n : $Text_j$에 있는 전체 단어의 수(중복된 단어는 중복된 횟수만큼 계산)
     - For Vocabulary에 있는 w_k do

       n_k : $Text_j$에 나타난 w_k의 횟수

       $P(w_k|v_j)$: $\frac{n_k + 1}{n + Vocabulary}$

CLASSIFY_NAIVE_BAYES_TEXT(Doc)

- position : Vocabulary에서 발견된 토큰들의 Doc내의 모든 단어들의 위치
- 다음 식에서 계산된 V_{NB}

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod P(a_i | v_j)$$

2.1.2 SVM

Support Vector Machine(SVM)은 Vladimir Vanpnik과 그의 AT&T Bell 연구소의 팀이 개발한 식별 방법으로 최근 몇 년 동안 이론적인 발전뿐만 아니라, 실제 구현되어 데이터 마이닝 분야는 물론 얼굴인식과 같은 패턴인식 응용 분야에도 널리 사용되고 있다.[23]

SVM은 다항식(polynomial), 방사 기저함수(Radial Basis Function), 그리고 다층 퍼셉트론 분류기(Multi-Layer Perceptron classifier)의 대안적인 학습 방법으로 패턴을 고차원 특징 공간으로 사상시킬 수 있다는 점과 대역적으로 최적의 식별이 가능한 특징을 가진다. 여기서 네트워크의 가중치(weight)는 비볼록(non-convex), 제약 조건이 없는 최소화 문제를 해결함으로써 구해지는 일반적인 신경망과는 달리 선형 부등 조건을 가진 QP(Quadratic Programming)문제를 해결함으로써 얻어진다. 또한 신경망을 포함하여 통계적 패턴인식 방법 등 전통적인 대부분의 패턴인식 기법들이 학습 데이터의 수행도를 최적화 하기 위한 경험적인 위험 최소화(Empirical Risk Minimization) 방법에 기초하는데 반해, SVM은 고정되어 있지만 알려지지 않은 확률 분포를 갖는 데이터에 대해 잘못 분류하는 확률을 최소화하는 구조적인 위험 최소화(Structural Risk Minimization) 방법에 기초하고 있다.

SVM의 가장 간단한 형태는 그림 2-1과 같이 최대 마진(Maximize Margin)을

가지고 최적 분류 초평면(OSH)을 결정해 선형 분류기로 사용하는 것이다. 즉, SVM은 학습 집단에서 마진을 최대화 하는 결정면을 찾아내는 알고리즘이라 할 수 있다.

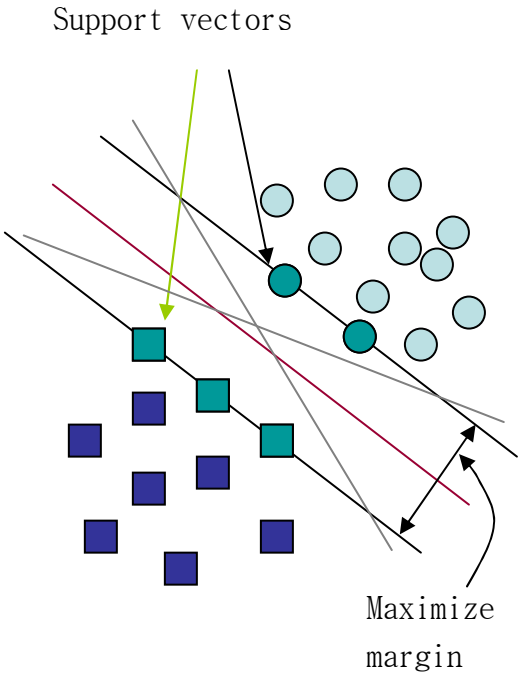

그림 2-1 선형 SVM

기본적으로 SVM은 선형 분리가 가능한 문제에서 출발하지만 모든 문제가 선형적으로 분리 될 수는 없다. 이처럼 입력 데이터의 선형 분리가 불가능할 경우 입력 공간을 분리하는 비선형 결정면을 이용하게 되는데, 비선형 결정면의 식을 분석적으로 계산해 낸다는 것은 매우 어려운 일이다.. 이런 경우 SVM에서는 그림 2-2과 같이 고차원의 속성공간을 효율적으로 처리하

기 위해서 커널함수를 이용하여 입력 벡터 x를 고차원 속성공간에서의 벡터로 변형 후 선형의 경계선을 찾는 문제로 전환하게 된다.

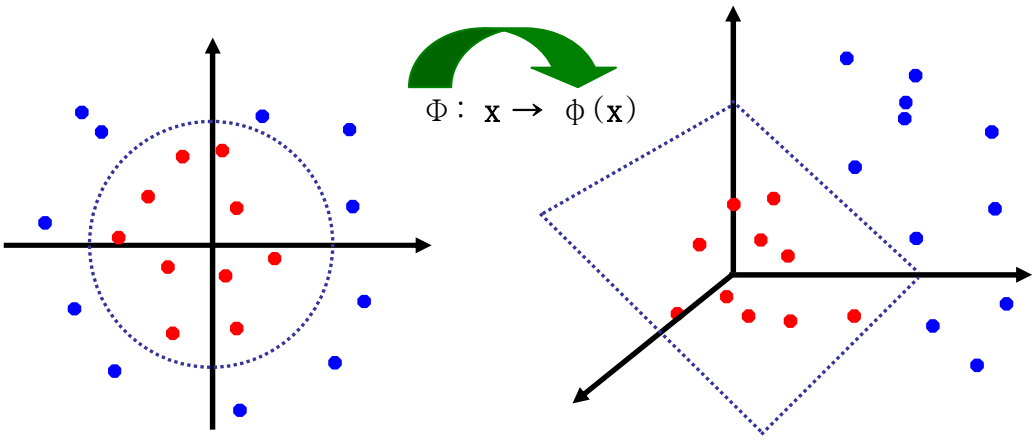

그림 2-2 차원 공간으로 사상시키는 2차원 데이터

커널 함수엔 RBF, Polynomial 등 여러 가지가 있으나 문서분류에는 일반적으로 선형 커널 함수가 사용이 간편하고 성능도 우수하다고 알려져 있다.[24], [25]

2.1.3 카이 제곱(Chi-Square)

카이제곱은 변수의 범주에 따라 관측 빈도와 기대 빈도간의 차이를 근거로 한 통계량이다. 두 빈도 값의 차이를 이용하여 독립성 혹은 관련성의 여부를 판단하게 되는데, 차이가 클수록 카이제곱의 값은 커진다.

먼저 n개의 독립적인 임의의 변수 χ_i가 이론적 평균 μ_i와 표준편차를 가지고 가우시안 형태로 분포되어 있다고 가정하면, 그 합은 식(6)과 같은 카이제곱이 된다.

$$X^2 = \sum_{i=1}^{k} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2 \qquad (6)$$

특성으로는 독립된 표본으로부터 계산된 카이제곱 통계량을 더하면 그 합도 카이제곱 통계량이 되고, 자유도는 각 자유도들의 합이 된다. 또한 카이제곱 통계량을 계산하는데 만약 모수의 추정치를 사용하였다면 그 통계량의 자유도는 추정된 모수의 개수만큼 감소하게 된다. 즉 모수를 알고 있을 경우의 자유도에서 추정된 모수의 수를 뺀 값이 자유도가 된다.

Z_1, Z_2, …가 표준 정규분포를 따르는 확률 변수일 때, $C_k = \sum_{i=1}^{k} Z^2{}_i$는 카이제곱분포를 따르는데, 이때 자유도(degree of freedom)는 k이다.

C_k의 밀도함수는 감마분포의 특수한 경우인데, 식(7)과 같은 식으로 표현할 수 있다.

$$f(x;k) = \begin{cases} \dfrac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2}, & x > 0 \\ 0, & x \le 0 \end{cases} \qquad (7)$$

카이제곱 분포의 모양은 자유도에 의존한다. 카이제곱 분포를 따르는 확률변수는 양의 값만 가지며 원점인 0에서 시작하여 축의 양의 방향으로 곡선을 가진다. 자유도가 작으면 왼쪽으로 치우친 모양으로 비대칭이며, 자유도가 커짐에 따라 곡선이 대칭에 가까워지며 자유도가 큰 경우 정규곡선과 같

은 모양을 가진다. 자유도가 1, 2인 경우 분포의 최고점은 0에서 일어나며, 자유도가 3이상인 경우 최고점은 (자유도-2)이다. 그림 2-3는 자유도에 따른 카이제곱 분포를 나타낸다.

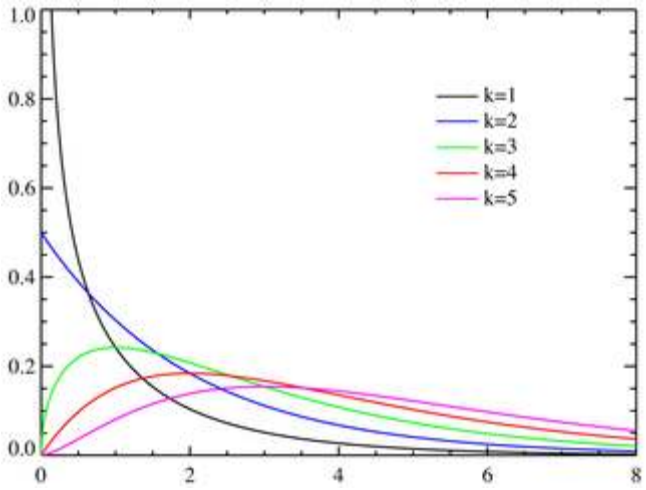

그림 2-3 자유도에 따른 카이제곱 분포

카이제곱 분포를 이용한 검증은 두 범주형 변수가 서로 관계가 있는지 독립인지를 판단하는 통계적 검증 방법이다.이 검증의 목적은 적합도 검증,독립성 검증, 동일성 검증으로 나뉜다.

적합도 검증은 조사에서 얻은 자료가 어떤 특정한 분포를 얻는가를 알고자 할 때이다. 이것은 관찰된 분포가 모집단에 대해 기대하는 분포와 어느 정도 일치하는가를 검증하는 것이다.독립성 검증은 자료의 두 개의 변수에 따라 분류시켜 표를 만들었을 때 두 변수간의 관계가 있는지를 검증하는 것으로,분할표의 행과 열의 합계에 의한 기대치와 각 칸 안의 실제 값을 비교하여 검정하는 것이다. 동일성 검증은 두 개 이상의 다항분포가 동일한지를

검증하는 것이다.

실험결과를 측정하거나 통계조사를 할 때 나타난 관측 값들이 어떤 속성에 따라 분류되어 도수로 주어지는 경우 이러한 자료를 범주형 자료(categorical data)라고 한다. 물론 이런 형태의 자료는 명목척도에 의해서 측정된다. 카이제곱 검증은 이런 명목 척도를 검증하는데 사용되는데, 피어슨(Pearson)에 의해서 제안되었기 때문에 피어슨의 카이제곱 적합도 검정이라고도 한다.

2000년 Oakes등은 문서 분류에 카이제곱 모델을 사용하였다[22]. 그들의 연구에서 카이제곱은 특별한 주제의 어휘 특성을 구분하는데 이용되었다.특성에 의해 구분되는 서로 상반되는 말뭉치(Corpus)를 하나의 큰 일반적인 말뭉치라 한다면, 그 말뭉치에 속하는 단어들은 해당되는 말뭉치의 특성을 가진다. 일반적인 말뭉치 에서의 모든 단어에 대한 계산을 한 뒤, 각 단어는 상반되는 두 말뭉치의 어느 한쪽 특성을 가지게 되고 이 특성을 기준으로 단어 리스트를 만든다. 각각의 단어들은 특성이 어느 말뭉치에 속하는가에 대한 태그를 할당 받아 단어 리스트에 저장된다.

시스템은 새로운 문서를 읽게 되고,문서 내의 각각의 단어들은 단어 리스트에 의해 어느 한쪽 말뭉치의 특징에 가깝게 나타난다.만약 그 단어가 특정 말뭉치에 가깝다면 그 말뭉치를 나타내는 태그를 할당 받고,문서 점수에 1을 더한다. 그렇지 않으면 상반되는 말뭉치를 나타내는 태그를 할당 받는다. 그리고 문서 점수는 1점을 감한다. 그 단어가 키워드 리스트에 있지 않으면 그것은 무시된다. 최종 점수에 의해 정렬된 문서의 리스트를 생산한다.

마지막으로 한계값는 일반적인 말뭉치에서 범주 내 문서 비율의 결정에 의해 계산된다. 한계 값은 리스트에서 얼마나 많은 문서가 범주 내에 있고,얼마나 많은 문서가 범주 밖에 있는지를 결정하는데 이용된다.같은 비율의 새로운 문서가 정렬된 리스트에서 높은 점수를 가지면 범주 내의 문서로 분류

하고,그렇지 않으면 범주 밖의 문서로 분류한다.

2.2 휴리스틱(Heuristic) 방법들

기존의 덧글 스팸 방지를 위한 여러 방법들을 소개하면 아래와 같다. [4]

- 덧글을 위한 로그인 절차.
- Capcha를 이용한 Turing test[7].
- HTML 태그 제한.
- 오래된 블로그 글에 덧글을 제한.
- IP 블랙리스트를 유지[11].
- 외부 링크를 내부 링크로 리다이랙트(redirect).
- 동일한 덧글이 한꺼번에 올라오는 것을 제한. (“throttling”)
- rel=”nofollow” 태그를 사용[12].
- 블로그 글과 동일한 언어로 덧글 제한.
- Language Model을 이용한 방법[2]

덧글을 이용한 스팸 로봇을 차단하기 위한 방법은 효율적인 방법처럼 보이나 실제 포털 블로그에서 무작위로 ID를 만들어 스팸을 올리는 것을 보면 확실한 해결책은 되지 못하는 것 같다. Capcha는 스팸 로봇인지 사람인지 확인하고자 하는 일종의 Turing test이다. 하지만 사람이 덧글을 쓰고자 할 때 Capcha에 의해 신경을 다른 곳으로 쏠리게 함으로 덧글의 신선도에 악영향을 끼칠 수도 있어서 실제 포털이나 블로그에서 그리 즐겨 쓰이지 못하고 있다. 또한 이상한 영어나 숫자를 잘 못 읽는 장애인이나 노약자에게는 상당한 인터넷 진입장벽이 될 수 있다.

덧글에 Html 태그를 쓰지 못하게 할 경우 일단 스팸성 데이터를 업로드가 가능하기 때문에 스팸이 난무할 가능성은 충분하고 IP로 인한 제한도 proxy를 이용하면 충분히 스팸 로봇을 돌리 수 있기 때문에 완벽한 해결책은 되지 못한다.

현재 많은 검색엔진은 html의 a 태그에서 rel = “nofollow”옵션을 주어 링크에 대해서 링크 점수를 무작위로 올리는 것을 방지하고 있다. 이는 덧글 스팸이 랭킹을 올리기 위한 덧글일 경우 검색엔진 랭킹에만 효과적인 방안이다. 실제로 블로그를 사용하는 사용자들은 스팸 방지 효과를 전혀 못보고 검색엔진 사용자들만이 그 효과를 보게 되므로 근본적인 스팸방지 대책은 되지 않는다.

블로그와 동일한 언어로 덧글을 제한하는 옵션은 한때 한글 블로그에 영어 스팸 덧글이 난무할 때 유행하던 것으로 초기 상당한 효과를 봤지만 영어 블로그일 경우 거의 쓸모가 없던 기능이다.

덧글 스팸이 e-mail 스팸과 다른 점이 있는데, 그것은 바로 덧글 스팸은 스팸 판정 결과를 스패머가 바로 알 수 있다는 것과 e-mail스팸의 경우는 결과를 바로 알 수 없다는 것이다. 따라서 위에서 나열된 방법은 스패머를 일시적으로 막을 수 있는 임시 방편이 될 수 밖에 없다.

2005년 www 컨퍼런스에서 덧글 스팸 제거를 위한 지금까지와 다른 접근 방법을 제시한 논문이 발표되었는데 이것이 바로 Language Model을 이용한 본문과 덧글 그리고 덧글이 링크된 페이지간의 유사도를 비교해 스팸 유무를 판단하는 논문이다[2]. 하지만 이 논문은 같은 내용의 덧글이 동시 다발적으로 올라오는 현실적인 스팸 덧글 특성에 대해서 필터를 학습하지 못하는 한계가 있다. 덧글 자체만으로 스팸인 것에 대해서 기본적인 스팸 가중치를 부여하지 못하고 단지 상호간에 유사성에 기반을 두고 덧글 스팸 판정

을 하기 때문이다. 이는 비교사(non-supervised) 기반의 필터링 시스템의 한계라 생각한다.

2.3 역 카이제곱 분류 알고리즘

역 카이제곱 알고리즘은 Paul Graham 의 베이지언 스팸 필터의 단점을 보완하고자 나온 알고리즘이다.[17]

2.3.1 역 카이제곱을 이용한 스팸 필터

역 카이제곱 스팸 분류 방법은 Paul Graham의 베이지언 확률을 이용한 스팸 필터[8]를 보완하기 위해 나온 개념으로서 베이지언 확률에서 나온 독립성 가정의 문제, 희소 단어 처리, 단어의 확률의 오류에 대한 문제점을 보완하기 위해 Robinson이 제시한 알고리즘이다[5].
Paul Graham은 주어진 단어의 확률을 구하기 위해 아래와 같은 식을 제안했다.

$$P(S \mid W) = \frac{P(W \mid S)}{P(W \mid S) + P(W \mid H)} \qquad (8)$$

S : Spam collection
H : Ham collection, W : word

Paul Graham 방법에서 문제점으로 지적된 희소한 단어에 대한 확률 계산의 문제점을 개선하기 위한 방법이 추가 되는데, 예를 들어 정확히 한 개의 스팸 메일이 입력이 되었고 그 메일에서 처음 나오는 하나의 단어의 스팸

확률은 100%가 되게 된다. 처음 나온 단어가 한번 스팸에 나왔다는 이유 하나만으로 앞으로 판단될 모든 메일에서 그 단어가 나왔을 때 스팸 확률이 100%로 계산되게 될 것이다. 사실 그 단어는 정확한 확률적인 정보를 가지기에는 미약한 근거를 가지고 있는 셈이다. 사람은 미래에 받을 이메일에서 위에서 판단된 단 한번만 나온 단어가 존재한다고 해서 100%의 신뢰도를 주지는 않는다. 그러한 이유는 우리가 다른 배경적인 지식을 이용하기 때문인데, 이러한 배경적인 지식 덕분에 한번만 스팸에 나온 단어에 대해서 100%의 신뢰를 주지 않는 것이다.

이러한 배경에 의해서 Robinson 은 아래와 같은 확률의 신뢰도 식을 제안했다.

$$f(W) = \frac{(s \times x) + (n \times P(S \mid W))}{s + n} \qquad (9)$$

s : 배경 지식에 대한 신뢰 강도
x : 배경 지식을 기반으로 한 단어의 초기 확률
n : 수신 문서 중 단어 W를 포함하는 문서의 수

이렇게 구한 단어의 확률 값들을 결합하기 위해 피셔(Fisher)의 역 카이제곱(inverse Chi-square) 검증을 적용한 후 하나의 척도 H 를 구한다. 이렇게 구한 확률의 결합은 본질적으로 스팸에 큰 영향을 미치는 단어들의 확률을 1에 가까운 값으로 계산하는 것이 아니고, 햄에 큰 영향을 미치는 단어들의 확률 결합을 0 에 가깝게 만들기 때문에 또 다른 척도 S 를 계산하게 된다. 이 척도 S 역시 단어들의 확률을 결합하지만,이번에는 단어들의 확률을 (1 - f(w))로 적용한다. 마지막으로 주어진 메일이 스팸인지 아닌지를 판단하기 위해 두 가지 척도를 결합한 제 3 의 척도 I 를 사용하게 된다. 앞에서 지적한 문제점의 해결책으로 단어의 독립성 가정은 연관성을 찾기가 어렵기 때

문에 베이지언 방법에서 사용된 가정을 그대로 적용한다. 희소단어의 처리는 사용자의 배경 지식에 대한 입력 값인 신뢰의 강도 s 와 어떤 단어가 처음으로 스팸에 나타날 확률 x 를 이용하여 처리하였고, 단어의 확률 계산은 위에서 설명한 것처럼 확률의 결합 H 의 역 확률인 S 를 구하여 해결한다. 마지막으로 비대칭은 확률 H와 확률 S를 결합한 새로운 확률 I로서 해결을 한다.

아래의 3 가지 식은 위에서 언급한 H, S, I 식을 의미한다.

$$H = C^{-1}(-2\ln\prod_{W} f(W), 2n) \qquad (10)$$

C^{-1} : 카이제곱 함수의 역함수

n : 문서 내 단어의 총 개수

$$S = C^{-1}(-2\ln\prod_{W} (1 - f(W)), 2n) \qquad (11)$$

$$I = \frac{(1 + H - S)}{2} \qquad (12)$$

위에서 구한 I 는 문서가 스팸에 가까울수록 1 에 가까운 값, 정상에 가까울수록 0 에 가까운 값을 가지는 스팸 및 정상 표시자가 된다. 물론 0.5 의 값을 가지는 문서는 결정 불가능한 메일을 의미한다. 이러한 gray area 를 표현하는 것이 가능한 것이 역 카이제곱 알고리즘의 장점 중에 하나이다.

3. 스팸 필터의 설계 및 구현

본 논문에서는 나이브 베이지언 분류기와 역 카이제곱 분류기, 그리고 동시출현 자질을 이용한 역 카이제곱 분류기를 구현했다. 단순히 구현에만 관점을 두자면 3 가지 알고리즘이 코드상으로 크게 다른 점은 없다. 따라서 이 장에서는 동시출현 자질을 이용한 역 카이제곱 분류기에 중점을 두고 설명을 하겠다.

2 장에서는 다양한 문서 분류 알고리즘과 실제 본 논문에서 제안하고자 하는 역 카이제곱 알고리즘에 대해서 알아보았다. 본 장에서는 실제 본인이 제안한 동시출현 자질을 이용한 역 카이제곱 방법을 사용해 실제 덧글 스팸을 필터링하는 시스템에 대해서 설명하고자 한다.

3.1 설계 방향

분류기는 학습과정을 거치고 그 학습 결과를 메모리와 파일에 동시에 저장하는 구조로 했다. 왜냐면 나중에 다시 학습하는데 걸리는 시간을 단축하기 위해서이다.

직접 분류한 덧글 데이터를 이용해 학습을 시켜 단어에 대한 확률 정보를 수집하는 것으로 학습단계는 일단락 되고 실제 본문과 덧글이 들어왔을 때 여러 전처리 단계를 거쳐 문장의 중요 단어에 대한 확률 정보를 추출해 판정 자질로 활용해 덧글의 스팸 유무를 판단하게 된다.

하나의 완벽한 시스템으로 구축되는 것을 목적으로 구현을 했고 실제 소

켓 통신을 위한 테스팅 환경도 구축하였다.

3.2 시스템 구조.

시스템은 크게 학습 단계와 테스트 전처리 그리고 테스트 단계로 나눠진다.

학습 단계에서는 덧글의 스팸성 판정 데이터들이 입력으로 들어가서 스팸 필터를 학습하게 된다. 이때 POS Tagger[9]를 이용해 명사만 추출한다.

테스트 셋의 경우는 그림 3-1 과 같은 형식으로 한 레코드당 한 줄씩 입력을 받게 된다. 그림에서와 같이 '<comment: * >' 형식의 comment 자체를 의미하는 문장이 들어가고, 그 뒤에 그 해당 comment 가 스팸덧글인지 정상덧글인지 확인하는 '<spam:'1 또는 0',>'테그가 들어가게 된다. 값에 대한 정의는 아래와 같이 하기로 하겠다.

정상덧글 = 0

스팸덧글 = 1

따라서 아래의 3-1 에서의 그림의 첫 번째 레코드는 스팸덧글이다.

<comment:71772 Title of washington va real estate><spam:1> ← 스팸덧글

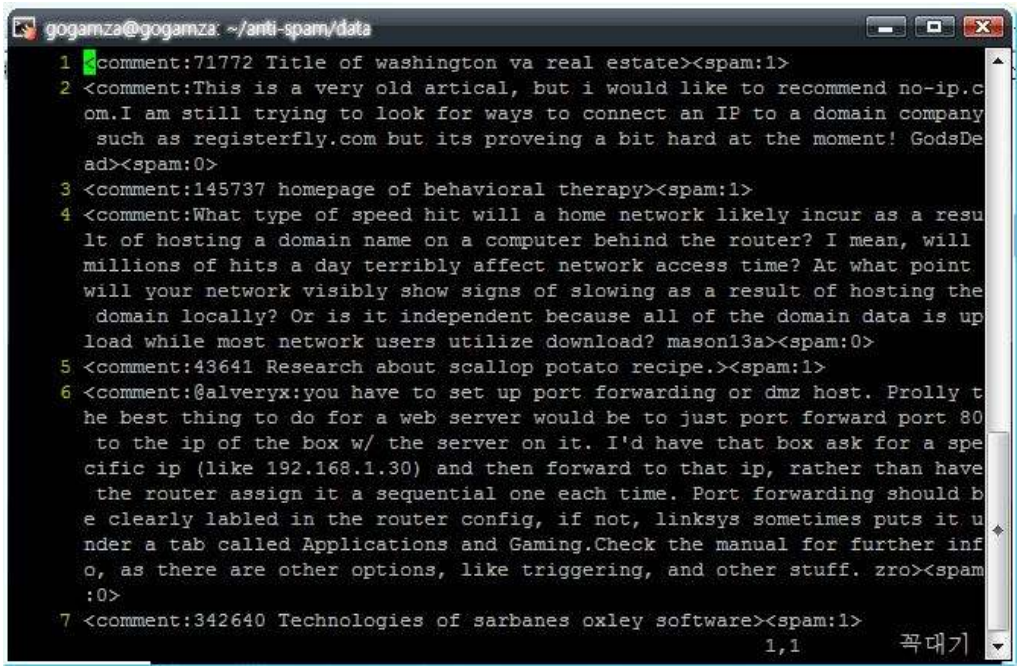

그림 3-1 학습 데이터 포맷

이러한 학습 데이터를 이용해 학습을 완료하고 그 데이터를 아래와 그림 3-2 와 같은 포맷의 파일구조로 저장을 하였다.

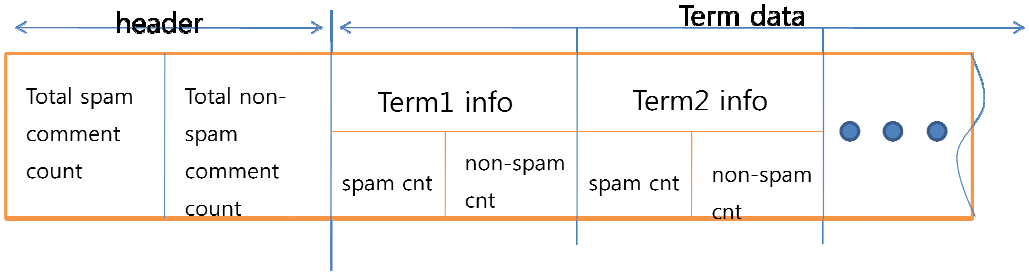

그림 3-2 단어 확률 정보 저장 구조

이제부터 테스트 단계에 대해서 설명을 하겠다. 먼저 데이터 입력 포맷 구조는 아래와 같다.

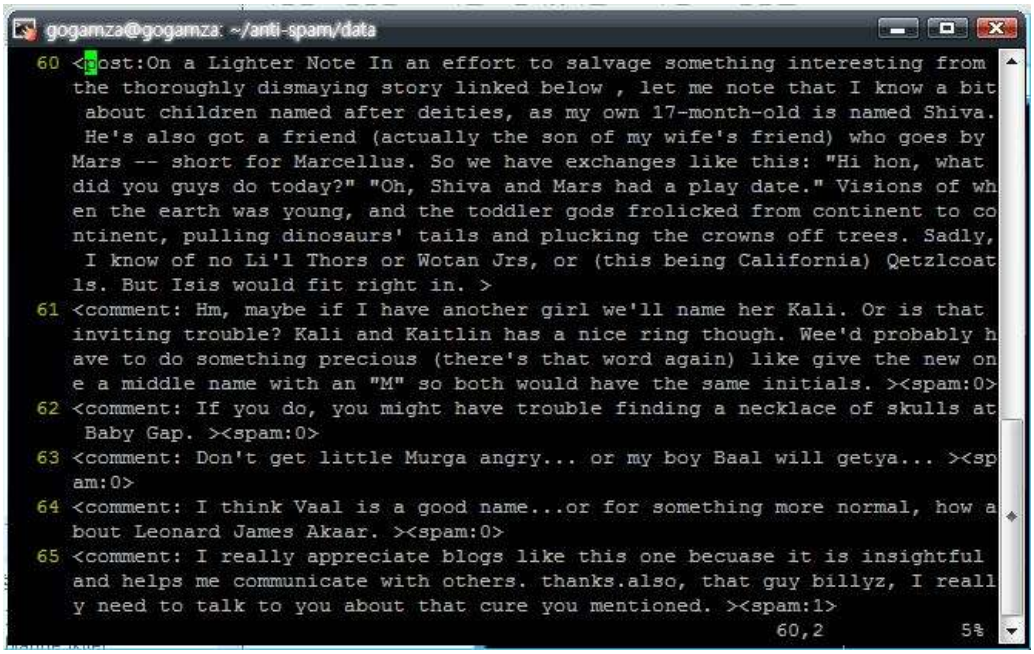

그림 3-3 테스트 데이터 포맷

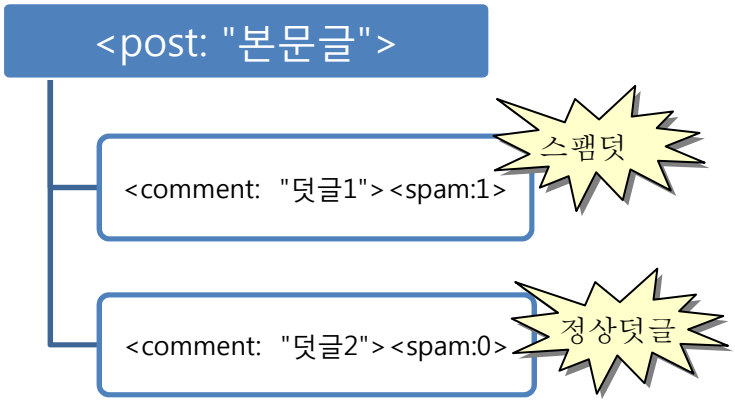

그림 3-4 테스트 데이터 포멧 구조

테스트 데이터의 포맷은 위와 같이 한 개의 본문 글에 여러 개의 덧글이 붙는 구조로 입력되어 있으며 기 판정 결과들은 학습 데이터의 경우와 같다.

테스트 전처리 단계에서는 덧글과 덧글과 관련된 본문을 입력으로 받는다. 이때 덧글은 명사만 추출해서 |P(W) - 0.5|값이 가장 큰 총 5 개의 중복을 1 번만 허용하는 리스트를 각 덧글 확률 정보로 유지하며, 본문은 tf-idf 단어 가중치를 계산하기 위해 (13)식을 이용 본문에서 핵심어로 판단되는 단어리스트를 내림차순 정렬해서 유지한다.

$$tf - idf_{t,d} = tf_{t,d} \times \log \frac{N}{df_t} \qquad (13)$$

이 식에서 $tf_{t,d}$ 는 문서 내 특정 단어의 빈도수를 의미하고, N 은 코퍼스 내의 총 문서 수, 그리고 df_t 는 코퍼스 내에서 단어의 출현 횟수를 의미한다.

덧글에서 확률의 영향력이 큰 단어 5 개(판정 자질 명사)만 유지하는 이유는 수집한 스팸의 평균 길이가 40 char 즉 8 단어 내외였고 실제로 스팸판정에 가장 영향을 많이 끼치는 단어를 포함하는 작업을 함으로서 덧글 길이의 영향을 최소화 하려고 한 것이다. 그리고 빈도수에 대한 정보를 수용하기 위해 1 번 반복된 단어는 허용했다[6].

10 개로 제한한 본문의 주제어를 뽑는 작업은 주제어가 덧글에 포함 유무에 대한 확률 값을 계산하기 위한 선 작업(Pre-Processing)이다.

테스트 단계에서는 덧글의 5 개의 판정 자질 명사 집합에 대해서 (2)번 식을 이용한 확률을 구하고 덧글과 본문에 동시에 존재하는 전처리 단계에서

뽑은 주제어가 있을 경우 이들에 대해 아래(3.3)에서 소개될 식을 이용한 추가 확률을 구하는 것이다. 이들 단어들의 확률을 기반으로 (10), (11), (12)번 식을 사용해 문서의 스팸 유무를 판별한다.

시스템의 전체적인 구조는 아래 그림 3-2 과 같다.

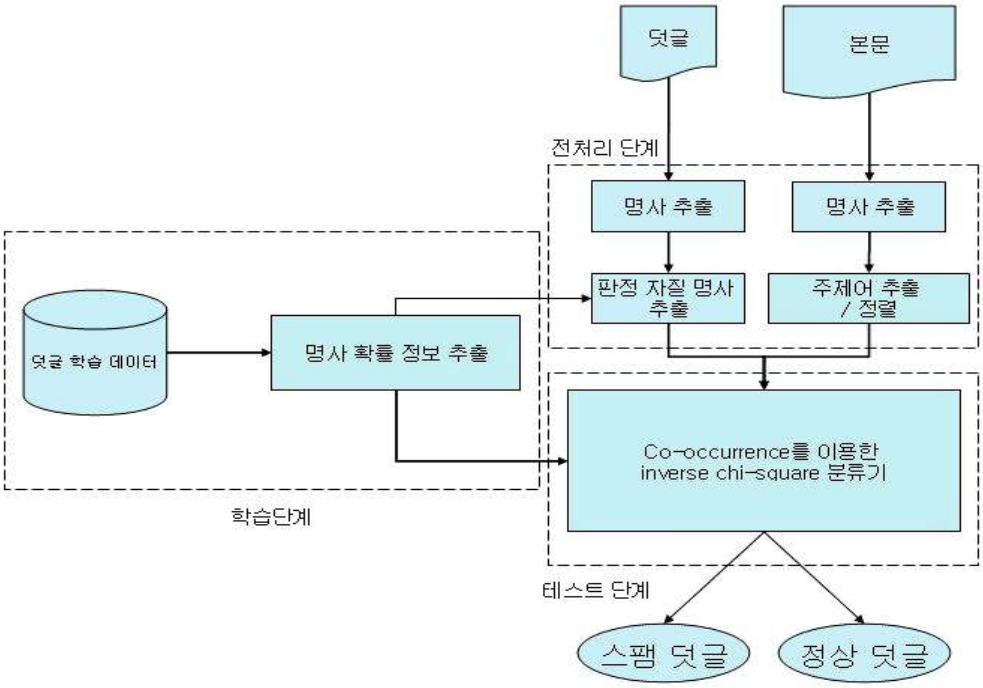

그림 3-5 시스템 구조

3.3 동시출현 단어 자질 정보

덧글과 본문에 동시 존재하는 주제어에 대한 확률 값을 알아야 하는데 이는 (9)번 식을 이용하면 유도할 수 있다.

(9)식을 다시 쓰면 아래의 (14)번식과 같이 쓸 수 있는데.

$$P(S \mid W) = \frac{P(W \mid S) \bullet P(S)}{P(W \mid S) \bullet P(S) + P(W \mid H) \bullet P(H)} \qquad (14)$$

여기서 추가된 P(S), P(H)는 각 스팸, 정상 문서 컬렉션에서 단어의 확률을 의미하는 P(W|S), P(W|H)의 신뢰 강도를 의미하기도 한다. Graham 의 경우[8]에서 스팸과 정상문서의 비율을 동일하게 두고 실험을 하였기에 생략된 확률들이다.

동시 출현하는 단어들에 대해서 신뢰강도(degree of belief)를 주어 단어 확률을 추가해 주는 작업을 할 수 있는데 아래와 같은 식을 이용해 동시 출현하는 주제어들의 확률을 추가한다. 단어들의 신뢰강도는 본문 내에서 주제어 빈도수에 비례할 것이다.

$$d.b = \frac{t.f}{dl} \qquad (15)$$

여기서 dl 은 모든 주제어의 총 빈도수를 의미하고 t.f 는 동시 출현한 주제어의 빈도수를 의미한다.

이렇게 구현된 동시출현 주제어들에 대한 스팸 확률은 아래와 같다.

$$P(S \mid C.O.W) = \frac{P(C.O.W \mid S) \bullet (1 - d.b)}{P(C.O.W \mid S) \bullet (1 - d.b) + P(C.O.W \mid H) \bullet d.b} \qquad (16)$$

미지의 값인 P(C.O.W|H)과 P(C.O.W|S)은 각각 동시출현 단어의 스팸 코퍼스와 정상 코퍼스에 나올 확률을 의미하는데 동시출현 단어에 대해서 정상 코퍼스에 나올 확률을 올려주는 것이 이 확률식의 목적이기 때문에 본 논문

에서는 아래와 같이 정의한다.

$$P(C.O.W|S) = 0.1 \quad (17)$$

$$P(C.O.W|H) = (1- P(C.O.W|S)) \quad (18)$$

이러한 (17), (18)번 식을 사용해 도출된 동시출현 자질 확률들을 기반으로 (12)번식을 사용 최종적인 덧글 스팸 필터링을 하게 된다.

4. 실험 및 결과

4.1 실험 환경

본 논문에서 제안한 실험 시스템은 아래 표 4-1 과 같은 환경에서 3 가지 필터를 직접 모두 구현하여 실험하였다.

CPU	AMD Turion64 x 2 Dual-core
MEMORY	2GB
HDD	120GB
OS	Ubuntu Linux
COMPILER	gdc v0.24

표 4-1 실험 환경

3 가지 필터를 모두 D language 를 이용해서 Linux 기반에서 개발했고, 알고리즘상 베이지언 필터와 역 카이제곱 필터의 유사성으로 인해 그렇게 많은 개발 리소스는 들어가지 않았다. 직접 개발함으로써 3 가지 필터의 여러 파라메터상의 동일성과 환경적인 조건을 모두 동일하게 할 수 있었던 장점이 있었다.

제안한 방법의 유효성을 검증하기 위해 직접 덧글 데이터를 수집해서 분류 후 학습데이터로 사용을 하였고 테스트 데이터 셋으로는 [2]에서 쓰인 본문과 덧글, 그리고 판정데이터까지 포함된 toy corpus[3]를 사용했다.

수집 대상이 되었던 대상이 되는 블로그는 http://technorati.com 에서 IT 분야의 영문 블로그 30 개와 정치, 경제 분야의 영문 블로그 30 개를 무작위 선

택해 직접 구현한 웹 크롤러를 사용하여 덧글 데이터를 수집, 분류했다.

이렇게 수집한 학습 덧글과 테스트 덧글에 대한 정보는 아래와 표 4-2 와 같다.

컬렉션	개수
학습 덧글 (스팸/정상)	19,586 / 10,000
테스트 덧글 (스팸/정상)	612 / 329
테스트에 사용된 본문	47
총 덧글	20,198 / 10,329

표 4-2 학습 데이터와 테스트 데이터

제안하는 필터 시스템은 덧글만을 대상으로 기본적인 베이지언 스팸 필터와 역 카이제곱 필터에 대한 성능 측정을 했고, 마지막으로 동시출현 자질 정보가 포함된 역 카이제곱 필터의 성능 측정 결과를 상호 비교하였다. 여러 베이지언이나 역 카이제곱의 확률 계산시 들어가는 여러 파라메터는 모두 동일한 조건하에 두고 실험을 하였다. 따라서 알고리즘상 차이점만 제외하고 모두 동일한 환경에서 실험을 하였다.

성능 평가 방법으로는 일반적으로 스팸 필터 성능을 평가할 때 주로 쓰이는 아래와 같은 평가 방법을 사용하였다.

a: ham (correctly classified)	[true negative]
b: spam (correctly classified)	[true positive]
c: ham misclassification	[false positive]
d: spam misclassification	[false negative]
e: total number of spam(real)	
f: total number of ham(real)	

hm% : ham misclassification rate

sm% : spam misclassification rate

lam% : average misclassification rate

거짓 긍정률(false positive rate)라고도 일컬어 지는 hm 은 긍정적 클래스로 예측된 부정적 사례의 비율로 정의된다. 스팸일 경우에는 정상덧글을 스팸 덧글로 판정한 비율을 의미한다.

또한 거짓 부정률(false positive rate)라고 불리는 sm 의 경우는 부정적 클래스로 예측된 긍정적 사례의 비율로 정의된다. 본 논문의 경우에는 스팸 덧글을 정상덧글로 오분석한 비율을 의미한다.

$$hm = \frac{c}{(a + c)} \quad (19)$$

$$Sm = \frac{d}{(b+d)} \quad (20)$$

평균적인 오분류율을 의미하는 lam 값의 계산은 아래와 같이 할 수 있다.

$$lam = \log it^{-1}(\log it(hm)/2 + \log it(sm)/2) \quad (21)$$

$$where: \log it(x) = \log(x/(1-x))$$

$$\log it^{-1}(x) = e^x/(1+e^x)$$

$$\text{error} = \frac{(c+d)}{(e+f)} \quad (22)$$

$$\text{accurate} = \frac{(a+b)}{(e+f)} \quad (23)$$

리콜(recall)과 정밀도(precision)는 어떤 특징 클래스의 성공적인 검출이 다른 클래스들의 분류에 비해서 훨씬 중요한 응용에서 널리 사용되는 두 가지 측정 기준이다. 이 기준들의 공식적인 정의는 다음과 같다.

$$\text{recall} = \frac{b}{e} \quad (24)$$

$$\text{precision} = \frac{b}{(b+c)} \quad (25)$$

정밀도는 분류기가 긍정적 클래스로 선언한 그룹에서 실제 긍정적 사례로 판명되는 항목들의 비율을 결정한다. 정밀도의 값이 클수록 분류기에 의하여 검출되는 거짓 부정(false negative) 오류의 수는 적어진다. 리콜은 분류기에 의하여 정확하게 예측되는 긍정적 사례의 비율을 측정한다. 리콜 수치가 큰 분류기는 부정적 클래스로 잘못 분류되는 긍정적 사례의 수가 매우 적다.

사실상 리콜의 값은 참 긍정률(true positive rate)과 같다.

정밀도와 리콜 수치를 모두 최대화 시키는 모델을 구축하는 것은 분류 알고리즘의 중요 도전과제이다.

정밀도와 리콜은 f_1-measure 라고 알려진 또 다른 기준으로 요약될 수 있다.

$$f_1 - measure = \frac{2 \times precision \times recall}{precision + recall} \qquad (26)$$

이론적으로 f_1-measure 는 리콜과 정밀도의 조화 평균을 의미한다. 즉,

$$F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}} \qquad (27)$$

이다. 두 숫자 x, y 의 조화 평균은 두 수 중 적은 숫자에 가까운 경향이 있다. 따라서 F_1-measure 치가 높다는 것은 정밀도와 리콜 모두가 상당히 크다는 것을 보장한다.

4.2 실험 결과

위에서 제시한 실험 환경과 측정 척도를 기준으로 3 가지 필터를 비교 실험해 봤다. 특히나 이런 분류기의 비교에서는 에러에 상당히 민감한데, 특히나 여기서 제시한 방법인 Hm, 일반적으로 False Positive 에러라고 알려진 척도가 필터의 성능측정에 중요한 기준이 되기도 한다.

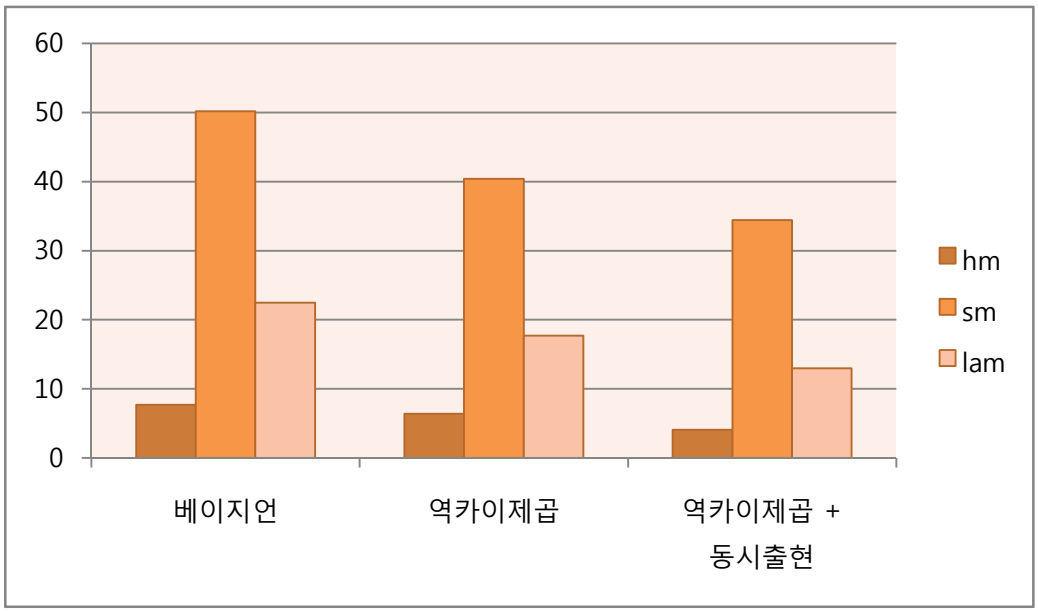

표 4-3 hm, sm, lam 결과

3 가지 필터 중에서 '역카이제곱 + 동시출현자질' 로 구현한 필터가 hm, sm, lam 부분에서 가장 좋은 성능을 보여주는 것을 알 수 있다.

이곳에서 보면 스팸덧글을 정상덧글이라고 분류해주는 sm 수치가 3 가지 필터가 모두 높은 것으로 보여지는데, 이런 원인이 되는 것이 이러한 스팸 덧글의 평균 텀 개수가 8 텀 이하라서 판정에 쓰일 자질이 거의 없는 덧글이 대부분 이였다는데 문제점이 있었다. 물론 이 부분은 제안한 방법을 써서 조금 줄어들기는 했지만 본 논문에서 제시한 방법 말고 다른 추가 자질을 발굴의 필요성을 역설하는 부분이 아닐까 한다. 오분석 되었던 예제를 보자면 "Just a test" 라는 덧글이 과연 스팸덧글일까 아닐까 하는 판정의 문제도 있는 걸로 보인다. 내가 쓴 테스트 셋의 판정은 본문과 무관한 동문서

답의 경우에 스팸덧글로 판정한 것으로 보인다.

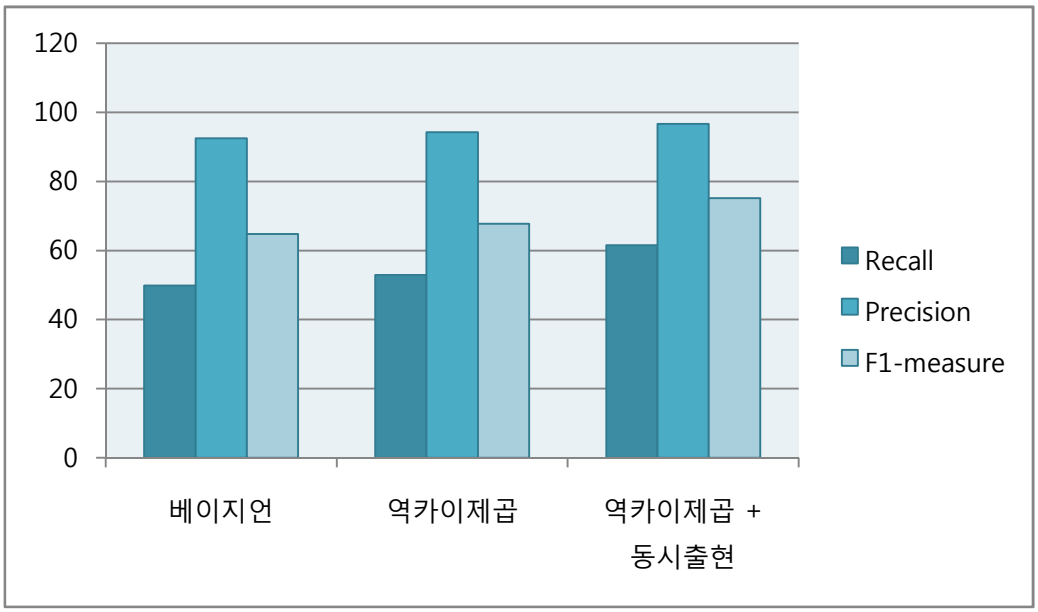

표 4-4 리콜(Recall), 정밀도(Precision), F_1-measure 결과

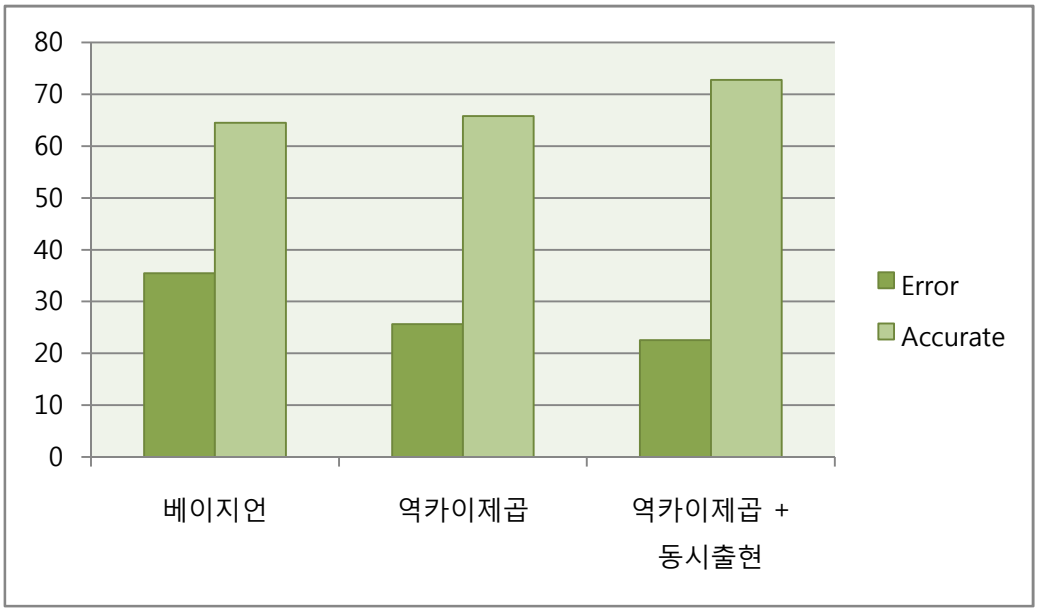

표 4-5 에러율(Error rate)과 정확도(Accurate rate) 결과

평가방법(%)	베이지언	역 카이제곱	동시출현 자질 + 역 카이제곱
Hm	7.69	6.38	4.07
Sm	50.16	40.40	34.44
Lam	22.45	17.70	12.99
Error	35.49	25.61	22.52
Recall	49.83	52.92	61.52
Precision	92.46	94.21	96.68
F_1-measure	64.76	67.77	75.19
Accurate	64.50	65.78	72.79

표 4-6 종합적인 비교 실험 결과표

스팸 필터의 성능을 평가하는 본 논문에서 제공한 모든 방법에 대해서 동시출현 자질을 추가한 역 카이제곱 필터가 가장 좋은 성능을 보여주는 것을 실험 결과 데이터를 통해 알 수 있다.

4.3 결과 분석

4.2.1 동시 출현 자질의 유효성

실제 역 카이제곱 필터와 비교할 때 Hm, Sm, Lam, Error, Recall, Precision 등 모든 측정 결과에서 동시출현 자질을 추가한 역 카이제곱 필터가 좋은 결과를 보여준다. 이는 본문과 덧글간 주제어 동시출현 정보가 덧글의 스팸 확

률을 평가하는데 중요한 요소로 쓰일 수 있다는 것을 보여준다.

4.2.2 오류 분석(Hm, Sm)

분류 필터는 Hm(False positive)가 성능 평가에 중요한 요소로 쓰인다. 실제 정상적인 덧글이 스팸으로 판단되는 손실이 스팸이 정상으로 판단되는 손실에 비해 크기 때문이다. 하지만 동시출현 정보를 포함한 필터가 가장 낮은 오류율을 보여주고 있어 오류에 강한 필터임을 보여주고 많은 정상덧글들이 주제어를 포함하고 있다는 것을 보여준다.

이러한 오류율이 동시출현 정보를 사용함으로써 낮아진 이유는 드물게 출현해서 단어의 스팸 확률이 초기값 0.4 에 근접하게 평가되어 덧글의 스팸 평가에 영향을 거의 미치지 않는 단어들이 동시출현 확률 계산식에 의해서 중요 단어로 계산식에 포함되면서 나온 결과이다. 덧글에 본문에서 주로 쓰이는 주제어가 포함이 됨으로 인해 이 덧글의 스팸 확률은 현격하게 낮아지게 되어 오류율이 적어지게 된 것이다.

하지만 Sm(False Negative)가 세가지 필터에 대해서 매우 높은 것에 대해서 생각해볼 필요가 있음을 실감해서 직접 제안한 동시 출현 정보를 추가한 역 카이제곱 필터의 Sm 결과를 가지고 오류 분석을 해보았다.

오류 분석 방법은 오류로 출력된 덧글에 대해서 단어를 추출해 그것들을 빈도수에 대해서 정렬을 했다. 데이터의 특성상 long tail 형식으로 특정 단어에 집중하는 현상이 있어서 지면관계상 그래프는 상위 30%를 차지하는 단어에 대해서만 넣었다.

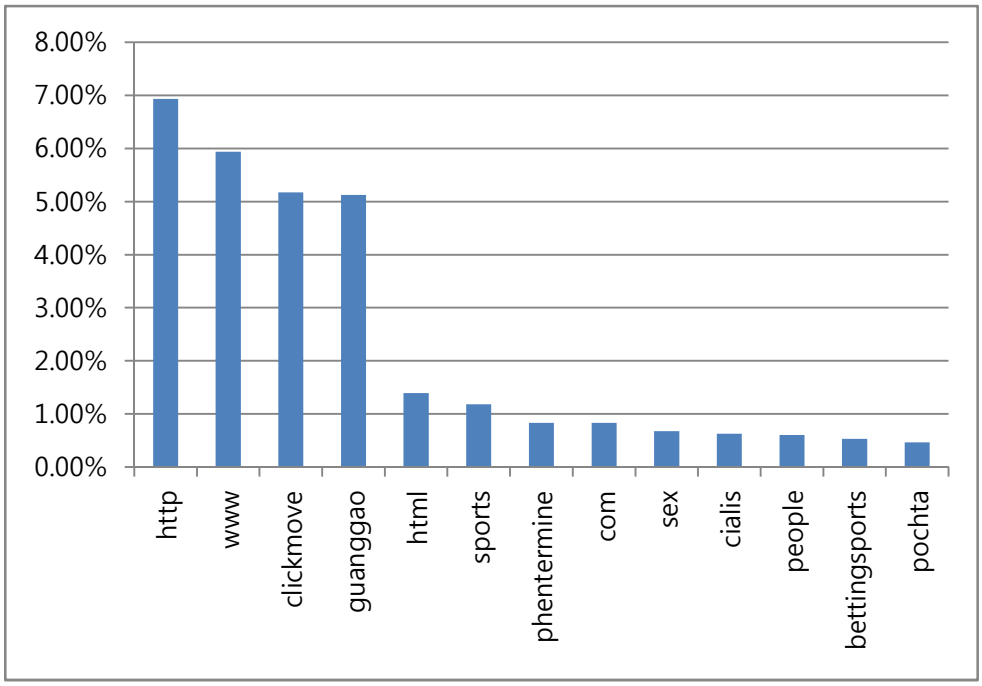

표 4-7 Sm(False Negative) 분석 결과

위 표를 보면 가장 많은 비율을 차지하는 단어가 http, www, html 등 URL 관련 단어인 것을 알 수 있다. 물론 테스트 도중에 이런 단어의 확률을 확인해본 결과 http 는 97.85 의 스팸 확률을 가지고 있었던 단어로 나왔으며, www 의 경우는 97.53 의 스팸 확률을 가지고 있던 것으로 나온다. 이러할 결과가 나온 이유는 URL 들을 구성하는 단어 자체가 스팸성 임에도 불구하고 다른 정상인 텀과 같이 출연함으로써 상쇄가 된 경우로 볼 수 있다. 또한 URL 주소들에 대해서 www, http 같은 텀을 추출해서 확률 정보에 넣는 횟수 또한 2 회로 제한을 했던 결과였음을 알 수 있다.

221 개의 오분석 덧글 중에 33 개의 덧글이 http 를 포함하고 있었고, 특정 덧글에 집중되어서 남발된 것으로 확인 되었다. 따라서 URL 자체에 대한

www, http, html 같은 텀에 대해서 페널티를 무조건 주기 보다는 특정 횟수 이상의 남용에 대한 페널티로 접근을 하는 것이 바람직한 방법이지 않을까 한다. 물론 스팸 판정을 위해서 남발 횟수를 자질로 넣을 경우 학습 기반으로 학습셋에서 도출될 수 있어야 하는 부분일 것이다.

오류 분석을 통해서 스팸을 판정하는데 URL 이 중요한 자질로 선택될 수 있다는 것을 보여주는 결과라 생각한다.

4.2.3 **Grey Area 에 대한 고찰**

제안한 방법으로 전체적인 성능향상은 있었으나 필터에서 grey area 로 판단되는 전체의 8.6%의 덧글에 대한 판단 기준과 방법에 대한 과제가 남게 된다. 이런 경향의 덧글들은 대체적으로 짧고 본문에서 나온 주제어들이 전혀 포함되어 있지 않아 본문에 대한 의견에 관한 덧글이라기 보다는 다분히 형식적인 덧글임에 판단의 모호함이 따른다.

실제 실험에서 grey area 영역을 좁혀서 판단률을 높여보려 했지만 거꾸로 정확도가 떨어지는 현상이 있는 것으로 봐서 grey area 에 대한 다른 고민들이 필요할거라 생각한다.

실험 결과 로그를 분석하면 “Nice Site!”라는 덧글이 grey area 로 판단이 되었고 그 덧글에 포함된 링크는 포르노(porn) 사이트로 연결이 되어 있었음이 확인 되었다. 따라서 그 덧글과 본문과의 연관성을 판단하기 힘든 이런 종류의 덧글일 경우 추가적인 자질을 발굴함으로 grey area 영역을 좁히면서 성능을 높일 수 있을 것이라 생각한다.

4.2.4 주제어를 포함한 스팸 덧글인 경우

무분별하게 본문에 나온 주제어를 포함한 스팸 덧글을 스팸로봇이 배포하게 된다면 이 알고리즘의 성능은 장담하지 못한다.

하지만 로봇이 본문의 데이터를 분석해 주제어를 정확하게 뽑아내야 된다는 숙제가 남게 된다. 본지에서는 tf.idf 를 사용해 전체 컬렉션에서 빈도수를 기반으로 뽑아냈는데 로봇이 컬렉션을 다른걸 쓴다면 전혀 다른 주제어가 나올 가능성이 있어 로봇 자체의 스팸 덧글 게재 성공률도 그리 높지 않으리라 본다.

5. 결론 및 향후 연구 과제

본 논문에서는 단편적인 확률기반 e-mail 스팸 필터의 필터링 대상 위주의 처리 방식을 탈피한 추가 자질을 발굴 함으로써 스팸 필터의 성능을 개선할 수 있다는 것을 밝혔다. 본문과 덧글의 주제어 동시출현 자질정보를 필터 확률정보에 추가함으로써 전체적인 성능 향상을 실험 결과로 보여준다.

제안한 방법은 단편적인 블로그 덧글 필터링에만 한정된 방법이 아니다. 본문과 덧글의 연관성이 보장된 어느 연결된 미디어의 스팸 필터 자질로 쓰일 수 있다. 따라서 일반적인 게시판이나 위키의 덧글 필터링 시스템에도 이러한 자질이 성능을 발휘하리라 생각한다.

본지에서는 역 카이제곱 방법의 필터와 동시출현 정보를 조합하였지만 여타 다른 분류 알고리즘과도 동시출현 정보를 결합 할 수 있을 것이다. 추후 SVM(Support Vector Machine)과 같은 다른 분류 알고리즘들을 기반으로 이러한 자질을 추가한 성능 상호 평가를 해보는 것은 추후 과제로 남겨두었다.

또한 고찰에서 제시한 grey area 부분 문제를 해결 할 수 있는 또 다른 추가 자질을 발굴하는 연구도 의미가 있을 거라 생각한다.

참고 문헌

[1] Aaaron Emigh, Automatically Detecting Textual Blog Spam at MIT Spam Conference (2007)

[2] Mishne, G., D. Carmel, et al. Blocking Blog Spam with Language Model Disagreement. Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (2005).

[3] Gilad Mishne , Toy corpus of spam in blog comments (2005), http://ilps.science.uva.nl/Resources/blogspam

[4] Spam in blogs, Wikipedia, http://en.wikipedia.org/wiki/Spam_in_blogs

[5] Gary Robinson, A Statistical Approach to the Spam Problem (2003), http://www.linuxjournal.com/article/6467

[6] Jonathan A. Zdziarski, Ending Spam, pages 63-83 , NO STARCH PRESS,(2005)

[7]L.vonAhn,M.Blum,andJ.Langford.Telling Humans and computers apart automatically. Commun.ACM,47(2):5660(2004)

[8] Paul Graham, "A Plan for Spam, http://www.paulgraham.com/spam.html (2002)

[9] Brill, Eric, Some Advances In Rule-Based Part of Speech Tagging. In Proceedings of AAAI(1994), http://research.microsoft.com/%7Ebrill/

[10] M. Sahami, S. Dumais, D. Heckernab, and E, Horvits. A baysian approach to filtering junk E-mail. In learning for text Categorization: Papers from the 1988

Workshop, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[11] Movable Type Black Filter, with content filtering, http://www.jayallen.org/projects/mt-blacklist/

[12] Preventing comment spam using “nofollow” tag(2005), http://googleblog.blogspot.com/2005/01/preventing-comment-spam.html

[13] comment spam statistics in Project Honey Pot, http://projecthoneypot.org/

[14] comment and trackback spam statistics, http://akismet.com/stats/

[15] MIT Spam Conference (2007), http://www.spamconference.org/

[16] James Seng's MT-Bayesian, http://james.seng.cc/about/projects.html

[17] Gray Robinson, Spam Detection, http://radio.weblogs.com/0101454/stories/2002/09/24/oldSpamDetection.html

[18] Little, Rarmond C., J. Leroy Folks (1971) Asymptotic Optimality of Fisher's Method of Combining Independent Tests. Journal of the American Statistical Association, 336(66), Pp. 802-805

[19] Chang, yong seok, A Design and implementation of an improved Baysian spam filter using Chi-square statistics, Department of Computer Engineering Graduate School Keimyung University (2004)

[20] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, chapter 5, (2005)

[21] Blog, Wikipedia, http://en.wikipedia.org/wiki/Blog

[22] Michael Oakes, Robert Gaizauskas, Helene Fowkes, Anna Jonsson, Vincent Wan & Micheline Beaulieu (2001) A method based on the chi-square test for document classification. Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR 01) 440-441, New Orleans.

[23] Vapnik, Vladimir N.,The Nature of Statistical Learning Theory, Springer-Verlag, 1995.

[24] Bekkerman R., El-Yaniv R., Tkshby N., Winter Y., "On Feature Distributioanl Clustering for Text Categorization",Proceedings of SIGIR 2001, theTwenty-Fourth AnnualInternationalACM SIGIR Conference,pp. 146-153,2001

[25] Tao Li,Shenghuo Zho,MitsunoriOrkhara,"Topic Hierarchy Generation via Linear Discriminant Projection",ProceedingsofSIGIR 2003,theTwenty-Sixth AnnualInternationalACM SIGIR Conference,pp.421-422,2003

감사 드립니다.

군대를 전역하고 바로 사회에 나와 일을 시작한지 4 년이 이제 막 넘었습니다. 정말 열심히 살았다고 자부하는 그 4 년 동안 그 중심에 있었던 것은 바로 대학원 생활이었습니다. 처음 정보검색일을 하면서 검색에 무지함을 느껴 무작정 임해창 교수님에게 이력서를 보내며 조언을 구한다는 첫 메일로 인연이 시작되어 자상하게 길을 가르쳐 주셨던 3 년 전 그때를 저는 잊지 못합니다. 아마도 교수님이 직접 걸으셨던 그 당시 전화 한통이 없었다면 지금 이 시간은 없었을지 모른다는 생각이 듭니다. 그런 교수님의 관심아래 대학원 생활을 시작했고 주어진 시간에 주경야독으로 열심히 살면서 대학원 내내 장학생으로 생활할 수 있었고, 누구에게 보여줘도 부끄럽지 않을 성적과 논문 한편을 만들었습니다. 그 처음 교수님 전화하셨을 때부터 대학원 내내 관심 가져주시고 가르침 주셨던 것에 대해서 정말 이 지면으로 빌어 말하기 부족할 정도로 감사 드린다고 전해드리고 싶습니다. 교수님 정말 감사 드립니다. 그리고 논문실험시작을 어려워하던 저에게 쉽게 접근할 수 있도록 수업 과제를 통해 시작 할 수 있게 지도해 주신 육동석 교수님께도 감사의 말씀 전해 드립니다.

회사를 다니면서 학교를 다니는 것에 대해서 물심 양면으로 배려를 많이 해주신 Yahoo! Korea Search Eng.팀 우경창 차장님과, 박승 과장님 그리고 우리 Search Eng. 팀 선배님들께 감사의 말씀 전해 드립니다. 그리고 논문 쓸 때 개인적으로 많은 조언 해주셨던 Yahoo! Asia Region 의 정후중 박사님과 지금 이 시간에도 책 쓰시고 계실 이문호 박사님 모두 감사 드립니다.

또한 같은 대학교 일반 대학원에 다니면서 많은 격려와 칭찬, 조언을 아끼지 않았던 단짝 R.O.T.C 39 기 동기이자 Best Friend 인 김한식, 너도 내가 고맙겠지만 나도 네가 정말 자랑스럽고 고마웠단다. 너와 같은 날 같은 시각에 함께 졸업을 할 수 있다는 게 꿈만 같구나.

무엇보다 나 자신에게 충실하고 열심히 하는 모습에 진심 어린 박수와 격려, 사랑을 보내주신 부모님에게 감사의 말씀 드리고 싶습니다. 항상 책 읽는 모습을 보여주시면서 나에게 독서하는 소중한 습관을 길러주신 아버지, 무엇이 진정 열심히 사는 것이고 더불어 사는 삶인지 행동으로 가르쳐 주신 어머니 항상 고맙습니다. 당신의 인생이 곧 나의 인생이라는걸 한시도 잊어본 적이 없답니다. 사랑합니다.

그리고 마지막으로, 내년 2 월이면 나의 가족이 될 보현이에게 이 대학원 생활에 들인 노력과 과정, 그리고 결과의 기쁨을 함께하고 싶습니다.