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

전희원, 임해창
고려대학교 컴퓨터 정보통신 대학원
gogamza@freesearch.pe.kr, rim@nlp.korea.ac.kr

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

Hee-Won Jeon, Hae-Chang Rim
Graduate School of Computer & Information Technology, Korea University

요 약

최근 대표적인 1인 미디어의 형태인 블로그는 개인 기록의 수단뿐만 아니라 기업의 홍보에까지 널리 사용되는 인터넷 미디어이다. 그러나 누구나 글을 쓸 수 있다는 자유로움 이면에 이를 이용한 덧글 스팸이 성행이 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.

1. 서론

최근 UGC(User Generated Contents)의 유행으로 블로그 문화가 빠르게 정착되고 있다. 이러한 블로그 문화의 중요한 이점이 되는 것은 글에 대해서 방문자가 의견을 올릴 수 있는 덧글이라는 기능이 존재한다는 것이다. 덧글은 누구나 올릴 수 있는 자유로운 공간이어서 이 공간에 무분별한 스팸 덧글이 올라올 가능성이 많다. 조사 결과에 따르면 정상 덧글에 비해 스팸 덧글의 트래픽이 비교할 수 없을 정도로 크다는 사실이 연구 결과 속속들이 밝혀지고 있으며[14] 지난 2007년에 있었던 MIT Spam Conference에서 상당수의 주제가 이런 블로그 스팸에 관련된 연구였다는 것이 이 문제의 관심과 심각성을 일깨워 준다[15].

현재 스팸을 효율적으로 차단하는 여러 방법이 존재하며[4] 현재 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[15]. 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.

성능 비교를 위해 [2]에서 제공한 toy corpus[3]를 이용해 분류 평가를 함으로 제안한 방법의 유효성을 검증하였다.

2장에서 기존 연구에 대한 조사를 실시하고 3장에서는 역 카이제곱 분류 알고리즘을 간단히 소개하며, 이어지는 4장에서는 역 카이제곱 방법과 동시출현 자질을 적용한 실험 시스템에 대한 설명과 학습 데이터와 테스트 데이터에 대한 소개를 한다. 이어지는 5장에서는 실험의 결과를 토대로 분석을 하고, 6장에서 결론과 향후 연구과제를 제시하도록 하겠다.

2. 관련 연구

기존의 덧글 스팸을 방지하기 위한 여러 방법을 소개하자면 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.
- 동일한 덧글이 한꺼번에 올라오는 것을 제한한다. (“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 판정을 하기 때문이다. 이는 비교사 기반의 필터링 시스템의 한계라 생각한다.

3. 역 카이제곱 스팸 필터(Inverse Chi-Square Spam Filter)

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.

$$P(S \mid W) = \frac{P(W \mid S)}{P(W \mid S) + P(W \mid H)} \qquad (1)$$

S : Spam collection
H : Ham collection
W : word

위 식에 Robinson이 제시한 단어의 희소성에 대한 신뢰도 개념을 추가한 식은 아래와 같다.

$$f(W) = \frac{(s \times x) + (n \times P(S|W))}{s + n} \qquad (2)$$

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

이러한 단어의 집합인 문서의 확률을 구하기 위한 확률 결합식 H는 아래와 같다[18].

$$H = C^{-1}(-2\ln\prod_{W} f(W), 2n) \qquad (3)$$

C^{-1} : 카이제곱 함수의 역함수
n : 문서 내 단어의 총 개수

H 값은 정상인 문서를 찾기 위한 식이고 다음과 같이 스팸인 문서를 찾기 위한 확률 S를 계산한다.

$$S = C^{-1}(-2\ln\prod_{W} (1 - f(W)), 2n) \qquad (4)$$

마지막으로 H와 S를 결합한 새로운 확률 I를 다음과 같이 정의한다.

$$I = \frac{(1 + H - S)}{2} \qquad (5)$$

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

4. 덧글 스팸 필터 시스템

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 전처리 단계에서는 덧글과 덧글과 관련된 본문을 입력으로 받는다. 이때 덧글은 명사만 추출해서 |P(W) - 0.5|값이 가장 큰 총 5개의 중복을 1번만 허용하는 리스트를 각 덧글 확률 정보로 유지하며, 본문은 tf-idf 단어 가중치를 계산하기 위해 (6)식을 이용 본문에서 핵심어로 판단되는 단어리스트를 내림차순 정렬해서 유지한다.

$$tf - idf_{t,d} = tf_{t,d} \times \log\frac{N}{df_t} \qquad (6)$$

$tf_{t,d}$: 문서 내 특정 단어의 빈도수
N : 코퍼스 내 문서 수
df_t : 코퍼스 내에서 단어의 출현 횟수

덧글에서 확률의 영향력이 큰 단어 5개(판정 자질 명사)만 유지하는 이유는 수집한 스팸의 평균 길이가 40 char 즉 8단어 내외였고 실제로 스팸판정에 가장 영향을 많이 끼치는 단어를 포함하는 작업을 함으로서 덧글 길이의 영향을 최소화 하려고 한 것이다. 그리고 빈도수에 대한 정보를 수용하기 위해 1번 반복된 단어는 허용했다[6].
10개로 제한한 본문의 주제어를 뽑는 작업은 주제어가 덧글에 포함 유무에 대한 확률 값을 계산하기 위한 선 작업(Pre-Processing)이다.

테스트 단계에서는 덧글의 5개의 판정 자질 명사 집합에 대해서 (2)번 식을 이용한 확률을 구하고 덧글과 본문에 동시에 존재하는 전처리 단계에서 뽑은 주제어가 있을 경우 이들에 대해 아래(4.1)에서 소개될 식을 이용한 추가 확률을 구하는 것이다. 이들 단어들의 확률을 기반으로 (3), (4), (5)번 식을 사용해 문서의 스팸 유무를 판별한다.
시스템의 전체적인 구조는 아래 그림1과 같다.

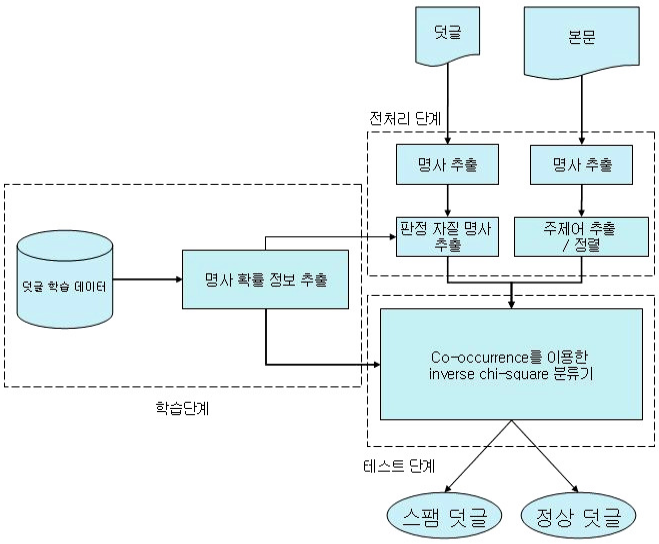

그림 1

4.1 동시출현 단어 자질 정보

덧글과 본문에 동시 존재하는 주제어에 대한 확률 값을 알아야 하는데 이는 (1)번 식을 이용하면 유도할 수 있다.

(1)식을 다시 쓰면 아래의 (7)번식과 같이 쓸 수 있는데.

$$P(S \mid W) = \frac{P(W \mid S) \bullet P(S)}{P(W \mid S) \bullet P(S) + P(W \mid H) \bullet P(H)} \quad (7)$$

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

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

$$d.b = \frac{t.f}{dl} \quad (8)$$

dl : 모든 주제어의 총 빈도수
t.f : 동시출현 주제어 빈도수

이렇게 구현된 동시출현 주제어들에 대한 스팸 확률은 아래와 같다.

$$P(S \mid C.O.W) = \frac{P(C.O.W \mid S) \bullet (1 - db)}{P(C.O.W \mid S) \bullet (1 - db) + P(C.O.W \mid H) \bullet db} \quad (9)$$

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 논문에서는 아래와 같이 정의한다.

P(C.O.W|H) = 0.1 (10)
P(C.O.W|S) = (1- P(C.O.W|S)) (11)

이러한 (8), (9)번 식을 사용해 도출된 동시출현 자질 확률들을 기반으로 (5)번식을 사용 최종적인 덧글 스팸 필터링을 하게 된다.

판정에 대한 기준은 아래와 같다.

I > 0.55 : 스팸
I < 0.45 : 정상 (10)
0.45 <= I <= 0.55 : 판정 불능(Gray Area)

5. 실험 및 결과

5.1 실험 환경

본 논문에서 제안한 실험 시스템은 아래 표1과 같은 환경에서 3가지 필터를 직접 모두 구현하여 실험하였다.

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

표1. 실험 환경

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.com에서 IT분야의 영문 블로그 30개와 정치, 경제 분야의 영문 블로그 30개를 무작위 선택해 직접 구현한 웹 크롤러를 사용하여 덧글 데이터를 수집, 분류했다.

이렇게 수집한 학습 덧글과 테스트 덧글에 대한 정보는 아래와 표2과 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 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.

성능 평가 방법으로는 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

hm = c / (a + c) (12)

sm = d / (b + d) (13)

$$lam = \log it^{-1}(\log it(hm)/2 + \log it(sm)/2) \quad (14)$$
$$where: \log it(x) = \log(x/(1-x))$$
$$\log it^{-1}(x) = e^x/(1+e^x)$$

error = (c + d) / (e + f) (15)

accurate = (a + b) / (e + f) (16)

recall = b / e (17)

precision = b / (b + c) (18)

$$f_1 - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (19)$$

5.2 실험 결과

표3는 테스트 셋에 대한 실험 결과를 나타낸다.

평가방법(%)	베이지언	역 카이제곱	동시출현 자질 + 역 카이제곱
Hm	8.51	7.57	5.26
Sm	50.33	40.70	34.67
Lam	23.49	19.17	14.65
Error	35.71	26.04	22.95
Recall	49.67	52.61	61.27
Precision	91.57	93.06	95.66
F_1-measure	64.41	67.22	74.70
Accurate	64.29	65.37	72.37

표3. 비교 실험 결과

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.

5.3 결과 분석

1) 동시출현 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 확률을 평가하는데 중요한 요소로 쓰일 수 있다는 것을 보여준다.

2) 오류율(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 0.4에 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.

3) Grey area에 대한 고찰

제안한 방법으로 전체적인 성능향상은 있었으나 필터에서 grey area로 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 area에 대한 다른 고민들이 필요할거라 생각한다.
실험 결과 로그를 분석하면 “Nice Site!”라는 덧글이 grey area로 판단이 되었고 그 덧글에 포함된 링크는 포르노(pron) 사이트로 연결이 되어 있었음이 확인 되었

다. 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) 주제어를 포함한 스팸 덧글이라면?

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.idf를 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.

6. 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] Tom M, Mitchell, Machine Learning, McGraw-Hill International Edition, chapter 6, (1997)
[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] Gary 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