KIPS Tr. Software and Data Eng.
Vol.6, No.9 pp.445~456 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2017.6.9.445

A Method for Twitter Spam Detection Using N-Gram Dictionary Under Limited Labeling

Hyeok-Jun Choi† · Cheong Hee Park††

ABSTRACT

In this paper, we propose a method to detect spam tweets containing unhealthy information by using an n-gram dictionary under limited labeling. Spam tweets that contain unhealthy information have a tendency to use similar words and sentences. Based on this characteristic, we show that spam tweets can be effectively detected by applying a Naive Bayesian classifier using n-gram dictionaries which are constructed from spam tweets and normal tweets. On the other hand, constructing an initial training set requires very high cost because a large amount of data flows in real time in a twitter. Therefore, there is a need for a spam detection method that can be applied in an environment where the initial training set is very small or non exist. To solve the problem, we propose a method to generate pseudo-labels by utilizing twitter's retweet function and use them for the configuration of the initial training set and the n-gram dictionary update. The results from various experiments using 1.3 million korean tweets collected from December 1, 2016 to December 7, 2016 prove that the proposed method has superior performance than the compared spam detection methods.

Keywords : Twitter, Spam, Retweet, N-Gram, Pseudo-Label

트레이닝 데이터가 제한된 환경에서 N-Gram 사전을 이용한 트위터 스팸 탐지 방법

최 혁 준† · 박 정 희††

요 약

본 논문에서는 트레이닝 데이터가 제한된 환경에서 n-gram 사전을 이용하여 불건전 정보를 포함하는 스팸 트윗을 탐지하는 방법을 제안한다. 불건전 정보를 포함하는 스팸 트윗은 유사한 단어와 문장을 사용하는 경향이 있다. 이러한 특성을 이용하여 스팸 트윗과 정상 트윗에 대한 n-gram 사전을 구축하고 나이브 베이스 분류기를 적용하여 효과적으로 스팸 트윗을 탐지할 수 있음을 보인다. 반면에, 실시간으로 대용량의 데이터가 유입되는 트위터의 특성은 초기 트레이닝 집합 구성에 매우 큰 비용을 요구 한다. 따라서, 초기 트레이닝 집합이 매우 작거나 존재하지 않는 환경에서 적용할 수 있는 스팸 트윗 탐지 방법이 필요하다. 이를 위해 트위터의 리트윗 기능을 활용하여 의사 라벨을 생성하고 초기 트레이닝 집합의 구성과 n-gram 사전 업데이트에 활용하는 방법을 제안한다. 2016년 12월 1일부터 2016년 12월 7일까지 수집된 한국어 트윗 130만 건을 사용한 다양한 실험 결과는 비교 방법들보다 제안하는 방법의 성능이 우수함을 입증한다.

키워드 : 트위터, 스팸, 리트윗, N-Gram, 의사 라벨

1. 서 론

2010년 이후 스마트폰 보급이 활성화되면서 트위터, 페이스북과 같은 소셜 미디어가 급성장하기 시작했다. 그 중에서 트위터는 포스팅 당 140자 제한과 단순한 인터페이스를 통한 용이한 접근성을 바탕으로 대표적인 소셜 미디어로써 자리 잡기 시작했다. 현재는 사용자들이 자신의 일상생활에 관련된 주제뿐만 아니라 정치, 경제, 사회 등 다양한 분야에 대해 자신의 의견이나 감정을 공유하는 하나의 시스템이 되었다. 2016년 기준 트위터의 월간 사용자의 수는 3억 명 이상이며[1], 하루에 작성되는 트윗의 양은 약 5억 건 이상이다[2]. 학계에서는 이러한 특징에 주목하여 오래 전부터 트위터 데이터를 사용하여 토픽 추출, 이벤트 탐지, 감성 분석 등을 수행하는 방법들을 제안하였다[3-5].

트위터가 대표적인 소셜 미디어로 성장하게 된 주요 요인

※ 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2015R1D1A1A01056622).
† 준 회 원 : 충남대학교 컴퓨터공학과 석사과정
†† 정 회 원 : 충남대학교 컴퓨터공학과 교수
Manuscript Received : July 11, 2017
Accepted : August 12, 2017
* Corresponding Author : Cheong Hee Park(cheonghee@cnu.ac.kr)

은 팔로우와 리트윗 기능이다. 두 기능은 트위터에서 정보를 확산시키는 데에 주요한 역할을 한다. 팔로우 기능은 다른 사용자와 관계를 맺는 기능으로써 팔로우 기능을 통해 자신이 팔로우하고 있는 사용자의 트윗을 실시간으로 받아볼 수 있으며, 리트윗 기능은 이미 존재하는 특정 트윗을 자신의 팔로워들에게 전파하는 역할을 한다. 만약 화제가 되는 사건이 발생할 경우 해당 사건에 대한 트윗이 급증함과 동시에 리트윗 수치 또한 급격히 높아지게 되며, 이에 따라 리트윗된 트윗을 접하는 사용자들이 급속히 증가하게 된다. 이처럼 트위터에서는 팔로우와 리트윗 기능을 통해 정보의 확산이 단순하게 이루어진다.

그러나 정보의 확산이 단순하게 이루어진다는 특징은 스패머들에게 역으로 이용되어지고 있다. 일반 사용자들은 리트윗 기능을 특정 사건에 대한 정보 전달의 목적을 위해 사용한다. 하지만 스패머들은 리트윗 기능을 자신의 이익을 목적으로 사용하여 스팸 정보가 마치 현재 화제가 되는 사건인 것처럼 위장하기 위해 사용한다. 실제로 트위터의 전체 포스트 중 약 10% 이상을 스팸이 차지하고 있으며[6], 트위터 사용자의 약 5%는 비정상적인 사용자로 프로그램을 통해 자동적으로 스팸을 작성하고 있다[7]. 또한, 최근 스패머들은 스팸 필터링 시스템을 회피하기 위한 목적으로 일반 사용자의 행동을 따라하려는 경향이 있으며, 스팸인 정보들을 인기 있는 토픽으로 위장하기 위해 트위터의 기능들을 악용하고 있다[8].

스팸 정보들을 차단하지 않는다면 스패머에 의해 유입된 각종 음란물, 불법 도박·약물 등의 불건전한 정보를 담고 있는 스팸 트윗들이 확산될 것이며, 일반 사용자들은 이러한 정보에 더욱 쉽게 노출될 것이다. 또한, 스패머에 의해 의도적으로 악용된 행위 때문에 트위터에서 가치 있는 정보를 추출하기 위한 각종 시스템들의 품질 저하를 야기할 수 있고, 이로 인해 트위터에서 실제로 발생하는 사건들에 대한 정보가 아닌 스패머에 의한 정보들로 통계 자료가 구축될 가능성이 있다. 그러나 트위터의 스팸 탐지에 대한 연구들은 트윗 내에 포함된 악성 URL을 탐지하기 위한 연구들이 대부분이며, 이러한 탐지 방법들은 음란물, 불법 도박, 약물 등과 같은 불건전한 정보를 담고 있는 스팸 트윗들을 효율적으로 탐지하는데 한계가 있다.

본 논문에서는 불건전한 정보를 포함하고 있는 음란물, 불법 도박·약물에 대한 트윗들을 탐지하는 방법을 제안한다. 불건전 정보를 포함하는 트윗의 경우 유사한 단어, 문장을 반복하여 사용하는 경향이 있기 때문에 문장에 대한 n-gram을 기반으로 하는 분류기를 사용하는 것이 효과적이다. 제안하는 방법은 스팸 트윗과 정상 트윗에 대한 n-gram 사전을 각각 구축하고 나이브 베이스 분류기를 사용하여 스팸 트윗을 탐지한다. 반면에, 실시간으로 대용량의 데이터가 유입되는 트위터의 특성상 초기 트레이닝 집합을 구성하는 것은 매우 높은 비용을 요구한다. 따라서 초기에 트레이닝 집합이 존재하지 않거나 매우 작은 크기의 트레이닝 집합이 있을 때 적용할 수 있는 탐지 방법이 필요하다. 이를 위하여 트위터의 리트윗 특성을 활용하여 의사 라벨을 생성하고 n-gram 사전 구축과 업데이트에 활용하는 방법을 제안한다. 본 논문에서 제안하는 초기 트레이닝 집합이 작거나 존재하지 않을 때의 스팸 탐지 방법은 학술대회 논문[9]를 확장한 방법이다. 논문의 전체적인 공헌은 다음과 같다.

- 불건전한 정보를 담고 있는 스팸을 탐지하기 위해 n-gram을 이용하는 분류기를 사용하는 것이 높은 성능을 나타냄을 보인다.
- 초기에 트레이닝 집합이 존재하지 않을 때, 트위터의 리트윗 특성을 활용하여 의사 라벨을 생성하고, 이를 트레이닝 집합으로 활용하는 방법에 대해 제안한다.
- 국내의 스팸 탐지에 대한 대부분의 연구들은 매우 적은 양의 데이터를 사용하여, 실제 스팸 탐지의 적용에 대한 확신을 갖기 어렵다. 본 논문에서는 2016년 12월 1일부터 2016년 12월 7일까지 한국어 트윗 약 1,300,000건을 수집하여 현실적인 스팸 탐지의 효용성을 보인다.

논문의 구성은 다음과 같다. 2장에서 관련 연구에 대해 정리하며, 3장에서는 트위터 데이터의 수집 및 라벨링 방법에 대해 기술한다. 4장에서는 제안하는 스팸 탐지 방법에 대해 기술하며, 5장에서는 의사 라벨을 활용하여 스팸 탐지에 적용하는 방법에 대해 기술한다. 6장에서는 실험 및 결과에 대해 정리하고, 7장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

2.1 트위터의 중복 현상에 대한 연구

현실에서 화제가 되는 사건이 발생하게 되면 트위터의 포스팅인 트윗이 급증하게 된다. 트윗들은 최대 작성 가능한 글자가 140자로 매우 짧게 구성되어 있고, 원문의 내용을 복사하여 전파하는 리트윗의 특성 때문에 대부분의 트윗들은 같거나, 매우 유사한 내용으로 구성된다. [10]에서는 트위터의 검색 품질을 높이기 위해 근 중복 탐지(Near Duplicate Detection)에 대한 연구를 진행하였다. 트윗들의 내용에 대한 유사도를 정확, 거의 정확, 강한 중복, 약한 중복, 부분 중복의 5가지로 분류하여 실험을 진행하였으며, 28개의 자질 값에 의한 실험 결과 트윗 내용 유사도에 기반을 두는 자질 값들을 사용하였을 때 근 중복 탐지율이 가장 높았다.

[11]에서는 소셜 미디어에서 존재하는 스패머들을 중복된 내용의 스팸을 반복해서 작성하는 행위, 성인 광고를 작성하는 행위, 상품에 대한 광고를 하는 행위, 사기 행위, 의도적으로 팔로우를 늘리는 행위로 분류하여 스패머들을 탐지하는 연구를 진행하였다. 이 중 중복된 내용의 스팸을 반복적으로 작성하는 스패머들은 작성하는 트윗 내용이 일부만 다를 뿐 거의 유사한 형태로 작성된다고 서술하였다. 본 논문에서 사용한 한국어 트윗 데이터의 경우에도 위와 같은

중복 현상들을 발견하였으며, 특히 스팸으로 라벨 되어 있는 트윗들의 경우에서 중복 현상이 더욱 두드러지게 나타나는 경향이 있었다.

2.2 트위터의 스팸 탐지에 대한 연구

트위터가 대표적인 소셜미디어로써 각광받기 시작한 이후로 트위터에서 스팸 탐지에 대한 많은 연구들이 제안되었다. 대표적인 방법으로, 트위터 API를 사용하여 자질 값을 추출하고, 이를 기계학습 알고리즘에 적용하여 탐지하는 방법들이 제안되었다[12-17]. 사용되는 자질 값들은 사용자 계정 및 콘텐츠에 대한 정보들로써 계정 생성일, 팔로워 수, 팔로잉 수, 포스팅 내의 URL 비율 등이 포함된다. 하지만 이러한 자질 값들은 스패머에 의해 쉽게 위조될 수 있으며, 결과적으로 일반 사용자와 스패머에 대한 분별력이 약해진다는 단점을 갖고 있다.

트윗 자체의 자질 값만을 사용하는 것에 대한 단점을 보완하기 위해 사용자 간의 관계에 대한 그래프 정보까지 포함하여 사용하는 방법들이 제안되었다[18, 19]. 이는 스패머와 일반 사용자는 팔로우 관계를 맺기 힘들다는 특징을 활용한 것으로, 사용되는 자질 값을 보다 확장한 것이다. 그래프 정보를 포함시킴으로써 일반 사용자와 스패머에 대한 분별력이 강해진다는 장점이 있으나, 그래프를 생성하기 위한 비용이 크고, 결과적으로 실시간으로 유입되는 대용량 데이터에 적용하기에는 시간이 오래 걸린다는 단점이 있다.

이 외에 트윗 내에 포함된 URL만을 사용하여 스팸을 탐지하는 연구들이 제안되었다. [20]에서는 도메인 토큰, 패스 토큰, URL 쿼리 파라미터, DNS 정보들을 자질 값으로 사용하여 스팸을 탐지하는 방법에 대해 제안하였다. [21]에서는 트윗 내에 포함된 URL들을 분석하여 리다이렉트 체인의 연관성을 통해 스팸을 탐지하는 방법에 대해 제안하였다. 이러한 URL자체를 사용하는 방법들은 자질 값을 사용하여 스팸을 탐지하는 것보다 효율적이지만, URL이 존재하지 않거나, 문장 자체에 의한 스팸들은 탐지할 수 없다는 단점을 갖고 있다. 트위터에서 스팸을 탐지하기 위한 많은 방법들이 제안되고 있으나, 대부분이 스팸의 범주를 악성 URL을 포함하는 트윗들로 한정하고 있으며, 불건전한 내용을 포함하는 스팸들을 탐지하기 위한 연구는 미흡한 실정이다.

트윗의 문장 자체에 대한 연구와 관련하여 [22, 23]에서는 소셜 미디어에서의 n-gram을 활용한 연구를 진행하였다. [22]에서는 트위터에 대한 연구 목적의 말뭉치(Corpus) 구성을 위해 트윗의 n-gram을 활용하였고, [23]에서는 링크드인(LinkedIn)에서 발생하는 스팸을 탐지하기 위해 n-gram에 기반을 두는 나이브 베이스 분류기를 사용하는 방법에 대해 제안하였다. [23]에서는 초기에 많은 양의 트레이닝 집합을 사용하여 분류기를 구축했을 때의 실험 결과를 보여주고 있다. 그러나, 많은 양의 트레이닝 집합을 구성하는 것은 매우 높은 비용을 요구하며, [23]에서 제안하고 있는 방법의 경우 초기 분류기의 구축 이후 기존의 n-gram에 대한 업데이트를 진행하지 않기 때문에 새로운 n-gram이 발생했을 경우 분류기의 성능이 저하될 여지가 있다.

국내의 스팸 탐지에 대한 연구들은 SMS, 이메일 등에 집중되어 있어[24-27], 한국어로 된 트윗에 대한 연구는 매우 미흡한 실정이다. [28]에서는 단어의 빈도수에 기반을 두는 나이브 베이스를 통해 스팸 트윗을 탐지하는 방법과 URL에 기반한 스팸 탐지 방법에 대해 제안하였으나, 실험 결과를 보면 전체적인 성능이 매우 낮고, 실험한 데이터의 수가 400개로 한정되어 실제 트위터에서 스팸 탐지에 적용 가능한지에 대한 검토가 필요하다.

3. 데이터 수집 및 라벨링

3.1 데이터 수집

본 논문에서 사용하는 트위터 데이터는 트위터의 Streaming API를 사용하여 수집하였다[29]. Streaming API는 트위터에서 실시간으로 발생하는 트윗의 1%를 무작위로 제공하는 API이다. Streaming API를 통해 수집되는 트윗들은 트윗을 작성한 사용자와 트윗에 대한 정보를 JSON 파일의 형태로 반환하게 되며 여기서 필요한 정보만을 파싱하여 사용할 수 있다. 본 논문에서는 2016년 12월 1일부터 2016년 12월 7일까지의 7일 동안 발생한 한국어 트윗만을 수집하였으며, 수집된 트윗의 수는 약 1,300,000만 건이다.

3.2 데이터 라벨링

인터넷 보안업체인 CYREN은 스팸에 대한 주요 카테고리를 성인 만남, 온라인 도박, 불법 약물, 피싱, 스캠 등의 항목으로 구분하였다[30]. 본 논문에서도 수집한 트윗들에 대해 [30]에서 정의한 개념을 이용하여 스팸으로 라벨링을 하였다. 전체적인 라벨링 과정은 모두 수동으로 진행하였다. 수동으로 라벨링을 할 경우 시간이 오래 걸리고, 고비용이 발생한다는 단점이 있으나, 본 논문에서는 URL에 의한 스팸이 아닌 불건전한 내용을 포함하는 스팸들을 탐지하는 것에 초점을 맞추므로 스팸인 단어가 갖는 의미를 파악하기 위해서는 사용자의 판단이 필요하다고 생각되어 수동으로 진행하였다.

Table 1은 라벨링된 데이터에 대한 일자별 스팸(Spam)과 논스팸(Non-spam) 트윗의 개수를 나타낸다.

Table 1. The Description of Tweet Data

Date	Spam	Non-spam	Total
2016/12/01	13,282	222,996	236,278
2016/12/02	12,602	234,885	247,487
2016/12/03	16,771	181,332	198,103
2016/12/04	17,047	165,097	182,144
2016/12/05	15,207	146,651	161,858
2016/12/06	18,721	146,586	165,307
2016/12/07	20,102	136,014	156,116
Total	113,732	1,233,561	1,347,293

4. n-gram 사전에 기반을 두는 스팸 탐지 방법

본 장에서는 분류 모델 학습을 위한 충분한 스팸과 논스팸 트윗 집합이 주어진 환경에서의 스팸 탐지 방법을 제안한다. 제안 방법을 전처리, n-gram 분해, n-gram 사전 구축, 나이브 베이스 분류기 적용 단계로 나누어 상세히 기술한다.

4.1 전처리

트위터에서 사용자들이 작성하는 트윗의 대부분은 맞춤법을 무시하는 경우가 많으며, 또한 과도한 특수문자를 포함하는 경우가 많다. 특히 트위터는 사용자 멘션, 해시태그, 리트윗 등의 기능을 갖고 있어, 사용자가 트윗을 작성할 때 해당 정보들이 트윗 내에 자동으로 포함된다. 이러한 정보들은 트윗 내용 자체에 대한 특성을 반영하지 못하며, 따라서 사전에 제거되어야 한다. 전처리 단계에서 제거되는 정보들은 다음과 같다.

- 리트윗 정보: 하나의 트윗에 대해 리트윗이 발생할 경우 “RT @user_id”의 형태로 문장 내에 자동으로 생성된다.
- 사용자 멘션 정보: 한 사용자가 다른 사용자에게 직접 트윗을 보낼 경우 “@user_id”의 형태로 문장 내에 자동으로 생성된다.
- URL 정보: 사용자가 트윗 내에 URL을 첨부할 경우 단축된 URL이 트윗 내에 첨부된다. 트위터의 경우 “http://t.co/...”의 형태로 URL이 첨부되며, 단축된 URL자체는 트윗 내용에 대한 특성을 반영하지 못한다.
- 특수문자: 스패머의 경우 문장 사이사이에 특수문자를 섞어서 의미적으로 동일한 단어를 반복적으로 사용하는 경우가 많은데, 특수문자를 제거함으로써 이러한 단어들을 찾아낼 수 있다.
- 공백 제거: 앞에서의 정보들을 제거한 후 문장 내에 존재하는 공백들을 제거한다. 즉, 하나의 트윗은 사용자가 작성한 문장을 기준으로 공백 없이 한 문장으로 연결된 형태가 된다.

Fig. 1은 전처리 과정 전·후의 트윗 문장에 대한 예를 보여준다.

전처리 과정에서 형태소 분석은 사용하지 않는다. 형태소 분석에 따른 시간 비용이 크기 때문에 실시간 탐지에 부적합하기 때문이다. 또한, 스패머들은 자신들의 이익을 위한 특정 단어만을 사용하는 경향이 있는데, 형태소를 분석할 경우 그러한 단어들의 의미가 사라질 수 있다. 예를 들어 “사설토토추천사이트”에 대해 형태소 분석을 진행하면 “사설토/NNG+하/XSV+도/EC+추천사/NNG+이/JKS”와 같은 형태로 분해될 수 있으며, 이로 인해 단어 자체가 가지고 있는 본래의 의미를 상실할 수 있다.

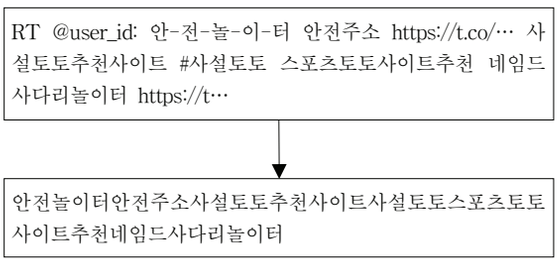

Fig. 1. An Example of Preprocessing

4.2 n-gram 분해

전처리 과정을 마친 트윗에 대해 n-gram으로 분해한다. n-gram에 대한 분해는 문자, 음절, 어절 등으로 분해될 수 있는데, 본 논문에서는 문자 단위로 문장을 분해하였다. 예를 들어, Fig. 1에서 전처리된 문장에 대해 문자 단위로 6-gram으로 분해하였을 때 Table 2와 같이 분해된다. 나이브 베이스 분류기를 사용한 트레이닝과 테스트 단계는 모두 n-gram으로 분해된 단어들을 기반으로 실행된다.

Table 2. An Example of the Transformation from a Tweet to a 6-gram Sequence

Sequence	n-gram
W_0	안전놀이터안
W_1	전놀이터안전
W_2	놀이터안전주
W_3	이터안전주소
…	…
W_{k-1}	드사다리놀이
W_k	사다리놀이터

4.3 n-gram 사전 구축

n-gram으로 분해된 트레이닝 집합의 트윗들을 이용하여 스팸과 논스팸에 대한 사전을 각각 구성한다. Table 2에서와 같이 한 트윗이 n-gram 시퀀스 $< W_0 W_1 ... W_k >$로 구성되고, $c \in \{spam, non-spam\}$를 클래스 정보라고 하자. W_k가 클래스 c에 대한 사전에서 처음 발생했을 경우 W_k에 대한 빈도수는 1로 저장되고, 이미 존재하는 경우에는 기존의 빈도수에 1을 더하여 값을 갱신한다. 이러한 과정은 모든 트레이닝 샘플에 대해 진행된다.

4.4 나이브 베이스 분류기 적용

나이브 베이스 분류기는 문서 분류와 스팸 필터링 분야에 사용되는 대표적인 알고리즘이다[31, 32]. 테스트 단계에서 클래스 라벨에 대한 예측은 트레이닝 단계에서 구축된 n-gram 빈도수 사전 정보를 기반으로 이루어진다. 하나의 트윗이 n-gram 시퀀스 $< W_0 W_1 ... W_k >$으로 구성되었다고 하자. 이에 대한 $c \in \{spam, non-spam\}$의 클래스 조건부 확률은 Equation (1)에 의해 계산된다.

$$P(c|W_0W_1...W_k)=\frac{P(c)\prod_{i=0}^{k}P(W_i|c)}{P(W_0W_1...W_k)} \quad (1)$$

$P(W_i|c)$는 클래스 c 사전에 나타난 n-gram들의 전체 빈도수에 대한 W_i의 빈도수의 비로 계산되고,

$$P(spam)\prod_{i=0}^{k}P(W_i|spam) > P(non-spam)\prod_{i=0}^{k}P(W_i|non-spam) \quad (2)$$

일 때 스팸으로 탐지하게 된다.

하지만 위와 같이 계산할 경우 사전에 존재하지 않는 새로운 n-gram인 W_i가 등장하였을 때 조건부 확률 값이 0이 된다는 문제점이 있다. 이를 해결하기 위해 사전에 존재하지 않는 n-gram이 발생했을 때 확률 값을 계산하기 위해 m-estimate를 사용하였다. m-estimate는 속성 값이 존재하지 않을 때 조건부 확률의 계산결과가 0이 되는 것을 방지한다. m-estimate를 적용하였을 때의 계산은 Equation (3)과 같다.

$$P(W_i|c)=\frac{f_{W_i}+mp_c}{f_c+m} \quad (3)$$

f_c는 클래스 c의 n-gram 사전에 있는 n-gram들의 빈도수 합을 의미하며 f_{W_i}는 W_i가 해당 클래스에서 발생한 빈도수를 의미한다. m은 등가 샘플 크기, p_c는 사용자 정의 매개변수로 이 두 값에 따라서 속성 값이 존재하지 않을 때 기본 확률을 부여할 수 있다. 하지만 m과 p_c의 값이 지나치게 클 경우 오히려 예측되어야 하는 값과 정반대의 결과를 낳을 수 있으며, 따라서 적절한 크기의 값으로 지정해주어야 한다. 본 논문에서는 $m=1$, $p_c=(c$에 대한 $n-gram$ 사전 크기$)^{-1}$로 주어 확률 계산에 사용하였다. p_c의 값을 n-gram사전 크기의 역수로 준 것은 $P(W_i|spam)$인 경우에 스팸 사전에서 발생했을 때의 기본 확률 값을 주는 것이며, $P(W_i|non-spam)$인 경우 논스팸 사전에서 발생했을 때의 기본 확률 값을 주는 것이다.

이에 더하여 본 논문에서는 스패머와 일반 사용자가 사용하는 단어 자체에 대해 주목하였다. [10]에서는 대부분의 트윗들이 중복되는 경우가 많다고 언급하였으며, [11]에서는 스패머가 작성하는 트윗들은 대부분 유사한 단어들로 구성되어 있다고 언급하였다. 실제로 수집한 데이터를 살펴본 결과 스패머가 작성한 트윗들은 대부분 유사한 단어들을 포함하고 있었으며, 차이가 나는 점은 단어 자체의 순서가 바뀌거나 음절 사이사이에 특수문자가 포함되어 있는 경우들이었다. 예를 들어, "사설토토추천사이트"라는 단어의 경우 "사-설-토-토-사-이-트-추-천", "토토사이트추천", "사설토토", "사설토토추천" 등의 변형이 존재하였으나, 이들은 실제로는 동일한 의미를 갖는다. 즉, 스패머는 대부분 유사한 단어를 사용하여 스팸 트윗을 작성하며, 이와는 다르게 일반 사용자들은 자신의 감정, 의견 등에 대해 다양한 단어들을 사용한다고 유추할 수 있다. 이러한 점에 착안하여 나이브 베이스 조건부 확률에 대한 계산을 4가지로 나누어 계산한다.

Table 3. The Four Cases

	Existence in a spam dictionary	Existence in a non-spam dictionary
Case 1	O	O
Case 2	O	X
Case 3	X	O
Case 4	X	X

Table 3은 조건부 확률 계산을 위한 모든 경우들을 나타낸다. Case 1~3의 경우 n-gram W_i가 스팸 또는 논스팸 사전에 존재하는 경우이며, 이러한 경우에는 Equation (3)을 적용하여 $f(W_i|c)$를 계산한다. 하지만 Case 4와 같이 스팸 및 논스팸 사전에 모두 존재하지 않는 경우에는 스팸 사전에서 새로운 n-gram이 발생할 확률보다 논스팸 사전에서 새로운 n-gram이 발생할 확률이 더 높을 것이라 예상할 수 있다. 따라서 Case 4의 경우처럼 n-gram 단어인 W_i가 두 개의 사전에서 모두 발생하지 않은 경우, $0\le\alpha<1$인 α에 대해 Equation (4)로 조건부 확률을 계산한다.

$$P(W_i|spam)=(1-\alpha)$$
$$P(W_i|non-spam)=(1+\alpha) \quad (4)$$

이렇게 함으로써 스팸 사전과 논스팸 사전 모두에 존재하지 않는 새로운 n-gram이 등장하였을 때 논스팸을 스팸으로 예측하는 경우가 적어지게 되며, 이에 따른 분류기의 오분류율을 감소시킬 수 있다. 6장의 실험에서는 α의 값으로 0.1을 사용하였다.

5. 트레이닝 집합이 제한된 상황에서의 스팸 탐지 방법의 적용

트위터와 같이 대량의 데이터가 실시간으로 유입되는 상황에서 초기 분류 모델을 구축하기 위해 트레이닝 집합을 구성하는 것은 매우 큰 비용을 요구한다. 이로 인해 가용한 트레이닝 집합의 크기가 아주 작을 수 있으며, 심지어는 트레이닝 집합 자체가 존재하지 않을 수도 있다. 본 장에서는 이러한 상황에 대한 문제점을 해결하기 위해 트위터의 리트윗에 대한 순간 증가율을 이용하여 의사 라벨을 생성하고, 생성된 의사 라벨을 이용하여 사전 구축과 업데이트에 활용

하는 방법에 대해 설명한다.

5.1절에서는 의사 라벨을 생성하기 위한 전체적인 과정에 대해 설명하며, 5.2절에서는 초기 트레이닝 집합의 크기가 작은 경우에 대해, 5.3절에서는 초기 트레이닝 집합이 전혀 존재하지 않는 경우를 가정하여 의사 라벨을 활용하는 방법에 대해 설명한다.

5.1 의사 라벨 생성

본 논문에서는 의사 라벨을 생성하기 위해 트위터의 리트윗에 대한 순간 증가율을 이용하였다. 리트윗 순간 증가율은 특정 사건이 발생했을 때, 해당 사건에 대한 트윗들의 리트윗이 급격히 증가하는 현상을 반영할 수 있다.

예를 들어 재난 소식, 대통령 선거 등과 같은 화제가 되는 사건이 발생하게 되면 이와 관련된 주제를 갖는 트윗이 급증하게 되는데, 이에 따라 리트윗 수도 급격히 증가하게 된다. 리트윗이라는 기능 자체가 트윗을 전파하기 위한 목적을 갖고 있기 때문에 리트윗 수가 급격히 증가할수록 해당 트윗이 화제가 되는 사건을 내포한다고 볼 수 있다. 그러나 스패머들은 이러한 특성을 악용하여 자신들이 작성한 스팸 트윗이 인기 있는 사건에 대한 내용인 것처럼 위장하기 위해 리트윗 기능을 사용한다.

리트윗의 순간 증가율을 활용하여 의사 라벨을 생성하기 위해서는 먼저 트윗의 특성에 대해 알아야 한다. 트위터에서 발생하는 모든 트윗들은 작성한 사용자의 정보와 콘텐츠에 관한 2가지 정보를 기본적으로 포함하고 있다. 사용자 정보는 사용자 아이디, 사용자의 팔로워와 팔로잉 수 등에 관한 정보를 포함하며, 콘텐츠 정보는 트윗의 고유 아이디, 작성된 시간, 현재 시점까지의 리트윗 수 등에 관한 정보를 포함한다. 특정 트윗에 대해 리트윗이 발생하게 되면 리트윗된 트윗은 자기 자신에 대한 정보와 원본 트윗에 대한 정보를 모두 포함하게 된다. 특히, 원본 트윗을 리트윗한 모든 트윗들은 원본 트윗의 트윗 아이디를 갖고 있으며, 그 시점까지의 원본 트윗에 대한 누적 리트윗 수를 갖는다.

Fig. 2는 원본 트윗 $T(a)$에 대해 발생한 리트윗을 시간에 따라 나타낸 것이다. Fig. 2에서 $Time_i(a)$는 트윗이 발생한 시간을 의미하며, $RC_i(a)$은 $Time_i(a)$까지 누적된 리트윗 수를 나타낸다. 원본 트윗 $T(a)$가 발생한 시점인 $Time_0(a)$에서는 리트윗이 발생하지 않았으므로 리트윗 수의 값은 0이다. $Time_i(a)$에서의 리트윗 수인 $RC_i(a)$과 이전 시점인 $Time_{i-1}(a)$에서의 리트윗 수인 $RC_{i-1}(a)$을 이용하면 구간에서의 리트윗에 대한 증가율을 계산할 수 있다. 특정 구간에서의 리트윗에 대한 순간 증가율(Instantaneous rate of increase)은 Equation (5)를 통해 계산한다. tid는 원본 트윗에 대한 트윗 아이디를 나타낸다.

$$IR_i(tid)=\left(\frac{RC_i(tid)-RC_{i-1}(tid)}{Time_i(tid)-Time_{i-1}(tid)}\right)(i=1,2,...) \quad (5)$$

$IR_i(tid)$은 짧은 시간 동안 리트윗이 많이 발생할수록 높은 값을 갖게 된다. 하지만 Equation (5)는 특정 구간에서의 리트윗 증가율만 반영할 뿐 스패머의 특징을 제대로 반영하고 있지 않다. 스패머의 경우 일반 사용자와 팔로우 관계를 맺기 힘들다는 특징이 있으며, 이로 인해 전체 리트윗 수와 팔로워 수는 일반 사용자보다 낮은 값을 갖게 된다. 이러한 스패머의 특징을 반영하여 리트윗 증가율을 정의한다.

일정 시간 구간 동안 트윗 아이디 tid를 가지는 원본 트윗에 대한 리트윗 $RT_1(tid),...,RT_m(tid)$이 수집되었다고 하자. 그 시간 구간 동안 트윗 $T(tid)$의 리트윗 증가율을 Equation (6)과 같이 정의한다.

$$IR(tid)=\left(\sum_{i=1}^{m} IR_i(tid)\right)/(followers_{tid}*RC_m(tid)) \quad (6)$$

$followers_{tid}$는 원본 트윗을 작성한 사용자의 팔로워 수를 나타내며, $RC_m(tid)$는 마지막 리트윗인 $RT_m(tid)$까지의 누적된 리트윗 수를 의미한다. 따라서 Equation (6)에 의한 계산은 스패머가 일반 사용자보다 높은 값을 갖게 된다. 일정 시간 구간 동안 리트윗이 한 번이라도 발생한 트윗들은 $IR(tid)$값을 가지게 되고, 이 값들을 내림차순으로 정렬하게 되면 상위에는 대부분 스팸 트윗들이 위치하게 되며, 하위에는 대부분 일반 사용자들의 트윗들이 위치하게 된다. 이렇게 순위화된 트윗들 중에서 상위 k개 중에서 스팸 트윗을, 하위 k개 중에서 논스팸 트윗을 추출하여 의사 라벨을 생성하고, 초기 트레이닝의 집합의 구성과 n-gram 사전 업데이트에 활용한다.

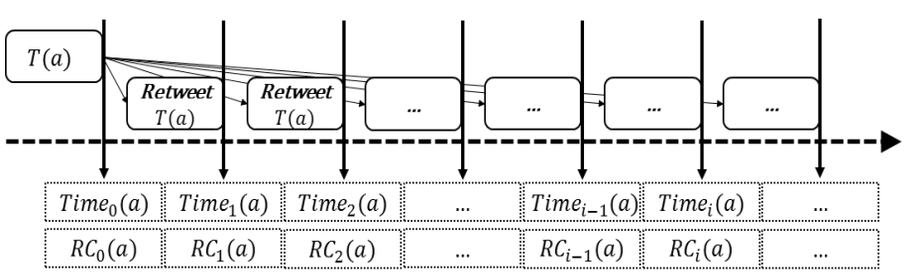

Fig. 2. The Occurrence of Retweets from an Original Tweet

5.2 트레이닝 집합의 크기가 작을 때

초기에 가용한 트레이닝 집합의 크기가 작을 때, 의사 라벨을 이용하는 준지도 학습 방법을 제안한다. 일정한 시간 간격마다 5.1절의 리트윗 증가율을 계산하여 의사 라벨을 가진 집합을 생성하고, n-gram 사전 정보를 주기적으로 업데이트한다. Fig. 3은 준지도학습 방법의 전체적인 흐름을 나타낸다. 업데이트 주기를 T라고 할 때 각 구간마다 클래스 예측과 n-gram 사전 업데이트를 수행한다. 사전 업데이트에 대한 자세한 설명은 아래와 같다.

초기 사전 구축

a) 초기에 주어진 트레이닝 집합으로 스팸과 논스팸에 대한 n-gram사전을 구축한다.

주기적인 사전 업데이트

a) 일정 시간 동안 수집된 트윗들에 대하여 5.1절에서 서술한 리트윗 증가율을 계산하고, 계산된 리트윗 증가율 값을 기준으로 트윗들을 내림차순 정렬한다.
b) 정렬된 트윗들에 대해 기존의 사전 정보를 기반으로4.4절에서의 방법으로 나이브 베이스 분류기를 적용하여 상위 k개, 하위 k개 트윗에 대한 클래스 라벨을 예측한다.
c) 상위 k개의 트윗 중에서 스팸으로 예측된 것들과 하위 k개의 트윗 중에서 논스팸으로 예측된 것들만을 사용하여 기존의 n-gram 사전 정보를 업데이트한다.
d) 일정 시간 간격으로 (a)-(c)의 과정을 되풀이하여 적용한다.

5.3 초기 트레이닝 집합이 주어지지 않았을 때

초기에 가용한 트레이닝 집합이 전혀 존재하지 않을 때, 의사 라벨을 가진 집합을 이용하는 스팸 탐지 방법을 제안한다. 이를 위해 리트윗 증가율을 계산하여 의사 라벨을 생성하고, 초기 사전을 구축한다. 그 이후에 일정한 시간 간격마다 리트윗 증가율을 이용하여 의사 라벨들을 생성하고 사전의 업데이트에 활용한다.

Fig. 4는 전체적인 흐름을 나타내며, 각각에 대한 설명은 다음과 같다.

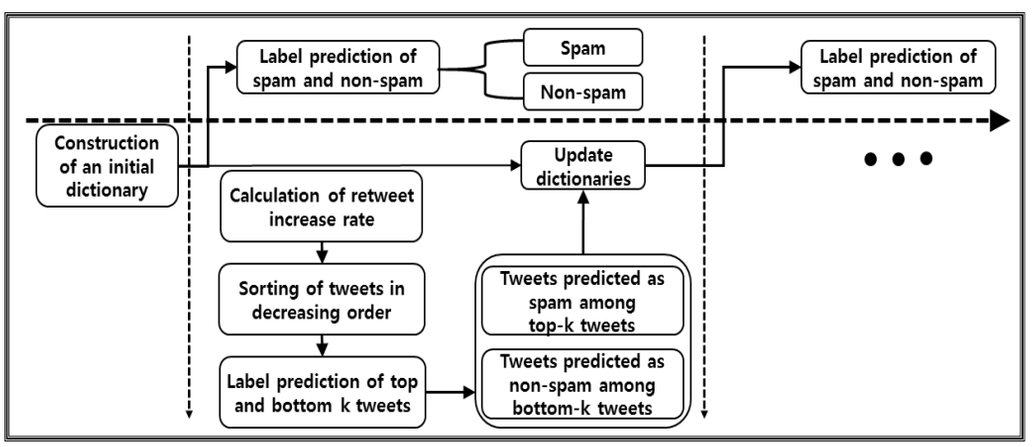

Fig. 3. A Flowchart of the Proposed Method When an Initial Training Set is Small

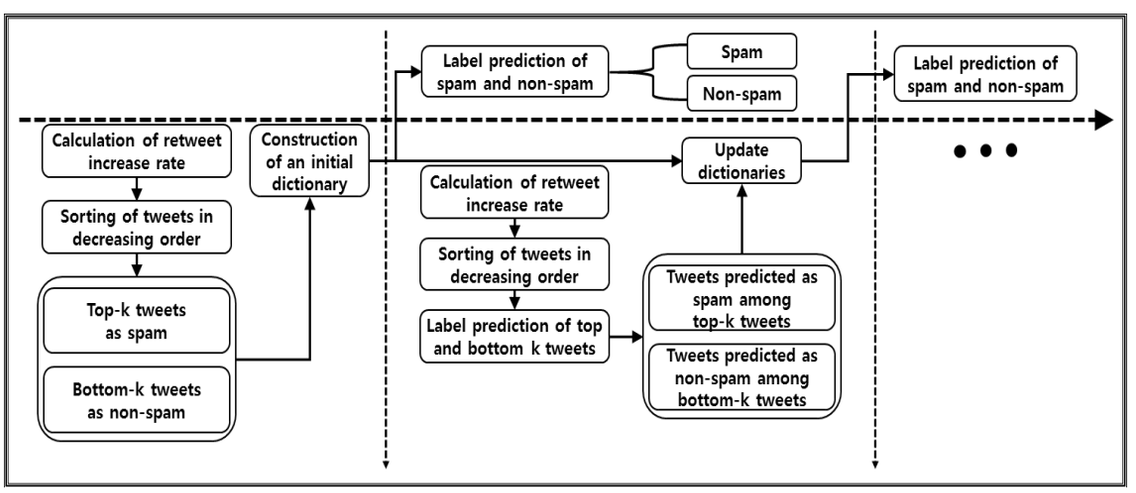

Fig. 4. A Flowchart of the Proposed Method When an Initial Training Set Does Not Exist

초기 사전 구축

a) 일정 구간(시간)동안 리트윗 증가율을 계산한다.
b) 리트윗 증가율 값을 기준으로 트윗들을 내림차순 정렬한다.
c) 정렬된 트윗들에 대해 초기 트레이닝 집합 구성을 위해 상위 k개를 스팸 트레이닝 집합으로 사용하고, 하위 k개를 논스팸 트레이닝 집합으로 사용한다.
d) 트레이닝 집합을 이용하여 사전을 구축한다.

주기적인 사전 업데이트

5.2절의 사전 업데이트 과정과 동일

6. 실험 및 결과

본 실험에서는 불건전한 스팸들을 탐지하기 위해 자질 값을 사용하는 것보다 n-gram을 사용하는 것이 더 효과적임을 보인다. 사용되는 자질 값은 [15]에서 제안한 사용자 및 콘텐츠 정보에 대한 12개의 자질 값이며 각각에 대한 정보는 Table 4와 같다. 분류기로는 나이브 베이스, SVM, 랜덤 포레스트를 사용하였다. 이에 대한 실험은 Weka를 사용하여 진행하였다[33].

Table 4. Twelve Features from [15]

Feature name	Description
account_Age	Age (days) of an account since its creation until the time of sending the most recent tweet
no_follower	Number of followers of this twitter user
no_following	Number of followings/friends of this twitter user
no_userfavourites	Number of favourites this twitter user received
no_lists	Number of lists this twitter user added
no_tweets	Number of tweets this twitter user sent
no_retweets	Number of retweets this tweet
no_hashtag	Number of hashtags included in this tweet
no_usermention	Number of user mentions included in this tweet
no_urls	Number of URLs included in this tweet
no_char	Number of characters in this tweet
no_digits	Number of digits in this tweet

6.1 성능 비교 척도

실험에서는 Precision, Recall, F-measure, FP-rate를 사용하여 성능을 비교하였다. 성능에 대한 평가는 F-measure값을 기준으로 분류기의 성능을 평가하되, 부분적으로 Precision과 FP-rate를 같이 사용하였다. Table 5는 성능 비교를 위한 혼동 행렬을 나타내며, 각각의 비교 척도에 대한 계산은 다음과 같다.

Table 5. A Confusion Matrix

		Predicted label	
		Spam	Non-spam
Actual label	Spam	TP	FN
	Non-spam	FP	TN

$$\mathrm{Precision} = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN},$$

$$F-measure = 2^{*}\frac{(\mathrm{Precision}^{*}Recall)}{(\mathrm{Precision}+Recall)}$$

$$FP-rate = \frac{FP}{FP+TN}$$

- Precision은 스팸으로 예측한 트윗에 대해 실제 스팸인 트윗의 비율을 의미한다.
- Recall은 실제 스팸인 트윗들에 대해 스팸으로 탐지된 것들의 비율을 의미한다.
- F-measure는 Precision과 Recall의 조화 평균이다.
- FP-rate는 실제 논스팸인 트윗들이 스팸으로 탐지된 것들의 비율을 의미하며, 해당 수치가 낮을수록 논스팸에 대한 오분류율이 적어짐을 의미한다.

6.2 스팸 탐지 성능 비교

Table 1에 나타낸 2016년 12월 1일부터 12월 7일까지의 트윗 데이터 중 12월 1일의 데이터는 트레이닝 집합을 구성하기 위해 사용하며, 12월 2일부터 7일까지의 데이터는 테스트 집합으로 사용하였다. n-gram을 이용한 분류기를 사용했을 때 트레이닝 집합의 크기에 상관없이 높은 성능을 나타내는 것을 보이기 위해 트레이닝 집합의 크기를 4가지로 나누어, 트레이닝 집합을 전부 사용하였을 때(스팸 트윗=13,282개, 논스팸 트윗=222,996개)와 스팸과 논스팸 샘플의 개수를 각각 13,282개/1,000개/100개로 조절하였을 때의 성능을 측정하였다. 추가적으로 n-gram의 크기에 따른 분류기의 성능을 측정하기 위해 n의 크기를 2～7로 변화해가며 실험을 진행하였다. Table 6은 모든 샘플들을 트레이닝 집합으로 사용한 실험 결과를 나타내며, Table 7, Table 8, Table 9에 대한 실험은 스팸 샘플의 개수를 각각 13,282/1,000/100개로 설정하고, 논스팸 샘플의 개수를 동일한 크기로 조정한 것이다. 일부의 샘플만 추출하여 사용하게 되는 경우에는 10회 랜덤 샘플링하여 실험을 진행하였고, 성능은 평균값으로 나타내었다.

Table 6, Table 7에 대한 실험은 초기에 충분한 크기의 트레이닝 집합이 존재하는 경우를 의미한다. Table 6, Table 7의 실험 결과를 보면 자질 값을 사용한 분류기의 경우 랜덤 포레스트가 가장 높은 성능을 보였으며, 제안하는 방법의 경우 각각 Table 6에서는 6-gram, Table 7에서는 7-gram을 사용했을 때 가장 높은 성능을 보여주었다. 두 실험 모두 제안하는 방법이 가장 높은 성능을 보여주었다.

Table 6. Results When Using All Training Samples

Method	Classifier	Precision	Recall	F-measure	FP-rate
The method in [15]	NB	0.2549	0.9784	0.4045	0.2842
	RF	**0.9870**	**0.9194**	**0.9520**	**0.0012**
	SVM	0.8023	0.5737	0.6690	0.0141
The proposed method	NB_2g	0.9704	0.9813	0.9758	0.0030
	NB_3g	0.9749	0.9936	0.9842	0.0025
	NB_4g	0.9768	0.9950	0.9858	0.0024
	NB_5g	0.9836	0.9940	0.9888	0.0016
	NB_6g	**0.9916**	**0.9899**	**0.9907**	**0.0008**
	NB_7g	0.9954	0.9773	0.9863	0.0005

Table 7. Results When the Number of Spam and Non-Spam Tweets is 13,282 Respectively

Method	Classifier	Precision	Recall	F-measure	FP-rate
The method in [15]	NB	0.2122	0.9854	0.3491	0.3645
	RF	**0.8933**	**0.9863**	**0.9375**	**0.0117**
	SVM	0.5205	0.9680	0.6770	0.0886
The proposed method	NB_2g	0.9005	0.9921	0.9441	0.0109
	NB_3g	0.9036	0.9971	0.9480	0.0106
	NB_4g	0.9136	0.9967	0.9533	0.0094
	NB_5g	0.9255	0.9950	0.9590	0.0080
	NB_6g	0.9701	0.9902	0.9801	0.0030
	NB_7g	**0.9862**	**0.9765**	**0.9813**	**0.0014**

Table 8. Results When the Number of Spam and Non-Spam Tweets is 1,000 Respectively

Method	Classifier	Precision	Recall	F-measure	FP-rate
The method in [15]	NB	0.1910	0.9815	0.3194	0.4185
	RF	**0.8218**	**0.9744**	**0.8914**	**0.0211**
	SVM	0.5320	0.9668	0.6862	0.0846
The proposed method	NB_2g	0.8653	0.9878	0.9224	0.0153
	NB_3g	0.8637	0.9915	0.9232	0.0156
	NB_4g	0.9025	0.9856	0.9422	0.0106
	NB_5g	0.9464	0.9675	0.9568	0.0055
	NB_6g	**0.9885**	**0.9347**	**0.9609**	**0.0011**
	NB_7g	0.9965	0.8976	0.9444	0.0003

Table 9. Results When the Number of Spam and Non-Spam Tweets is 100 Respectively

Method	Classifier	Precision	Recall	F-measure	FP-rate
The method in [15]	NB	0.2481	0.9688	0.3936	0.3013
	RF	**0.6898**	**0.9598**	**0.8018**	**0.0435**
	SVM	0.4824	0.9691	0.6440	0.1036
The proposed method	NB_2g	0.6852	0.9811	0.8060	0.0454
	NB_3g	0.7958	0.9675	0.8725	0.0251
	NB_4g	**0.9120**	**0.9292**	**0.9203**	**0.0090**
	NB_5g	0.9499	0.8650	0.9052	0.0046
	NB_6g	0.9941	0.7950	0.8834	0.0005
	NB_7g	0.9988	0.7456	0.8537	0.0000

Table 8, Table 9에 대한 실험은 초기에 가용한 트레이닝 집합의 크기가 제한되어 있는 경우를 의미한다. Table 8, Table 9의 실험 결과를 보면 Table 6, Table 7에서와 마찬가지로 자질 값을 사용한 분류기의 경우 랜덤 포레스트가 가장 높은 성능을 보였으며, 제안하는 방법의 경우 Table 8에서 6-gram, Table 9에서 4-gram일 때 가장 높은 성능을 보였다.

두 실험 모두 제안하는 방법을 사용했을 때 가장 높은 성능을 보여주었다. 그러나 Table 8, Table 9에 대한 실험에서는 Table 6, Table 7에서의 실험 결과와는 확연히 다른 양상을 보이고 있다. 먼저, Table 8에서의 실험결과를 보면 랜덤 포레스트의 Precision은 0.8218, 제안하는 방법에 대해 6-gram을 사용했을 때는 0.9885로 약 16%의 차이를 보인다. 또한, 랜덤 포레스트의 FP-rate는 0.0211, 제안하는 방법에 대해 6-gram을 사용했을 때의 FP-rate는 0.0011로 약 2%의 차이를 보인다. Table 9에서 이와 같은 차이는 더욱 두드러지게 나타난다. 랜덤 포레스트의 경우 Precision은 0.6898이고, 제안하는 방법에 대해 4-gram을 사용했을 때의 Precision은 0.9120으로 약 23%의 차이가 발생했다. 또한, 랜덤 포레스트의 FP-rate는 0.0435, 제안하는 방법에 대해 4-gram을 사용했을 때의 FP-rate는 0.0090으로 약 4%의 차이가 발생하는 것을 확인할 수 있다.

랜덤 포레스트의 경우 트레이닝 집합의 크기가 작아질수록 성능이 크게 감소했으며, 제안하는 방법의 경우 트레이닝 집합의 크기가 감소하더라도 준수한 성능을 보여주었다. 이러한 실험 결과를 통해 불건전한 정보를 포함하는 스팸들을 탐지할 때, 자질 값을 이용하는 분류기를 사용하는 것보다 n-gram을 이용하는 분류기를 사용하는 것이 더 효율적이라고 판단할 수 있다.

6.3 초기 트레이닝 집합의 크기가 제한적일 때 의사 라벨생성을 이용한 스팸 탐지 방법에 대한 실험

초기 트레이닝 집합의 크기가 매우 작을 때 의사 라벨을 생성하여 분류기 업데이트에 적용하는 방법에 대한 성능을 테스트한다. 의사 라벨을 생성하기 위한 구간의 크기는 24시간, k=10,000으로 설정하여 실험을 진행하였다. 초기 트레이닝 집합의 크기가 작을 때의 상황을 가정하여, 수집 1일차인 2016년 12월 1일에 수집된 트윗에서 트레이닝 집합으로 스팸 샘플 100개, 논스팸 샘플 100개를 랜덤 샘플링하여 초기 사전을 구축하고, 5.2절에서 설명한 방법으로 일자 별로 의사 라벨을 생성하여 사전을 업데이트하였다. 리트윗 증가율에 의한 내림차순 정렬 후 순위화된 상·하위 k개 트윗들에 대해 라벨을 예측하여 상위 k개에서 스팸으로 예측되는 것과 하위 k개에서 논스팸으로 예측되는 것들만을 사용하여 주기적으로 사전을 업데이트하였다. Table 10에 나타낸 실험 결과에서는 제안하는 방법에 대해 5-gram을 사용했을 때 F-measure의 값이 0.9504로 가장 높은 성능을 보여주었다. 조금 더 자세히 살펴보기 위해 5-gram을 사용했을 때의 일자별 Precision과 F-measure값의 변화를 살펴보면,

Table 10. Results When an Initial Training Set is Small

n-gram	Precision	Recall	F-measure	FP-rate
NB_2	0.8075	0.9483	0.8684	0.0246
NB_3	0.8380	0.9698	0.8985	0.0190
NB_4	0.9336	0.9628	0.9478	0.0069
NB_5	**0.9627**	**0.9384**	**0.9504**	**0.0036**
NB_6	0.9938	0.9015	0.9454	0.0006
NB_7	0.9984	0.8522	0.9195	0.0001

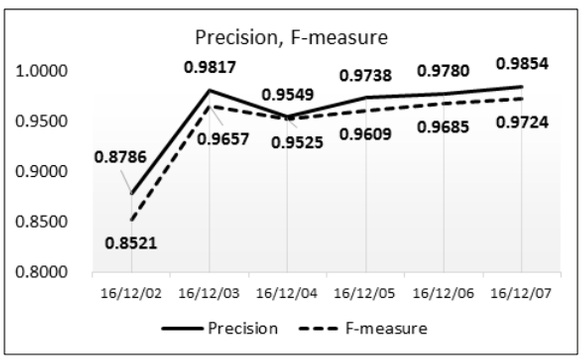

Fig. 5. Precision and F-measure When Using 5-gram

Table 11. Results When an Initial Training Set Does Not Exist

Classifier	Precision	Recall	F-measure	FP-rate
NB_2	**0.8536**	**0.8863**	**0.8696**	**0.0151**
NB_3	0.7823	0.9542	0.8598	0.0264
NB_4	0.7315	0.9537	0.8280	0.0348
NB_5	0.7267	0.9339	0.8174	0.0349
NB_6	0.7654	0.9000	0.8273	0.0274
NB_7	0.8111	0.8552	0.8326	0.0198

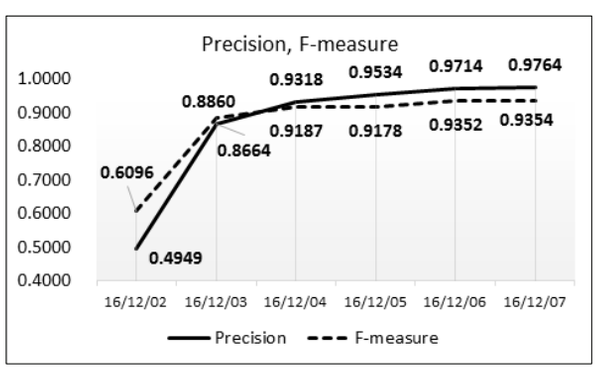

Fig. 6. Precision and F-measure When Using 2-gram

Fig. 5에서 테스트 과정 첫 날인 16/12/02의 경우 Precision은 0.8786, F-measure의 값은 0.8521이었으나, 일자 별로 사전 정보가 업데이트됨에 따라 마지막 날인 16/12/07의 경우 Precision은 0.9854, F-measure는 0.9724로 첫 날인 16/12/02에 비해 각각 약 11%, 12%의 향상된 결과된 결과를 보여주고 있다. 이러한 결과는 Table 6에서 2016년 12월 1일의 모든 샘플을 트레이닝 집합으로 사용하였을 때의 랜덤 포레스트의 F-measure값과 유사한 결과를 보이는 것을 확인할 수 있다. Table 11의 실험은 초기 트레이닝 집합이 주어지지 않은 상황에서 스팸 탐지를 수행하기 위해 5.3절에서 설명한 방법을 적용한 실험 결과이다. 24시간으로 구간을 설정하여 2016년 12월 1일에 발생한 트윗들에 대해 리트윗 증가율을 계산하여 상위 k개는 스팸 집합으로, 하위 k개는 논스팸 집합으로 사용하여 초기 사전을 구축하였다. 또한 24시간마다 주기적으로 사전 업데이트를 수행하였다. Table 11에 나타낸 실험 결과를 보면 제안하는 방법에 대해 2-gram을 사용했을 때 가장 높은 F-measure 값을 보여주었으나, 전체적으로는 성능이 저조한 것처럼 보인다. 그러나 2-gram을 사용했을 때의 일자별 Precision과 F-measure값의 변화를 살펴보면, Fig. 6에서 테스트 첫 날인 16/12/02의 경우 Precision은 0.4949, F-measure는 0.6096으로 저조한 성능을 보여주었으나, 일자별로 사전 업데이트가 이루어짐에 따라 마지막 날인 16/12/07의 경우 Precision은 0.9764, F-measure는 0.9354로 테스트 첫 날인 16/12/02을 기준으로 Precison은 약 37% 향상되었고, F-measure는 약 44% 향상된 수치를 보여주었다. 이러한 결과는 Table 7의 실험 결과에서 스팸과 논스팸의 샘플의 개수를 각각 13,282개로 설정하여 학습을 진행한 랜덤 포레스트와 유사한 성능을 나타낸다. 따라서 본 실험을 통해 의사 라벨을 생성하여 분류기의 초기 트레이닝 집합을 구성하고, 사전 업데이트에 활용했을 때, 분류기의 성능이 빠른 속도로 향상됨을 확인하였다.

7. 결론 및 향후 연구

본 논문에서는 n-gram을 이용하는 나이브 베이스 분류기를 사용하여 트위터에서 불건전한 정보를 포함하는 스팸들을 탐지하는 방법에 대해 제안하였다. 실험 결과를 보면 불건전한 정보를 포함하는 스팸 트윗에 대해 자질 값을 이용하는 분류기를 사용하는 것보다 n-gram을 이용하는 분류기를 사용하는 것이 더욱 뛰어난 성능을 보여주었다. 또한, 트위터의 기능인 리트윗에 대한 증가율을 계산하여 의사 라벨을 생성하고, 이를 스팸 탐지에 활용하는 방법에 대해서도 제안하였다. 실험 결과를 보면 의사 라벨을 생성하여 사용했을 때 트레이닝 집합의 크기가 매우 제한적일 때 또는 전혀 주어지지 않았을 때에도 매우 준수한 성능을 보여주었다.

제안하는 방법에서는 n-gram 사전 구축을 위한 n값을 설정해야 한다. 다양한 환경에서의 실험 결과는 초기 트레이닝 집합의 크기가 작은 경우 n값이 작을수록 높은 성능을 보여주었으며, 트레이닝 집합의 크기가 큰 경우 n값이 클수록 높은 성능을 보여주었다. 따라서 초기 트레이닝 집합의 크기에 따라 적절한 n의 값을 선택하는 것이 필요하다. 또한, 본 논문에서 사용한 데이터는 특정 기간 동안 발생한 한국어 트윗만을 대상으로 삼았으나, 실험 데이터의 확충 및 다른 언어 환경에 대한 실험을 통해 스팸 탐지에 대한 적용 범위를 더욱 확장할 수 있을 것이다.

References

[1] Statista, Number of Monthly Active Twitter Users Worldwide from 1st quarter 2010 to 4th quarter 2016 (in millions) [Internet], https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/.

[2] David Sayce, Number of tweets per day? [Internet], http://www.dsayce.com/social-media/tweets-day/.

[3] L. M. Aiello et al., "Sensing Trending Topics in Twitter," *IEEE Trans. Multimedia.*, Vol.15, No.6, pp.1268-1282, 2013.

[4] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors," in *Proc. 19th International Conference on World Wide Web,* ACM, pp. 851-860, 2010.

[5] A. I. Baqapuri, S. Saleh, M. U. Ilyas, "Sentiment Classification of Tweets using Hierarchical Classification," in *Proc. IEEE International Conference on Communications*, IEEE, 2016.

[6] Neal Ungerleider, Almost 10% of Twitter Is Spam [Internet], https://www.fastcompany.com/3044485/almost-10-of-twitter-is-spam/.

[7] Judy Mottl, Twitter acknowledges 23 million active users are actually bots [Internet], http://www.techtimes.com/articles/12840/20140812/twitter-acknowledges-14-percent-users-bots-5-percent-spam-bots.htm/.

[8] C. Chen, J. Zhang, Y. Xiang, W. Zhou, and J. Oliver, "Spammers Are Becoming "Smarter" on Twitter," *IEEE Trans. IT Professional.*, Vol.18, No.2, pp.66-70, 2016.

[9] H. J. Choi and C. H. Park, "A Twitter Spam Detection Method based on n-gram Dictionary," in *Proc. Korea Computer Congress*, Jeju, pp.227-229, 2017.

[10] K. Tao, F. Abel, C. Hauff, G. J. Houben, and U. Gadiraju, "Groundhog Day: Near-Duplicate Detection on Twitter," in *Proc. 22nd International Conference on World Wide Web*, ACM, pp.1273-1284, 2013.

[11] K. M. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers : social honeypots + machine learning," in *Proc. 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp.435-442, 2010.

[12] F. Benevenuto, G. magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," *Presented at the 7th annual Collaboration Electronic Messaging Anti-Abuse Spam Conference (CEAS)*, Vol.6, 2010.

[13] A. H. Wang, "Don't follow me : spam detection in twitter," in *Proc. International Conference on Security and Cryptography (SECRYPT)*, 2010.

[14] S. Liu, J. Zhang, and Y. Xiang, "Statistical Detection of Online Drifting Twitter Spam," in *Proc. 11th ACM on Asia Conference on Computer and Communications Security*, ACM, pp.1-10, 2016.

[15] C. Chen, et al, "A Performance Evaluation of Machine Learning-Based Streaming Spam Tweet Detection," *IEEE Trans. Computational Social Systems*, Vol.2, No.3, pp.65-75. 2015.

[16] C. Chen, J. Zhang, Y. Xiang, and W. Zhou, "Asymmetric Self-Learning for Tackling Twitter Spam Drift," in *Proc. IEEE Conference on Computer Communications Workshops*, IEEE, pp.208-213, 2015.

[17] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. 26th Annual Computer Security Applications Conference*, ACM, pp.1-9, 2010.

[18] J. Song, S. Lee, and J. Kim, "Spam filtering in Twitter using sender-reeiver relationship," in *Proc. 14th International Conference on Recent Advances in Intrusion Detection*, Springer Berlin/Heidelberg, pp.301-317, 2011.

[19] C. Yang, R. Harkreader, and G. Gu, "Empirical evaluation and new design for fighting evolving twitter spammers," *IEEE Trans. Information Forensics and Security*, Vol.8, No. 8, pp.1280-1293, 2013.

[20] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proc. IEEE Symposium on Security and Privacy, Washington*, pp.447-462, 2011.

[21] S. H. Lee and J. Kim, "Warningbird : A near real-time detection system for suspicious URLs in Twitter spammers," *IEEE Trans. Information Forensics and Security*, Vol.8, No. 8, pp.1280-1293, 2013

[22] D. M. Freeman, "Using Naive Bayes to Detect Spammy Names in Social Networks," in *Proc. the 2013 ACM Workshop on Artificial Intelligence and Security*, ACM, pp. 3-12, 2013

[23] A. Herdağdelen, "Twitter n-gram corpus with demographic metadata," *Language Resources and Evaluation*, Vol.47, No. 4, pp.1127-1147, 2013.

[24] S. J. Lee and D. J. Choi, "Personalized Mobile Junk Message Filtering System," *The Journal of the Korea Contents Association*, Vol.11, No.12, pp.122-135, 2010.

[25] H. N. Lee, M. G. Song, and E. G. Im, "A Study on Structuring Spam Short Message Service(SMS) filter," in *Proc. Symposium of the Korean Institute of communications and Information Sciences*, pp.1072-1073, 2011.

[26] S. W. Lee, "Spam Filter by Using X2 Statistics and Support Vector Machines," *KIPS Journal B (2001~2012)*, Vol.17B, No.3, pp.249-254, 2010.

[27] I. W. Joe and H. T. Shim, "A SVM-based Spam Filtering System for Short Message Service (SMS)," *The Journal of The Korean Institute of Communication Sciences*, Vol.34, No.9, pp.908-913, 2009.

[28] Y. H. Kim et al., "Spam Twit Filtering using Naïve Bayesian Algorithm and URL Analysis," in *Proc. Korean Institute of Information Scientists and Engineers*, Vol.38, No.2B, pp. 375-378, Nov., 2011.

[29] Twitter, Inc., Streaming APIs [Internet], https://dev.twitter.com/streaming/overview.
[30] Cyren, Q3 Trend Report Highlights Real-Time Malware Campagigns And Increase In Phishing [Internet], https://blog.cyren.com/articles/commtouch-internet-threats-trend-report-q3-2013.html.
[31] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?," in *Proc. the Third Conference on Email and Anti-Spam*, pp.28-69, 2006.
[32] J. Graovac, "Text Categorization Using n-Gram Based Language Independent Techniques," in *Proc. 35th Anniversary of Computational Linguistics*, pp.124-135, 2014.
[33] Machine Learning Group at the University of Waikato, Weka3: Data Mining Software in Java [Internet], http://www.cs.waikato.ac.nz/ml/weka/.

최 혁 준

e-mail : nujhch90@gmail.com
2016년 충남대학교 컴퓨터공학과(학사)
2016년~현 재 충남대학교 컴퓨터공학과 석사과정
관심분야 : 데이터마이닝, 정보 검색, 자연언어 처리

박 정 희

e-mail : cheonghee@cnu.ac.kr
1998년 연세대학교 수학과(박사)
2004년 University of Minnesota, Computer Science & Engineering(박사)
2005년~현 재 충남대학교 컴퓨터공학과 교수
관심분야 : 데이터마이닝, 기계학습, 패턴 인식