Topic Signature를 이용한 댓글 분류 시스템

배민영○ 차정원
창원대학교 컴퓨터공학과
nikismy@changwon.ac.kr, jcha@changwon.ac.kr

Comments Classification System using Topic Signature

Min-Young Bae○ Jeong-won Cha
Dept. of Computer Engineering Changwon National University

1. 서론

토론과 비판의 장이 되었던 인터넷 공간이 익명성을 악용한 범죄의 공간으로 변하고 있다. 지난 2007년 개최 된 MIT Spam Conference의 주제 중 상당수가 스팸 관련 연구였다는 점에서 악성댓글의 관심과 심각성을 일깨워 준다[1]. 현재 악성댓글을 방지하기 위해 연구 된 방법들의 대부분은 품사 태거 혹은 명사추출기 등을 이용한 연구이다. 그러나 악성댓글의 경우 정상적인 댓글에 비해 길이가 매우 짧고, 대부분 띄어쓰기나 정확한 단어의 사용이 이루어지지 않는다는 특징이 있어 기존의 방식으로는 높은 성능을 기대하기 어렵다. 또한, 분류 과정에서 신조어나 변형어가 많아 주요어(Keyword)를 기반으로 분류 할 경우 오류 발생 확률이 높아져 분류의 정확성이 떨어지게 된다.

2. 본론

악성댓글은 일반댓글과 달리 문장의 길이기 짧고 띄어쓰기가 거의 없으며 비형식적인 특징을 가진다. 따라서 기존에 제안된 방법으로 자질을 추출하는데 있어 오류 발생의 확률이 높아질 수 있다. 본 논문에서는 인터넷으로부터 수집된 댓글은 XML형식으로 만들며, 일반댓글의 경우 <title>과 <body> 부분, 악성댓글의 경우 사용자에 의해 수집된 특정 구간(<block>)의 내용을 학습하여 악성댓글 여부를 판별하는 시스템을 구축하였다. 시스템은 크게 학습 단계와 테스트 단계로 이루어진다.

학습 단계에서는 댓글을 수집하여 짧은 문장에서 너무 많은 주변 정보를 이용함에 따른 오류 발생의 문제를 최소화하기 위하여 N-gram으로 분리한다. 본 논문에서는 7-gram을 사용한다. 그 이유는 악성댓글 구간 추출 과정에서 구간의 평균 문자수를 계산해 본 결과 7에 근접하는 값을 가졌기 때문이다. 이 7-gram을 다시 tri-gram 단위로 나누어서 학습을 한다. 본 논문에서는 이 tri-gram을 단어라 하고, 한 단어를 자질이라고 한다.

신생 단어가 생겨나고 악성댓글의 문장 길이가 길지 않은 악성댓글의 특징에 따라 학습 데이터베이스를 구축하는데 있어 많은 어려움이 존재한다. 일반댓글만을 학습할 경우 악성댓글보다 가능한 예들이 많으므로 현실에서 나타날 수 있는 모든 문장을 학습시키는 것은 불가능 하다. 따라서 악성댓글과 일반댓글 모두를 학습한 후 자질을 추출하여 판별에 적용하는 토픽 시그너처를 이용한다. [표 1]의 테이블에서 토픽 시그너처는 식(1)과 같다. [표 1]에서 t는 단어를 의미하며, $TS_s(t)$는 단어 t가 SPAM에 속할 때의 토픽 시그너처 값이다.

표 1 토픽 시그너처의 Contingency 테이블

	SPAM	NON-SPAM
t	v_{11}	v_{12}
~t	v_{21}	v_{22}

$$TS_s(t) = 2 \times (v_{11}+v_{12}+v_{21}+v_{22}) \times \left(\frac{v_{11}}{(v_{11}+v_{21}) \times (v_{11}+v_{12})} \right) \quad (1)$$

테스트 단계에서는 학습된 이터베이스를 이용하여 각 댓글의 카테고리를 결정한다. 각 댓글에서 자질을 추출하고 댓글에 나타난 자질에 대한 확률값을 학습 데이터베이스에서 찾는다.

$$P(C|D) = \left(\frac{P(C)PC(D|C)}{P(D)} \right) \quad (2) \qquad P(D|C) = \frac{\prod_{i=1}^{AC} TS_{c,i}(t)}{AC} \quad (3)$$

식 (2)와 같이 베이시안(Naive Bayes) 모델을 사용하여 문서에 대한 카테고리의 확률을 계산한다. 이때, 확률을

문서에서 나타난 총 자질의 수만큼 나누어 문서의 길이에 따라 분류에 영향을 미치는 것을 방지한다. 식 (3)에서 $P(D|C)$는 카테고리에서 문서가 나타날 확률이고, C는 카테고리를 나타낸다. AC는 하나의 댓글에서 나타나는 총 자질의 수이다.

이 실험을 위한 데이터는 YAHOO Korea (http://kr.yahoo.com/)의 정치 뉴스 분야의 기사를 무작위 선택, 댓글을 수집하였으며, 악성댓글의 특정 구간은 사람이 직접 선택하였다. 시스템 평가를 위한 문서 집합은 [표 4]와 같다. 성능 평가의 방법으로는 분류 시스템의 성능 평가를 위해 주로 사용되는 Precision, Recall, F_1-measure를 이용하였다. 구하는 공식은 식 (4)-(6) 와 같다.

System \ Gold	SPAM	NON-SPAM
SPAM	A	B
NON-SPAM	C	D

$$P(\mathrm{Precision}) : \frac{A}{A+B} \quad (4)$$

$$R(Recall) : \frac{A}{A+C} \quad (5)$$

$$F_1(F_1-measure) : \frac{2\times Recall\times \mathrm{Precision}}{Recall+\mathrm{Precision}} \quad (6)$$

[표 2]은 토픽 시그너처, 카이 제곱 통계량에 대한 각각의 실험 결과이다. 실험 A는 문장을 tri-gram으로만 나눈 방법이고, 실험 B는 문장을 N-gram으로 나누고 N-gram을 다시 tri-gram으로 나눈 방법이다. [표 3]은 실험 B의 방법에서 (1) 불용어[1]를 제거하지 않은 경우와 (2) 악성댓글 전체를 학습한 경우에 대한 실험 결과이다.

표 2 토픽 시그너처, 카이스퀘어를 이용한 성능 비교

Model / 평가(%)	Topic Signature		Chi-square	
	실험 A	실험 B	실험 A	실험 B
P	0.7125	0.7724	0.7972	0.7288
R	0.8769	0.9923	0.8769	0.9923
F_1	0.7862	**0.8686**	0.8351	0.8403

표 3 실험 B 방법을 이용한 각 모델의 성능 비교

Model / 평가(%)	(1)		(2)	
	TS	Ch-S	TS	Ch-S
P	0.7530	0.7815	0.6993	0.6709
R	0.9615	0.7153	1	1
F_1	**0.8445**	0.7469	0.8230	0.8030

[표 2]와 [표 3]을 통해 모든 모델이 74% 이상의 성능을 보임을 확인할 수 있다. 또한 토픽 시그너처를 이용한 실험에서 출현빈도에 따른 성능을 측정한 결과 출현빈도가 1일 때 더 좋은 성능을 보였다.

3. 결론

날로 증가하는 악성댓글은 이제 개인만의 문제를 넘어 사회 전반의 문제로 대두되었다. 이러한 악성댓글의 해결을 위해 다양한 연구가 진행되고 있으나 일반댓글과는 다른 악성댓글의 특징에 따라 자동으로 악성댓글을 판별해 줄 수 있는 시스템의 개발이 쉽게 이루어지지 못하고 있는 실정이다.

본 논문에서는 악성댓글의 특징을 이용하여 단순한 패턴 매칭 방법을 이용한 방법이 악성댓글의 분류 성능을 개선할 수 있다는 것을 보였다. 정형화되지 않은 악성댓글의 다양한 패턴 학습을 통하여 기존 연구에 적용 된 선행 작업들(품사부착, 특정 품사추출, 등) 없이도 전체적인 시스템의 성능 향상이 가능함을 실험 결과로 보여준다. 또한 제안된 방법은 간단한 방법으로 이루어졌다. 다양한 형태의 악성댓글의 학습을 통하여 은유나 비유적인 표현으로 작성된 댓글도 분류해 낼 수 있었다.

본 논문에서는 댓글의 각 문장을 모두 N-gram으로 나눈 후 2차적으로 tri-gram으로 나누어 tri-gram의 출현 빈도와 확률을 계산하는 방식을 이용하였다. 그러나 대부분의 악성댓글의 경우 짧은 문장 길이에도 불구하고 특정 부분에 악성댓글임을 암시하는 단어나 문장이 존재했다. 만약 모든 문장을 N-gram으로 나누지 않고, 악성댓글의 특정 구간을 판별해 낼 수 있다면 더 빠른 속도로 악성댓글을 분류할 수 있는 시스템이 구현될 수 있을 것이라 생각된다.

참고문헌

[1] MIT Spam Conference 2007, http://www.spamconference.org/

1) 악성댓글의 대부분이 띄어쓰기, 맞춤법을 고려하지 않고 비속어의 등록을 위해 단어 사이에 기호들과 공백을 사용하는 점을 감안하여 불필요한 기호와 공백을 제거하여 모든 단어를 한 문장에서의 문자 나열로 인식한다. 본 논문에서는 제거된 기호들(?!/.,)과 공백을 불용어라 한다.