한글 전자메일에 대한 베이지언 필터의 성능비교

이창범° 김지수 김수형 박혁로
전남대학교 전자컴퓨터정보통신공학부
chblee@empal.com, kimjisoo@iip.chonnam.ac.kr, {shkim, hyukro}@chonnam.ac.kr

Comparison of Performance for Korean E-mail Filtering using Bayesian Classifier

Chang-Beom Lee° Ji-Soo Kim Soo-Hyung Kim Hyuk-Ro Park
Dept. of Computer Science, Chonnam National University

요 약

전자 메일은 매우 많은 사람들이 사용하는 편리하고 효율적인 통신 수단이다. 그러나 전자메일 주소를 쉽게 획득할 수 있다면 점을 악용하기 때문에 사용자가 원하지 않는 메일 즉 스팸 메일에 대한 문제가 심각해지고 있다. 이러한 스팸 메일을 자동으로 분류해주는 스팸 필터는 주로 영어를 대상으로 하고 있으며, 규칙 기반 필터링보다는 통계적 학습을 통한 필터링 방법을 주로 사용하고 있다.
본 논문에서는 베이즈 정리를 기반으로 하는 3가지 분류 알고리즘을 한글 전자메일을 대상으로 하여 스팸 메일 특히 음란성 메일을 분류하는데 있어 그 성능을 평가하고자 한다. 실험 결과, 단어의 스팸일 확률만을 이용하는 방법이 나이브 베이즈 알고리즘이나 m-estimate를 이용하는 방법보다는 성능이 우수함을 알 수 있었다. 특히, 단어의 스팸일 확률만을 이용하는 방법은 false positive rate를 0%로 유지하면서도 다른 방법들보다는 필터링을 잘 해내고 있음을 확인할 수 있었다. 그리고, 자질 선정에서는 명사나 명사/형용사를 사용할 경우에 그 에러율이 가장 적었다.

1. 서 론

전자 메일은 매우 많은 사람들이 사용하는 편리하고 효율적인 통신 수단이다. 그러나 전자 메일 주소만 알면 메일을 보낼 수 있다는 점과, 대부분 사용자가 전자메일 주소를 인터넷으로부터 쉽게 구할 수 있다는 점은 전자 메일을 상품이나 기업 선전에 활용하게 만들었다.

전체 메일 중 사용자가 원하지 않는 상품 선전 메일이나 음란성 메일인 스팸 메일의 숫자는 계속 증가하고 있다. 이러한 스팸 메일은 수신자로 하여금 스팸 메일을 확인하기 위해 많은 시간을 소비하게 만든다. 만약, 메일 사용자가 모뎀이나 무선 인터넷을 사용하는 경우 스팸 메일을 처리하기 위해 많은 비용이 들 수도 있어서 더욱 심각한 문제가 되고 있다.

스팸 메일에 대처하는 방법 중 하나는 메일 필터 프로그램이 자동적으로 스팸 메일을 제거하는 것이다. 이렇게 함으로써 메일 사용자의 시간과 비용을 절감하게 하는 프로그램을 스팸 필터라 부른다.

스팸 필터 프로그램은 MicroSoft의 OutlookExpress에 내장된 필터 등 여러 종류의 필터가 나와 있으며, 이들 필터를 크게 구분하면 규칙 기반 필터와 통계적학습을 통한 필터로 나눌 수 있다. 스팸 제작자들이 필터링 규칙을 피해서 새로운 형태의 스팸을 만들 수 있으며 또한 여러 개의 다른 계정을 이용해 스팸 메일을 보내기 때문에 통계적 학습을 통한 필터링이 규칙 기반 필터링보다 더 유용하다고 알려져 있다. 통계적 학습을 통한 필터링 방법 중 가장 널리 사용되는 방법은 나이브 베이지언 분류자(Naïve Bayesian Classifier)를 이용하는 방법이다. 베이지언 분류자를 이용하는 통계적 방법을, 이후에서 베이지언 필터라 부르도록 하겠다.

베이지언 필터는 어떤 특정한 자질(feature)이 각각 스팸 메일과 일반 메일에 나타날 확률을 이용하여 스팸 여부를 판단한다. 이러한 확률은 스팸 메일과 일반 메일 샘플로부터 학습할 수 있지만, 초기에 어떤 자질을 사용하느냐에 따라서 그 성능이 매우 달라질 수 있다.

그리고, 대부분의 스팸 필터는 영어를 대상으로 하고 있기 때문에 영어 단어를 자질로 사용하고 있다. 따라서 이러한 필터를 국내 메일에 적용하는 것은 불가능하다. 결국, 국내 메일에 맞는 스팸 판별 자질에 대한 연구와 스팸 필터에 대한 연구가 필요하다.

본 논문에서는 영어를 대상으로 한 스팸 필터[1-3]를 한글 전자메일에 적용하여 그 성능을 비교하였고, 자질 추출 문제에 대해서는 명사, 동사, 형용사의 모든 경우와 형태소 분석을 하지 않고 어절 자체를 자질로 사용할 경우를 비교하였다.

본 논문의 구성은 다음과 같다. 2장에서는 베이즈 정리와 나이브 베이지안 분류자에 대해 언급하고, 3장에서는 성능 평가한 세 가지 알고리즘에 대해 기술한다. 그리고, 4장에서는 실험 및 평가를, 5장에서는 결론 및 향후 연구에 대해서 기술한다.

2. 관련 연구

이번 장에서는 베이즈 정리와 나이브 베이지언 분류자에 대해 설명한다[1,4].

2.1 베이즈 정리(Bayes theorem)

가설 공간 h와 주어진 훈련 데이터 D로부터 가장 좋은 가설을 구하고자 할 때 식(2.1)과 같은 베이즈 정리를 사용할 수 있다. $P(h)$는 h의 사전확률(prior probability)을 나타내고, $P(h|D)$는 D가 주어졌을 때 h의 사후확률(posterior probability)이다.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (2.1)$$

사후확률이 최대인 가설(MAP : Maximum a posterior)을 찾기 위해 베이즈 정리를 사용하면 식(2.2)와 같다.

$$\begin{aligned} h_{MAP} &\equiv \underset{h\in H}{argmax}\, P(h|D) \\ &\equiv \underset{h\in H}{argmax}\, \frac{P(D|h)P(h)}{P(D)} \\ &\equiv \underset{h\in H}{argmax}\, P(D|h)P(h) \end{aligned} \quad (2.2)$$

이때, $P(D)$는 식(2.2)의 마지막 단계에서 생략된다. 이는 $P(D)$는 h에 독립적인 상수이기 때문이다.

2.2 나이브 베이지언 분류자(Naïve Bayesian Classifier)

나이브 베이지언 분류자는 베이지언 학습 방법 중에서 널리 쓰이는 통계적 학습 방법이다. 나이브 베이지언 분류자는 속성값들의 결합으로 이루어진 각 인스턴스(instance) x와 특정 유한 집합 V에서 어떤 값을 갖는 목적함수 $f(x)$가 존재하는 학습 단계에 적용된다.

새로운 인스턴스를 분류하기 위한 베이지언 접근법은 인스턴스를 구성하는 속성값들 $\langle a_1, a_2, \cdots, a_n \rangle$로부터 구해지는 가장 큰 확률값을 v_{MAP}에 대입하는 방식이다. 아래의 식(2.3)은 이를 나타내고 있다.

$$v_{MAP} = \underset{v_j\in V}{argmax}\, P(v_j|a_1, a_2, \cdots, a_n) \quad (2.3)$$

식(2.3)을 베이즈 정리를 이용하여 다시 쓰면

$$\begin{aligned} v_{MAP} &= \underset{v_j\in V}{argmax}\, \frac{P(a_1, a_2, \cdots, a_n|v_j)P(v_j)}{P(a_1, a_2, \cdots, a_n)} \\ &= \underset{v_j\in V}{argmax}\, P(a_1, a_2, \cdots, a_n|v_j)P(v_j) \end{aligned} \quad (2.4)$$

와 같다.

이제, 식(2.3)을 이용하는 대신에 식(2.4)을 이용하여 v_{MAP}을 구해낼 수 있다. 여기서, $P(v_j)$는 훈련 데이터에서 v_j 가 출현하는 횟수를 이용하여 쉽게 구할 수 있지만, $P(a_1, a_2, \cdots, a_n|v_j)$는 매우 많은 훈련 데이터 집합을 갖고 있지 않는 한 구하기가 쉽지 않다. 하지만, 나이브 베이지언 분류자는 속성값들이 주어진 주어진 목적값에 조건부 독립적(conditionally independent)이라는 가정을 기반으로 하기 때문에 $P(a_1, a_2, \cdots, a_n|v_j) = \prod_i P(a_i|v_j)$와 같다. 결국, 식(2.4)는 식(2.5)와 같이 다시 쓸 수 있으며, 이 식이 나이브 베이지언 분류자가 된다.

$$v_{NB} = \underset{v_j\in V}{argmax}\, P(v_j)\prod_i P(a_i|v_j) \quad (2.5)$$

여기서, v_{NB}는 나이브 베이지언 분류자가 출력하는 목적값을 나타낸다.

3. 성능 평가한 세 가지 알고리즘

이번 장에서는 한글 전자메일에 대해 성능 평가한 세 가지 알고리즘에 대해 기술한다. 첫 번째는 나이브 베이즈 알고리즘[1]이며, 두 번째는 M-Estimate를 이용하는 방법[2]이다. 그리고, 마지막은 나이브 베이지언 분류자의 변형이라고 볼 수 있는, 단어의 스팸일 확률만을 이용하는 [3]에서 제안한 알고리즘이다.

3.1 나이브 베이즈 알고리즘

본 연구에서는 받은 메일이 스팸메일인지 아닌지에만 관심이 있으므로 목적값의 집합인 V는 v_{spam}과 $v_{nonspam}$로만 이루어진다. *Vocabulary*는 *Examples*로부터 추출한 중복 없는 단어들의 집합이며, 단어는 실험실에서 보유하고 있는 형태소분석기를 실행하여 내용어(보통명사, 고유명사, 동사, 형용사)을 추출하였다. 텍스트 학습 및 분류를 위한 나이브 베이즈 알고리즘은

〈표 3.1〉과 같다.

[표 3.1] 텍스트 학습 및 분류를 위한 나이브 베이즈 알고리즘

Learn_Naive_Bayes(Examples, V)
*Examples*은 목적값을 갖는 문서들의 집합
*V*는 가능한 목적값들의 집합

1. *Examples*에서 발생한 모든 단어 및 토큰들을 모은다.
   · *Vocabulary*는 *Examples*에서 발생한 중복없는 각 단어 및 토큰의 집합

2. $P(v_j)$와 $P(w_k|v_j)$를 계산한다.
   · $Docs_j \leftarrow$ *Examples*에서 목적값이 v_j인 문서들의 집합
   · $P(v_j) \leftarrow \frac{|Docs_j|}{|Examples|}$
   · $Text_j \leftarrow Docs_j$의 모든 멤버들을 더해서 만든 하나의 문서
   · $n \leftarrow Text_j$ 안의 단어 수
   · *Vocabulary* 안의 각 단어 w_k에 대해서 반복
      · $n_k \leftarrow Text_j$안에서 단어 w_k가 나온 단어수
      · $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Classify_Naive_Bayes(Doc)
문서 *Doc*에 대해 평가된 목적값을 돌려준다.
a_i는 문서 *Doc*안의 i번째 위치에서 발견된 단어를 나타낸다.
· *positions* $\leftarrow$ *Vocabulary*에 포함된 단어들의 *Doc*안의 위치
· $v_{NB} = \underset{v_j \in V}{argmax}\ P(v_j) \prod_{i \in positions} P(a_i|v_j)$

위 알고리즘에서 *Learn_Naive_Bayes(Examples, V)*에 의해 학습을 한 후, 아래의 식(3.1)를 만족하는 단어들의 확률 즉, $P(w_k|v_j)$를 제거한다[2]. 이는 스팸인지 논스팸인지 분류하는 데 있어 노이즈 데이터를 제거하는 효과라 볼 수 있다.

· $N(W, Spam) + N(W, NonSpam) < 4$; or

· $\frac{P(W|Spam)}{P(W|Spam)+P(W|NonSpam)} \in [0.45, 0.55]$ (3.1)

즉, 어떤 단어의 총 출현 회수(스팸+논스팸)가 4미만인 단어나, 식(3.1)의 아래와 같이 계산된 확률이 0.45와 0.55사이인 단어들의 학습 정보를 제거한다.

그리고, 새로운 전자메일이 스팸인지 논스팸인지를 판단하기 위해서는 *Classify_Naive_Bayes(Doc)*을 이용한다. 이때, 분류를 하기 위해 어떤 임계치를 사용하는 대신에, 확률 값이 큰 쪽으로 분류를 한다.

3.2 M-Estimate를 이용하는 알고리즘

*m−estimate*을 이용하는 알고리즘은 〈표 3.1〉과 같은 방법으로 학습과 분류를 한다. 다만, $P(w_k|v_j)$의 값을 구하기 위해 *m−estimate*방법을 사용한다. *m−etsimate*는 표본 데이터와 가상의 m 균등 분포의 데이터를 혼합한 것으로 간주된다[1,2]. 확률의 *m−estimate*는 식(3.2)과 같이 정의된다. 그런데, [2]에서는 식(3.3)과 같이 $m=1$로, 그리고 $p = \frac{1}{\lambda}$로 사용하고 있다. 여기서, λ는 학습 데이터의 중복 없는 총 단어의 수를 의미하며, 〈표 3.1〉의 $|Vocabulary|$와 동일하다.

m−estimate of probability :

$$\frac{n_k+mp}{n+m} \qquad (3.2)$$

$$P(w_k|v_j) = \frac{n_k+\frac{1}{\lambda}}{n+1} \qquad (3.3)$$

그리고, 노이즈를 제거하기 위해 식(3.1)과 같은 조건을 사용한다.

3.3 단어의 스팸일 확률만을 계산하는 알고리즘

[3]에서 제안하는 이 방법은 어떤 단어에 대해 스팸일 확률과 논스팸일 확률을 이용하는 것이 아니라, 단어가 스팸일 확률만을 이용하여 새로운 전자메일에 대해 스팸인지 논스팸인지를 판단한다.

어떤 단어의 스팸일 확률을 계산하기 위해 먼저, 스팸과 논스팸 코퍼스에서 발생하는 단어들의 출현 횟수를 이용하여 해쉬 테이블(hash table)을 각각 생성한다. 그런 후에, 다음의 〈표 3.2〉와 같이 각 단어의 스팸일 확률을 계산하여 세 번째 해쉬 테이블을 생성한다.

[표 3.2] 단어의 스팸일 확률을 계산하는 알고리즘

```
(let ((g (* 2 (or (gethash word good) 0 )))
      (b (or (gethash word bad) 0)))
  (unless (< (+ g b) 5)
    (max .01
         (min .99 (float (/ (min 1 (/ b nbad))
                            (+ (min 1 (/ g ngood))
                               (min 1 (/ b nbad))))))))
```

〈표 3.2〉에서, word는 확률을 계산할 단어를 나타내며, good과 bad는 논스팸과 스팸에 대한 해쉬테이블이다. 그리고, ngood과 nbad는 논스팸과 스팸 전자메일의 개수이다.

good에서 발견된 모든 단어의 출현 회수에 2배를 하고 있으며, 이는 논스팸을 스팸으로 판단하는 오류(false positive)를 피하기 위해서이다. 그리고, 출현 회수가 5번 이상인 단어(실제적으로는 논스팸에서 3번이상 발생한 단어면 충분하다)에 대해서만 확률을 계산하고 있다. 또한, 한 쪽 코퍼스(논스팸 또는 스팸)에만 나타나는 단어의 확률은 0.01과 0.99로 선택된다.

이제, 새롭게 도착한 전자메일에 대해 스팸 여부를 판단하기 위해 그 전자메일에 포함된 가장 관심있는(단어의 확률이 0.5와의 거리가 멀수록 관심있다) 15개의 단어를 선택한다. 이때, 학습과정에서 출현하지 않는 단어의 확률은 0.4로 하고 있다. 이렇게 선택된 15개의 단어들의 결합확률(combined probability)을 계산하여 스팸 여부를 판단한다. 결합확률을 계산하는 방법은 〈표 3.3〉과 같으며, 이 확률이 0.9이상이면 스팸 메일로 간주한다.

[표 3.3] 결합확률 계산 방법

a, b가 두 독립 사상에 대한 확률이라 한다면, 결합확률은 다음과 같이 계산된다. $\frac{ab}{ab+(1-a)(1-b)}$

〈표 3.3〉은 두 단어(a,b)에 대해서만 보이고 있지만, 실제 결합확률을 계산할 때는 같은 방법으로 15까지 확장한다.

4. 실험 및 평가

4.1 실험 자료

논스팸 메일 72개, 스팸메일 81개를 학습에 사용하였다. 그리고, 논스팸 메일과 스팸 메일 각각 20개를 테스트에 사용하였다. 특히, 스팸 메일은 그 대상을 광고성 메일을 제외하고 음란성 메일만을 대상으로 하였다. 또한, 학습에 사용된 자질(feature)는 한글 단어만을 대상으로 하였고, 단어는 실험실에서 보유하고 있는 형태소 분석기를 이용하여 명사, 동사, 형용사 등의 내용어를 추출하여, 이들 품사의 모든 조합에 대해서 자질로 선정하여 상호 비교하였다. 또한, 형태소 분석을 전혀 시행하지 않고 어절(영문자, 숫자, 기호는 제외) 자체를 자질로 선정하여 분류하는 실험도 하였다.

4.2 평가 측도

평가 측도[2]는 false positive rate R_{fp}, false negative rate R_{fn}, 그리고 error rate R_e를 사용하였다. 〈표 4.1〉은 이러한 평가 측도를 계산하는 방법을 보이고 있다.

[표 4.1] 평가 측도

$$R_{fp} = 1 - \frac{CorCount(NonSpam)}{TrueCount(NonSpam)}$$

$$R_{fn} = 1 - \frac{CorCount(Spam)}{TrueCount(Spam)}$$

$$R_e = 1 - \frac{CorCount(Spam) + CorCount(NonSpam)}{TrueCount(Spam) + TrueCount(NonSpam)}$$

$TrueCount(Spam)$과 $TrueCount(NonSpam)$은 각각 테스트에 참여한 스팸과 논스팸 메일의 개수이다. 그리고, $CorCount(Spam)$과 $CorCount(NonSpam)$은 정확하게 분류된 스팸과 논스팸 메일의 개수를 나타낸다.

4.3 실험 결과

자질은 형태소 분석을 하지 않는 경우와 명사, 동사, 형용사의 조합 7가지를 이용하여 선정하였다. 〈표 4.2〉, 〈표 4.3〉, 그리고 〈표 4.4〉는 자질 선정에 따른 나이브 베이즈 알고리즘(naive), m-estimate를 이용한 알고리즘(m-etimate), 그리고 스팸일 확률만을 이용하는 방법(graham)의 성능을 나타내고 있다. 각 표에서 "명"은 명사를, "동"은 동사를, 그리고 "형"은 형용사를 의미한다.

세 가지 방법 모두에서 명사 단독으로 선택하거나 명사와 형용사를 같이 이용하는 경우의 성능이 가장 우수함을 보이고 있지만, 결국 자질로써 명사만 선택하더라도 충분하다는 의미로 생각할 수 있다.

〈표 4.2〉 naive 방법에 대한 자질 선정 실험

	분석 안함	명	동	형	명동	명형	형동	명동형
R_{fp}	75%	55%	65%	85%	55%	55%	65%	55%
R_{fn}	40%	35%	65%	100%	50%	35%	65%	50%
R_e	58%	45%	65%	93%	53%	45%	65%	53%

[표 4.3]〉 m-estimate 방법에 대한 자질 선정 실험

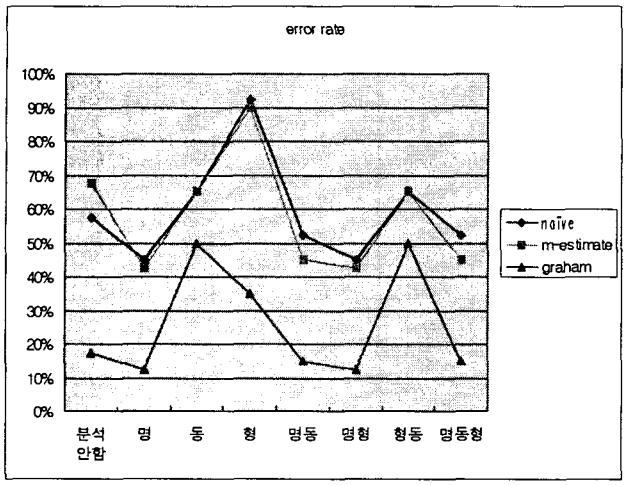
	분석 안함	명	동	형	명동	명형	형동	명동형
R_{fp}	100%	55%	65%	80%	55%	55%	65%	55%
R_{fn}	35%	30%	65%	100%	35%	30%	65%	35%
R_e	68%	43%	65%	90%	45%	43%	65%	45%

[표 4.4] graham 방법에 대한 자질 선정 실험

	분석 안함	명	동	형	명동	명형	형동	명동형
R_{fp}	0%	0%	0%	0%	0%	0%	0%	0%
R_{fn}	35%	25%	100%	70%	30%	25%	100%	30%
R_e	18%	13%	50%	35%	15%	13%	50%	15%

(그림 4.1)은 세 가지 방법에 대한 각 자질별 error rate를 보이고 있다. 모든 경우에 스팸일 확률만을 이용하는 방법(graham)이 분류 정확도가 가장 높고, false

positive가 모든 경우에 0%이다.



[그림 4.1] 세 가지 알고리즘에 대한 error-rate

기존 연구[2,3,4]에서는 분류 정확도가 모두 90%이상으로 기록하고 있으며, 이들 연구는 광고성 메일을 포함한 스팸 메일을 대상으로 하고 있다. 하지만, 본 논문에서는 광고성 메일을 제외한 음란성 메일만을 대상으로 하고 있으며, 음란성 메일에는 학습을 시킬 단어를 추출하는데 어려움이 있다. 추출된 단어는 대부분 메일의 제목에서 발생하고 있고, 본문에는 이미지나 URL로 처리를 하고 있기 때문이다. 본 논문의 분류 정확도가 전체적으로 떨어지는 이유가 여기에서 기인된다고 볼 수 있다.

또한, 스팸과 논스팸 메일에 대해 같은 비중을 두고 스팸과 논스팸 확률을 각각 계산하는 naive 방법이나 m-estimate 방법보다 스팸 메일에 더 비중을 둔다고 볼 수 있는 graham 방법이 더 좋은 성능을 보이는 이유도 학습 데이터의 부족이라고 생각된다. 하지만, (그림 4.1)에서 볼 수 있듯이, 학습 데이터가 부족한 상황에서도 graham 방법이 최고 87%의 분류 정확도를 나타내고 있으며, 이러한 상황에서는 다른 두 가지 방법보다는 graham 방법이 더 적당하다고 할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 주로 영어를 대상으로 하는 세 가지 베이즈 정리를 이용하는 필터 즉, 나이브 베이즈 알고리즘을 이용하는 방법[1], $m-estimate$를 적용한 방법[2], 그리고 단어의 스팸을 확률만을 계산하여 분류하는 방법[3]을 한글 전자메일에 적용하여 그 성능을 비교 평가하였다. 스팸메일은 광고성 메일을 제외한 음란성 메일만을 대상으로 하였고, 자질 추출시에서는 한글 단어만을 고려하였다.

실험 결과, 단어의 스팸일 확률만을 이용하는 [3]의 방법이 다른 두 가지 방법보다 그 성능이 우수함을 확인할 수 있었다. 특히, [3]의 방법은 false positive rate를 0%로 유지하면서, 최고 87%의 스팸 분류 정확성을 보이고 있다. 그리고, 자질 선정 문제는 명사만을 추출하더라도 충분히 해결될 수 있다는 점을 확인하였다.

본 논문에서는 베이지언 필터 간의 성능을 비교하였으나, 다른 분류기(SVM, K-NN, NN 등)의 한글 전자메일 분류에 대한 연구 또한 필요할 것으로 보인다. 또한, 메일의 본문에 주로 나타나는 이미지를 처리하여 이미지 내의 한글 텍스트를 포함한다면 성능이 더 우수할 것으로 기대된다.

6. 감사의 글

본 논문은 한국전자통신연구원 개인정보보호연구팀에서 연구중인 내용기반 유해정보방지기술개발의 위탁 연구 과제 "고성능 베이지언 필터링 기술 연구" 로 수행한 결과입니다.

참고 문헌

[1] Tom M. Mitchell, "Machine Learning", McGraw-Hill, 1997.

[2] P. Pantel, D. Lin, "SpamCop : A Spam Classification & Organization Program," In Learning for Text Categorization : Papers from the AAAI Workshop, pp.95-98, 1998 (AAAI Technical Report WS-98-05).

[3] http : //www.paulgraham.com/spam.html

[4] 조한철, 조근식, "나이브 베이지안 분류자와 메시지 규칙을 이용한 스팸메일 필터링 시스템," 2002년도 정보과학회 춘계학술대회, pp. 223-225, 2002.

[5] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, C. D. Spyropoulos, "An evaluation of Naive Bayesian anti-spam filtering," Proc. workshop on Machine Learning in the New Information Age, G. Potamias, V. Moustakis and M. van Someren (eds.), 11th European Conference on Machine Learning, Barcelona, Spain, pp. 9-17, 2000.

[6] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, C. D. Spyropoulos, "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages," Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, N.J. Belkin, P. Ingwersen and M.-K. Leong (Eds.), Athens,

Greece, pp. 160-167, July 24-28, 2000.

[7] K. M. Schneider, "A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering," Proc. 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003), Budapest, Hungary, pp. 307-314, Apr. 2003.

[8] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," In Learning for Text Categorization : Papers from the AAAI Workshop, Madison Wisconsin, Madison Wisconsin, pp. 55-62, 1998 (AAAI Technical Report WS-98-05).

[9] http : //www.paulgraham.com/better.html