Journal of Korean Institute of Intelligent Systems, Vol. 24, No. 5, October 2014, pp. 482-488
http://dx.doi.org/10.5391/JKIIS.2014.24.5.482

대용량 소셜 미디어 감성분석을 위한 반감독 학습 기법

Semi-supervised learning for sentiment analysis in mass social media

홍소라 · 정연오 · 이지형†

Sola Hong, Yeounoh Chung, and Jee-Hyong Lee†

성균관대학교 정보통신대학

College of Information and Communication Engineering, Sungkyunkwan University

요 약

대표적인 소셜 네트워크 서비스(SNS)인 트위터의 내용을 분석하여 자동으로 트윗에 나타난 사용자의 감성을 분석하고자 한다. 기계학습 기법을 사용해서 감성 분석 모델을 생성하기 위해서는 각각의 트윗에 긍정 또는 부정을 나타내는 감성 레이블이 필요하다. 그러나 사람이 모든 트윗에 감성 레이블을 붙이는 것은 비용이 많이 소요되고, 실질적으로 불가능하다. 그래서 본 연구에서는 “감성 레이블이 있는 데이터”와 함께 “감성 레이블이 없는 데이터”도 활용하기 위해서 반감독 학습 기법인 self-training 알고리즘을 적용하여 감성분석 모델을 생성한다. Self-training 알고리즘은 “레이블이 있는 데이터”의 레이블이 있는 데이터를 활용하여 “레이블이 없는 데이터”의 레이블을 확정하여 “레이블이 있는 데이터”를 확장하는 방식으로, 분류모델을 점진적으로 개선시키는 방식이다. 그러나 데이터의 레이블이 한번 확정되면 향후 학습에서 계속 사용되므로, 초기의 오류가 계속적으로 학습에 영향을 미치게 된다. 그러므로 조금 더 신중하게 “레이블이 없는 데이터”의 레이블을 결정할 필요가 있다. 본 논문에서는 self-training 알고리즘을 이용하여 보다 높은 정확도의 감성 분석 모델을 생성하기 위하여, self-training 중 “감성 레이블이 없는 데이터”의 레이블을 결정하여 “감성 레이블이 있는 데이터”로 확장하기 위한 3가지 정책을 제시하고, 각각의 성능을 비교 분석한다. 첫 번째 정책은 임계치를 고려하는 것이다. 분류 경계로부터 일정거리 이상 떨어져 있는 데이터를 선택하고자 하는 것이다. 두 번째 정책은 같은 개수의 긍/부정 데이터를 추가하는 것이다. 한쪽 감성에 해당하는 데이터에만 국한된 학습을 하는 것을 방지하기 위한 것이다. 세 번째 정책은 최대 개수를 고려하는 것이다. 한 번에 많은 양의 데이터가 “감성 레이블이 있는 데이터”에 추가되는 것을 방지하고 상위 몇%만 선택하기 위해서, 선택되는 데이터의 개수의 상한선을 정한 것이다. 실험은 긍정과 부정으로 분류되어 있는 트위터 데이터 셋인 Stanford data set에 적용하여 실험하였다. 그 결과 학습된 모델은 “감성 레이블이 있는 데이터” 만을 가지고 모델을 생성한 것보다 감성분석의 성능을 향상 시킬 수 있었고 3가지 정책을 적용한 방법의 효과를 입증하였다.

키워드: 트위터, 감성분석, 반감독 학습 기법, Self-training, SVM

Abstract

This paper aims to analyze user's emotion automatically by analyzing Twitter, a representative social network service (SNS). In order to create sentiment analysis models by using machine learning techniques, sentiment labels that represent positive/negative emotions are required. However it is very expensive to obtain sentiment labels of tweets. So, in this paper, we propose a sentiment analysis model by using self-training technique in order to utilize "data without sentiment labels" as well as "data with sentiment labels". Self-training technique is that labels of "data without sentiment labels" is determined by utilizing "data with sentiment labels", and then updates models using together with "data with sentiment labels" and newly labeled data. This technique improves the sentiment analysis performance gradually. However, it has a problem that misclassifications of unlabeled data in an early stage affect the model updating through the whole learning process because labels of unlabeled data never changes once those are determined. Thus, labels of "data without sentiment labels" needs to be carefully determined. In this paper, in order to get high performance using self-training technique, we propose 3 policies for updating "data with sentiment labels" and conduct a comparative analysis. The first policy is to select data of which confidence is higher than a given threshold among newly labeled data. The second policy is to choose the same number of the positive and negative data in the newly labeled data in order to avoid the imbalanced class learning problem. The third policy is to choose newly labeled data less than a given maximum number in order to avoid the updates of large amount of data at a time for gradual model updates. Experiments are conducted using Stanford data set and the data set is classified into positive and negative. As a result, the learned model has a high performance than the learned models by using "data with sentiment labels" only and the self-training with a regular model update policy.

Key Words : Twitter, Sentiment analysis, Semi-supervised learning, Self-training, SVM

1. 서 론

트위터는 웹상에서 이용자들이 인적 네트워크를 형성할 수 있게 해주는 서비스를 제공하는 소셜 네트워크 서비스 (SNS)의 하나이다. 트위터는 정보 공유와 자신의 감성을 표현하는 수단으로 이용되기도 한다. 트위터에 나타나는 감성을 토대로 트위터에 나타난 감성을 분석하기 위해 많은 연구들이 진행되고 있다[1][4][5][6][7]. 데이터를 분석하고자 할 때에는 학습 데이터가 많을수록 학습은 정교해 진다[14]. 같은 맥락에서 긍정 또는 부정의 감성을 구분하는 분류 모델을 생성하기 위해서는 학습에 사용될 “감성 레이블이 있는 데이터”가 많을수록 좋은 성능을 낸다. 하지만, “감성 레이블이 있는 데이터”를 구하는 것은 시간과 노력이 수반된다. 반면 “감성 레이블이 없는 데이터”는 상대적으로 구하기 쉬우며 방대하다. 본 논문에서는 적은 양의 “감성 레이블이 있는 데이터”와 방대한 양의 “감성 레이블이 없는 데이터”를 가지고 감성 분석 모델을 생성하고자 한다. “감성 레이블이 있는 데이터”와 “감성 레이블이 없는 데이터“를 모두 활용할 수 있는 기계학습 기법인 self-training 알고리즘을 이용하여 트위터에 나타난 감성을 분석하고자 한다. Self-training 알고리즘은 “감성 레이블이 있는 데이터”를 이용해 초기 모델을 생성한 후, 생성된 모델로 “감성 레이블이 없는 데이터”의 레이블을 결정하여 모델을 업데이트 하는 기법이다. 이 기법은 비교적 적은 양의 “감성 레이블이 있는 데이터”로 초기 모델을 생성할 수 있다.

Self-training 알고리즘은 데이터의 레이블이 한번 확정되면 향후 학습에서 계속 사용되므로, 초기의 오류가 계속적으로 학습에 영향을 미치게 된다. 그러므로 조금 더 신중하게 “감성 레이블이 있는 데이터”로 업데이트 할 필요성이 있다. 본 논문에서는 self-training 알고리즘을 이용하여 보다 높은 성능의 감성 분석 모델을 생성하기 위하여, “감성 레이블이 있는 데이터”를 확장하기 위한 3가지 정책을 제시하고 이에 따라 생성된 감성 분석 모델의 성능을 비교 분석하였다.

첫 번째 정책은 임계치를 고려하는 것이다. 분류 경계에 가까운 데이터는 상대적으로 레이블이 불확실하고, 분류 경계에서 먼 데이터는 레이블이 확실하다. 분류 경계로부터 일정거리 이상 떨어져 있는 데이터를 선택하여 업데이트 하고자 임계치를 적용한다. 두 번째 정책은 같은 개수를 고려하는 것이다. 모델에 의해 긍정 또는 부정의 감성으로 분류되어서 예측된 레이블을 붙여 “감성 레이블이 있는 데이터”에 추가할 데이터를 선택할 때, 한 쪽 감성에만 국한된 학습하는 것을 방지하기 위하여 긍정과 부정의 개수를 맞추어 선택하고자 하는 정책이다. 세 번째 정책은 최대 개수를 고려하는 것이다. 한 번에 많은 양의 데이터가 “감성 레이블이 있는 데이터”로 추가되는 것을 방지하고 상위 몇%만 선택하기 위해서, 선택되는 데이터의 개수를 제한하여 업데이트 하고자 하는 정책이다. 이러한 3가지 정책에 기반하여 데이터를 선택하여 “감성 레이블이 있는 데이터”에 추가함으로써 감성 분석 모델의 분류 성능을 향상시키고자 하였다.

실험에서, 감성 분석 모델은 긍정과 부정으로 분류되어 있는 트위터 데이터 셋인 Stanford data set 중에서 추출한 200개의 “감성 레이블이 있는 데이터”와 9,800개의 “감성 레이블이 없는 데이터”로 생성하였으며, 모델의 성능은 359개의 “감성 레이블이 있는 데이터”를 사용하여 검증하였다. 그 결과 학습모델은 실제 답이 매겨진 “감성 레이블이 있는 데이터”만으로 모델을 생성한 것 보다 감성 분석의 성능이 향상되었다.

논문의 구성은 2장에서는 관련 연구, 3장은 배경 지식, 4장은 제안 기법, 5장은 실험, 6장은 결론으로 구성된다.

2. 관련 연구

감성 분석은 영화 흥행이나 마케팅 분야에서 많은 관심을 가지고 있다.

Bo Pang 외 2인은 영화 리뷰에 나타난 감정을 자동으로 긍정/부정으로 분류하고자 하였다[1]. 감정분류에 효율적인 특징들을 알고자 하며, SVM, Maximum entropy, Naive Bayes의 3가지 ML 기법을 사용하여 각각의 장단점을 비교해 보고자 하였다. 특징들의 조합을 비교분석 하여 해당 단어의 존재 여부를 특징으로 사용하는 것이 제일 좋은 결과를 보인다.

강한훈 외 2인은 상품리뷰에 대해서 속성별 긍/부정 분류 시스템을 설계하는 기법을 제안하였다[2]. 사전에 ‘명사+동사’등의 품사들의 조합과 긍/부정의 감성레이블로 구성된 “패턴 DB”를 구축한다. 입력된 리뷰가 “패턴 DB”내의 패턴과 일치한다면 그 패턴에 기록되어 있는 감성이 입력된 리뷰의 감성이 된다. 리뷰에서는 특정 상품, 특정 속성을 언급하므로 구축된 “패턴 DB”와의 비교를 통해 감성을 판단할 수 있으나, 특정한 패턴이 없는 트윗의 감성 분석에의 적용은 힘들다.

AGARWAL외 4인은 트위터에 나타난 감정을 분석하고자 이모티콘과 줄임말등의 특징들을 정의하였다[3]. 모든 분류는 SVM으로 이루어 졌고 데이터를 unigram 모델상에서 감정을 분석한 결과가 성능이 좋았다. 하지만 특징의 값이 감정 표현 단어들의 개수나 이모티콘의 개수로 정의함으로서 감정 표현단어의 강도를 고려하지 않으므로, good과 excellent의 감정 분석 결과는 동등하다. good, excellent의 단어와 동시에 부정적인 단어가 쓰였을 때의 감정 분석 결과 역시 동등하게 나오는 단점이 있다.

접수일자: 2014년 3월 9일
심사(수정)일자: 2014년 4월 1일
게재확정일자 : 2014년 9월 12일
† Corresponding author
감사의 글: 본 연구는 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 연구결과 입니다. (NRF-2012R1A1A2008062) 또한, 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였습니다. (10041244, 스마트TV 2.0 소프트웨어 플랫폼) 연구비 지원에 감사드립니다.

강인수는 단어의 의미나 품사의 차이에 따라 감성 분석의 정확도가 얼마나 향상되는지 알아보는 연구를 하였다[4]. 감성단어사전인 SentiWordNet을 기반으로 단어의 의미별, 품사별 극성을 측정하여 고려/비고려에 따른 4가지 경우를 비교실험 하였다. 이 연구는 문장에서 특징을 추출하는 과정을 세분화함에 따라 더 향상된 정확도를 보이는 감성분석을 하고자 하였다. 하지만 감성단어사전에 존재하지 않는 단어에 대한 극성정보(감성 레이블)를 알 수 없다.

Hogenboom외 2인은 이모티콘과 감성언어를 활용한 감성분석을 한다[5]. 1개의 문장 혹은 문단에 대해서 문장이나 문단에 속하는 모든 요소들에 대해서 감성을 부여한다. 이 과정에서 이모티콘이 있으면 이모티콘에 대한 감성을 수치로 나타내고, 감성 단어가 있으면 감성 단어에 대한 감성을 수치로 나타낸다. 이모티콘이 존재하면 감성 단어에 대한 감성 점수는 고려하지 않고, 이모티콘이 없으면 감성 단어에 대한 감성 점수만을 적용하여 한 문장이나 문단에 대한 감성을 분류한다. 이 연구는 이모티콘의 유무에 따라 감성을 판단하는 방식이 다르며 문장 단위의 감성분석과 문단 단위의 감성분석을 하였다. 문단 단위의 감성 분류가 잘 된 것으로 봐서, 긴 문장 혹은 여러 단락으로 구성된 글귀에 대한 감성을 잘 분류하는 것으로 보인다. 그래서 비교적 짧은 문장인 트윗의 감성을 알고자 할 때는 분류 성능이 좋지 않다.

위에서 언급한 기존의 방법들은 특정 단어나 품사에 의해 감성을 알 수 있으며, 정답 레이블이 있는 경우의 학습 결과를 보여준다[1][2][3][4][5]. 트윗은 특정한 패턴이 없고 단문으로 나타나며, “감성 레이블이 있는 데이터”를 구하는 비용이 많이 든다. 본 연구에서는 생성된 모델을 이용해 “감성 레이블이 없는 데이터”에 감성 레이블을 매기어 나감으로써 감성 분석 모델을 개선하는 self-training 알고리즘을 사용한다.

3. 배경 지식

이 장에서는 제안 기법에 사용되는 배경지식에 대해 설명한다. 기본적인 감성 분석 모델의 학습과정과 모델을 생성할 때 사용된 기계학습 기법, 그리고 실험에 사용된 트윗을 벡터형식으로 표현하는 법을 기술한다. 마지막으로 “감성 레이블이 없는 데이터”를 활용하고자 적용한 기계학습 기법인 self-training 알고리즘에 대해서 설명한다.

3.1. 감성 분석 모델

감성 분석을 위해서는 “감성 레이블이 있는 데이터”를 이용하여 모델을 학습시켜야 하며, 감성 분석 모델의 학습과정은 그림 1과 같다.

감성 분석 모델은 “감성 레이블이 있는 데이터”를 이용하여 학습한다. 그림 1에서는 학습 데이터로 4개의 문장이 있으며, 부정을 나타내는 감성 레이블이 붙은 문장1과 문장2가 있으며, 긍정을 나타내는 감성 레이블이 붙은 문장3과 문장4가 있다. 감성 분석 모델은 “감성 레이블이 있는 데이터”인 4개의 문장을 학습한다. 학습된 감성 분석 모델은 새로운 문장인 문장5가 입력

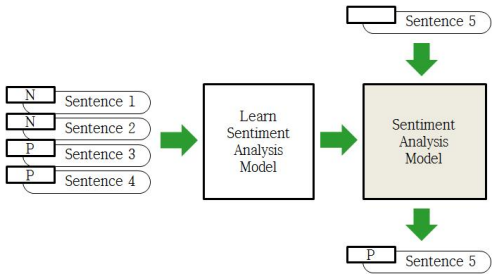

그림 1. 감성 분석 모델
Fig. 1. Sentiment Analysis Model.

으로 들어왔을 때, 문장5가 긍정을 나타내는 문장인지 부정을 나타내는 문장인지 판단한다. 문장5는 감성 분석 모델에 의해 긍정적인 문장으로 판단되었으며, 이 과정은 긍정인지 부정인지 모르는 문장에 대해서 문장의 감성을 예측하는 과정이다.

3.2. Support Vector Machine (SVM)

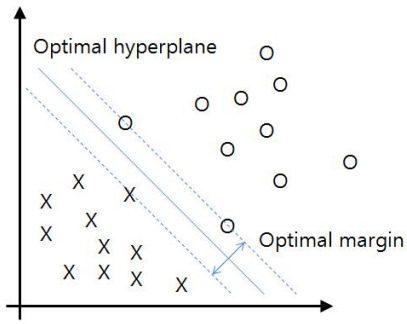

그림 2. Support Vector Machine.
Fig. 2. Support Vector Machine.

Support Vector Machine은 기계학습 기법으로 1979년 Vapnik에 의해 제안된 기법으로, 주어진 데이터들을 2개의 집단으로 분리시키는 최적의 초평면(Hyperplane)을 찾고자 하는 알고리즘이다[8]. SVM은 "레이블이 있는 데이터"를 학습시켜 모델을 생성하는 감독 학습 기법 중 하나이다. 그림 2에서는 O와 X로 표시된 데이터들을 분류하기 위한 분류경계인 초평면이 실선으로 형성되었고, 두 부류 사이에 존재하는 margin을 최대화하고자 하는 분류법이다.

3.3. 문장의 표현

문장을 벡터로 표현하기 위해서 bag of words 모델을 생성한다. bag of words 모델이란 글에 포함된 단어의 분포를 보기 위한 기법이다. 본 연구에서는 학습에 사용된 트윗에 나타난 단어들을 기반으로 단어 사전을 구축한다. 트윗에 나타난 단어의 존재 유무에 따라 단어가 있으면 1로 표현하고, 단어가 없으면 0으로 표현한다. 예를 들어 "I like you." 이라는 문장을 벡터로 표현하게 되면 그림 3와 같다.

단어 사전을 구축하는 순서는 다음과 같다. 1단계로 학습 데이터에 나오는 모든 단어를 추출한다. 2단계로

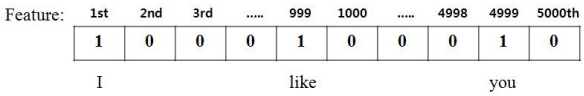

그림 3. 단어 벡터 표현.
Fig. 3. Word vector representation.

는 추출된 모든 단어들을 대문자에서 소문자로 치환하는 작업을 한다. 이 작업은 같은 단어이지만 대문자와 소문자의 차이로 다른 단어로 인식될 수 있기 때문에 수행한다. 3단계로는 특수문자와 URL을 제거하는 작업을 한다. 그리고 4단계에서는 a/an/the등의 불용어를 제거한다. 5단계에서는 중복된 단어를 제거한다. 예는 그림 4과 같다.

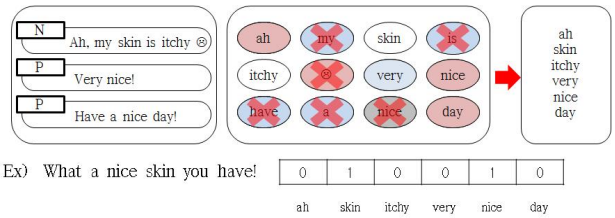

그림 4. 단어 벡터 표현 예시.
Fig. 4. Word vector representation example.

그림 4에서는 "Ah, my skin is itchy ☹", "very nice!", "Have a nice day!"라는 3개의 문장이 있다. 1단계로 이 3개의 문장에서 나타난 단어들을 모두 추출한다. 이 단어들은 단어 사전을 생성하기 위한 후보군이 된다. 2단계로는 추출된 단어 사전 후보군들을 대문자에서 소문자로 치환한다. 그래서 단어 "Ah,"는 "ah," 로 단어 "Very"는"very"로 단어 "Have"는 "have"로 치환된다. 3단계로는 특수문자와 URL을 제거하는 과정으로서, 단어 "ah,"는 "ah"로 이모티콘 "☹"는 삭제되며, 단어 "nice!"와 "day!"는 느낌표가 제거되어 각각 "nice"와 "day"로 치환된다. 그리고 4단계로 a/an/the 등의 불용어를 제거한다. 여기서 불용어에 해당하는 단어인 단어 "my", "is", "have", "a"의 단어가 제거된다. 5단계에서는 2번 이상 나타난 단어인 "nice"가 제거되어 최종적으로 "ah", "skin", "itchy", "very", "nice" ,"day"의 6단어가 단어 사전을 구성한다. 구축된 단어 사전으로 예문 "What a nice skin you have!"를 단어 벡터로 표현한다면 예문에 단어"skin"과 "nice"가 존재하므로 예문은 (0 1 0 0 1 0) 으로 표현될 수 있다.

3.4. Self-training

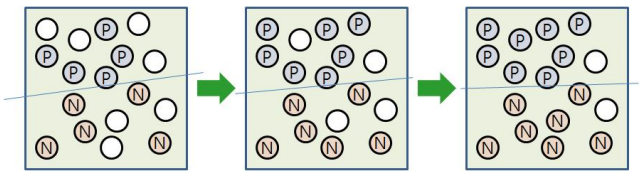

그림 5. Self-training.
Fig. 5. Self-training.

Self-training 알고리즘은 반감독 학습 기법 중 하나로 "레이블이 있는 데이터"와 "레이블이 없는 데이터"를 둘 다 활용하는 기계 학습 기법이다. Self-training 알고리즘은 "레이블이 있는 데이터"로 모델을 생성하고, 생성된 모델로 "레이블이 없는 데이터"의 레이블을 예측한다. 예측 결과, 레이블이 확실하다고 여겨지는 "레이블이 없는 데이터"를 선택하고 예측된 레이블을 붙여 "레이블이 있는 데이터"에 추가한다. 추가하여 업데이트된 "레이블이 있는 데이터"를 이용하여 학습하여 모델을 업데이트하고, "레이블이 없는 데이터"의 레이블을 예측하는 과정을 반복함으로써 학습 모델을 수정해 나가는 것이다. 그림 5는 self-training의 동작원리를 나타내는 그림으로서 매 단계마다 "레이블이 있는 데이터"를 이용해 모델을 생성하고, 생성된 모델에 의해 "레이블이 없는 데이터"의 레이블을 예측하여 신뢰도가 높은 몇몇 데이터를 선별하여 "레이블이 있는 데이터"로 추가되며, "레이블이 있는 데이터"가 업데이트됨에 따라 모델도 업데이트 되는 것을 보여준다.

그러나 self-training 알고리즘은 데이터의 레이블이 한번 확정되면 향후 학습에서 계속 사용되므로, 초기의 오류가 계속적으로 학습에 영향을 미치게 된다. 그러므로 조금 더 신중하게 "레이블이 없는 데이터"의 레이블을 결정할 필요가 있다. 본 논문에서는 self-training 알고리즘을 이용하여 보다 높은 정확도의 감성 분석 모델을 생성하기 위하여, self-training 중 "감성 레이블이 없는 데이터"의 레이블을 결정하여 "감성 레이블이 있는 데이터"로 확장하기 위한 3가지 정책을 제시하고, 각각의 성능을 비교 분석한다.

4. 제안 기법

본 절에서는 "감성 레이블이 있는 데이터"를 이용하여 모델을 생성하고, 생성된 모델을 통해 "감성 레이블이 없는 데이터"를 학습 하는 방법에 대해 기술하고, self-training 알고리즘을 이용하여 보다 높은 정확도의 감성 분석 모델을 생성하기 위한 방법을 제안한다. 본 논문에서는 "감성 레이블이 있는 데이터"뿐만 아니라 "감성 레이블이 없는 데이터"도 활용하기 위하여 "감성 레이블이 있는 데이터"로 모델을 생성하고, 생성된 모델로 "감성 레이블이 없는 데이터"의 감성 레이블을 예측한다. 예측된 결과 중 다른 데이터들보다 긍정 또는 부정의 감성이 확실하다고 판단되는 데이터들을 "감성 레이블이 있는 데이터"에 추가하고자 한다. 예측된 결과를 기반으로 "감성 레이블이 없는 데이터"를 선별하여 선택하여 "감성 레이블이 있는 데이터"에 추가하고자 하는 판단을 도울 수 있는 3가지 정책을 제안한다. 3가지 정책은 그림 6과 같다.

첫 번째 정책으로는 임계치를 고려하는 방법이다. 임계치는 데이터가 분류경계로부터 일정 거리 이상 떨어져 있는 데이터들을 선택하고자 하는 방법이다. 분류경계에 가까운 데이터는 레이블이 불확실하다는 것을 나타내며, 노이즈일 가능성이 크다. 반면 분류 경계에서 먼 데이터는 레이블이 확실하다는 것을 나타낸다. 임계치를 고려함으로써 분류경계로부터 일정거리 이상 떨어져 있는 데이터를 선택하여 "감성 레이블이 있는 데이

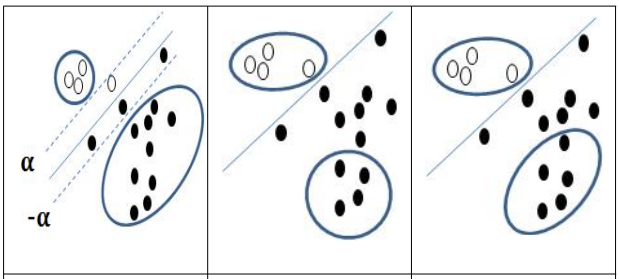

그림 6. 3가지 정책.
Fig. 6. 3 policys.

터"로 업데이트 하고자 하는 방법이다. 그림 6의 첫 번째 그림은 임계치를 α로 정하고, 분류경계로부터 α보다 먼 데이터들을 선택한다.

두 번째 정책은 같은 개수를 고려하는 방법이다. "감성 레이블이 없는 데이터"의 감성 레이블을 예측한 결과는 긍정 또는 부정으로 나뉘는데, 모델이 한쪽 감성에 해당하는 데이터에만 국한된 학습이 되는 것을 막기 위하여 긍정과 부정의 개수를 맞추어 "감성 레이블이 있는 데이터"로 추가하여 업데이트 하고자 하는 방법이다. 그림 6의 두 번째 그림으로서 감성 레이블이 긍정인 데이터가 4개 있으므로 4개를 선택하고, 부정인 데이터 중에서 4개의 데이터를 선택한다.

세 번째 정책으로는 최대 개수를 고려하는 방법이다. 한 번에 많은 양의 데이터가 추가 되는 것을 방지하고 상위 몇%만 선택하기 위해서, 선택되는 데이터의 개수를 제한하여 업데이트하고자 하였다. 그림 6의 세 번째 그림이며, 최대 개수를 5로 정하였다. 부정을 나타내는 데이터는 5개 이상이여서 5개까지 선택하며, 긍정을 나타내는 데이터는 4개뿐이다. 이 경우 4개 모두 선택되며, 최대 개수 5개 이하라는 조건이 만족한다.

위의 3가지 정책에 의해 업데이트 되는 데이터의 개수가 상이하게 달라지며, 실험에서는 이러한 3가지 방법을 적용/미적용에 따라 8번의 실험에 의해 성능을 비교하였다. 제안 기법은 self-training 알고리즘에 적용하여 실험하였다. 그림7은 제안하는 방법론의 처리 프로세스를 보여준다.

1단계로 수집된 각각의 트윗을 전처리한다. 트윗에 포함된 단어들의 분포를 보고 감성을 분류하기 위해 특수문자와 URL을 제거한다. 그리고 같은 단어지만 대/소문자 차이로 다른 단어로 인식할 수 있기 때문에 대문자를 소문자로 치환하는 작업을 한다. 또한 a/an/the 등의 의미 없는 단어인 불용어도 제외하였다.

2단계에서는 감성 레이블의 존재 여부에 따라 "감성 레이블이 있는 데이터" 와 "감성 레이블이 없는 데이터"로 분류한다. 여기서 "감성 레이블이 있는 데이터"는 실제 긍정 또는 부정의 감성 레이블이 매겨진 트윗들로 구성된다.

3단계에서는 "감성 레이블이 있는 데이터"를 사용하여 감성 분석 모델을 생성한다. 감성 분석 모델은 기계학습 기법인 SVM 을 이용하여 생성한다.

4단계에서는 3단계에서 생성된 감성 분석 모델을 이용하여 "감성 레이블이 없는 데이터"의 긍정 또는 부정의

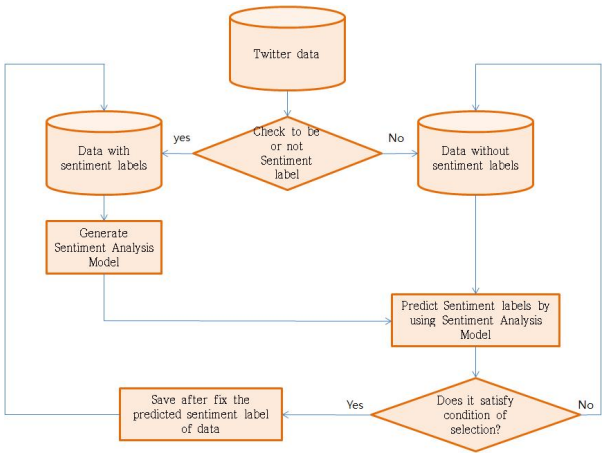

그림 7. 제안하는 방법론의 처리프로세스.
Fig. 7. Process of the proposed approach.

감성 레이블을 예측한다.

5단계에서는 "감성 레이블이 없는 데이터"의 감성 레이블을 예측한 결과 긍정 또는 부정이 확실한 데이터를 선별하고자 위에서 제안한 3가지 정책을 적용한다. 첫 번째 정책은 임계치를 적용하는 것이다. 두 번째 정책은 개수를 고려하여 같은 개수를 업데이트 하고자 하는 것이다. 세 번째 정책도 마찬가지로, 개수를 고려하여 업데이트 할 수 있는 데이터의 양을 제한하는 것이다.

6단계에서는 추가된 "감성 레이블이 있는 데이터"를 이용하여 3~5단계를 반복한다. 이 실험의 종료조건은 "감성 레이블이 없는 데이터"가 모두 선택조건에 만족되
지 못하여 "감성 레이블이 있는 데이터"의 개수가 늘어나지 않았을 경우이다.

생성된 모델은 별도의 "감성 레이블이 있는 데이터"로 검증한다. 성능지표는 Accuracy, Recall, Precision, F-measure를 사용한다. 성능지표는 Accuracy는 정답을 맞춘 것들의 비율이고, Precision은 결과가 나올 것이라 예측한 값 중에 실제 정답의 비율을 나타낸다. 그리고 Recall은 정답이라고 한 것 중에 실제 정답의 비율을 나타낸다. F-measure는 Precision과 Recall을 이용한 식 (2*Precision*Recall) / (Precision+Recall)로 나타내며, Precision과 Recall이 모두 높은 모델을 선택하고자 F-measure값을 사용하여 성능을 비교하였다.

5. 실 험

트윗 데이터는 2009년 4월 6일부터 6월 25일까지 수집된 Stanford data set을 사용하였다. 긍정과 부정으로 분류되어 있는 트위터 데이터 셋으로, Stanford data set중 학습 데이터에서 긍정 5,000개, 부정 5,000개의 데이터를 선택하였다. 선택된 학습 데이터는 "감성 레이블이 있는 데이터" 200개와 "감성 레이블이 없는 데이터" 9,800개로 구성하였으며, 학습에서 생성된 모델은 Stanford data set중 테스트 데이터에 해당하는 "감성 레이블이 있는 데이터" 498개의 데이터 중에서 중립에 해당하는 데이터를 제외한 359개의 데이터로 검증하였

다. 트윗은 선택된 학습 데이터 10,000개의 트윗에서 2번 이상 나타난 단어들을 추출하여 5,839개의 특징벡터로 표현하였으며, "감성 레이블이 있는 데이터"의 각각의 트윗들은 긍정 또는 부정의 감성 레이블을 갖는다.

표 1. 실험 별 성능.
Table 1. Performance by experiment.

policy			Performance indicator			
T	S	M	Accuracy	Precision	Recall	F-measure
X	X	X	59.3%	61.4%	53.3%	57.1%
X	X	O	61.0%	62.4%	58.2%	60.2%
X	O	X	61.6%	63.4%	57.1%	60.1%
X	O	O	61.6%	62.2%	61.5%	61.9%
O	X	X	58.2%	60.5%	50.6%	55.1%
O	X	O	59.6%	61.5%	54.4%	57.7%
O	O	X	61.6%	62.8%	59.3%	61.0%
O	O	O	61.0%	61.4%	62.1%	61.7%
Baseline			57.7%	58.8%	55.0%	56.8%

표 1에서는 Baseline과 임계치, 같은 개수, 최대 개수 유무에 따른 8개의 실험에 대한 성능을 표로 나타내었다. Baseline은 기존의 200개의 "감성 레이블이 있는 데이터"만 이용해 모델을 생성하였을 때의 성능지표인 Accuracy, Precision, Recall, F-measure 값을 나타낸다. 8개의 실험은 200개의 "감성 레이블이 있는 데이터"와 9,800개의 "감성 레이블이 없는 데이터"를 이용해 학습한 결과를 나타낸다.

"감성 레이블이 없는 데이터"에 감성 레이블을 부여한 후, 이 중에서 어떤 데이터를 "감성 레이블이 있는 데이터"에 추가할 것인가를 결정할 때, 논문에서 제시한 3가지 정책을 선택적으로 사용하였다. 즉, 임계치(**T**), 같은 개수(**S**), 최대 개수(**M**)를 적용 유무를 O와 X로 표시하였다. 표 1에서 첫 번째 실험은 **T**=X, **S**=X, **M**=X 인데, 이것은 임계치(**T**) 정책을 적용하지 않고, 같은 개수(**S**) 정책을 적용하지 않으며, 최대 개수(**M**) 정책을 적용하지 않았을 때의 성능을 나타낸다. 이 경우 Accuracy는 59.3%, Precision은 61.4%, Recall은 53.3%, F-measure는 57.1%의 성능을 보인다. 두 번째 실험(**T**=X, **S**=X, **M**=O)은 임계치와 같은 개수 정책은 적용하지 않고, 최대 개수 정책만을 적용하여 실험한 결과를 나타낸다. 본 실험에서, 첫 번째 정책의 임계치는 0.5로 두었으며 세 번째 정책의 최대 개수는 1,000개로 두었다. 이러한 8개의 실험은 성능 지표인 F-measure 값이 Baseline보다 대체로 높아졌다.

표 2에서는 임계치, 같은 개수, 최대 개수의 각 정책 별로 적용 유무에 따른 성능 지표를 나타낸다. 표 1의 Baseline을 제외한 8개의 실험에 대해서 같은 정책을 적용한 것끼리 모아서 평균 낸 것을 보여준다. 'Threshold: X' 란 표 1의 8개 실험에서 임계치(**T**)를 적용하지 않은 실험에 해당하는 4개의 실험의 평균을 나타낸다.

첫 번째는 임계치 정책의 적용 유무에 따른 성능의 변화를 나타낸다. 임계치를 적용하지 않은 Threshold: X'의 F-measure보다 임계치를 적용한 'Threshold: O'의 F-measure가 0.9% 하락하는 결과를 보였다. 이는 임계치는 모든 데이터에서 효과를 나타내지 않음을 보여준다.

표 2. 성능 비교.
Table 2. Comparison of performance.

	Accuracy	Precision	Recall	F-measure
Threshold: X	60.9%	62.3%	57.6%	59.9%
Threshold: O	60.1%	61.6%	56.6%	59.0%
Same number: X	59.5%	61.4%	54.1%	57.5%
Same number: O	61.4%	62.5%	60.0%	61.2%
Max number: X	60.2%	62.0%	55.1%	58.4%
Max number: O	60.8%	61.9%	59.1%	60.4%
Baseline	57.7%	58.8%	55.0%	56.8%

두 번째는 같은 개수 정책의 적용 유무에 따른 성능의 변화를 나타낸다. 같은 개수를 업데이트 하고자 하는 정책인 'Same number: O'의 F-measure는 같은 개수를 고려하지 않은 정책인 'Same number: X'의 F-measure에 비해 3.7%의 성능 향상을 보였으며, 이는 긍정과 부정의 균형을 맞추어 업데이트하여 이루어진 결과로 보인다.

세 번째는 최대 개수 정책의 적용 유무에 따른 성능의 변화를 나타낸다. 최대 개수를 업데이트 하고자 하는 정책인 'Max number: O'의 F-measure는 최대 개수를 고려하지 않은 정책인 'Max number: X'의 F-measure에 비해 2.0%의 성능 향상을 보였으며, 이는 한 번에 많은 개수의 데이터가 업데이트 되는 것보다 학습의 효과가 좋다는 것을 보여준다.

데이터를 선택하기 위한 3가지 정책을 적용했을 때의 F-measure는 각각 59.0%, 61.2%, 60.4%로 Baseline의 F-measure인 56.8%보다 높은 결과를 도출하였다. 임계치와 같은 개수, 최대 개수를 적용하여 self-training 알고리즘에 적용한 결과는 Baseline보다 좋은 결과를 보였다.

6. 결 론

기계학습 기법으로 감정을 분류하는 모델을 만들기 위해서는 "감성 레이블이 있는 데이터"들이 필요하다. 방대한 양의 데이터로 모델을 생성하기 위해서 모든 데이터에 사람이 일일이 감성 레이블을 매길 수 없다. 그래서 "감성 레이블이 있는 데이터"를 활용해 "감성 레이블이 없는 데이터"에 감성 레이블을 매겨 사용하기 위해 self-training 알고리즘을 사용하였다. 또한 감성분석모델의 성능을 높이기 위하여 생성된 감성 분석 모델에 의해 예측된 결과에 따라 "감성 레이블이 없는 데이터"를 "감성 레이블이 있는 데이터"로 추가하고자 하는 3가지 정책을 제시하고 검증하였다. 임계치, 같은 개수, 최대 개수의 측면에서 업데이트할 데이터를 선택하여 감성 분석 모델을 업데이트한 결과, 200개의 "감성 레

이블이 있는 데이터" 만을 이용하여 감성 분석 모델을 생성한 Baseline의 성능보다 향상되었다.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," *In Proceeding of the ACL-02 conference on Empirical methods in natural language processing*. Volume 10. Association for Computational Linguistics, pp. 79-86, 2002.

[2] H. H. Kang, S. J. Yoo, and D. I. Han, "Design and Implementation of System for Classifying Review of Product Attribute to Positive/Negative," *In proceeding of The 36th KIISE Fall Conference*, vol. 36, no. 2, pp. 1-6, 2009.

[3] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," *In Proceeding of the Workshop on Languages in Social Media. Association for Computational Linguistics*, pp.30-38. 2011,

[4] I. S. Kang, "A Comparative Study on Using SentiWordNet for English Twitter Sentiment Analysis," *Journal of The Korean Institute of Intelligent System*, vol. 23, no. 4, pp. 384-388, 2013.

[5] A. Hogenboom, D. Bal, F. Frasincar, M. Bal, F. de Jong, and U. Kaymak, "Exploiting Emoticons in Sentiment Analysis," *In Proceeding of the 28th Annual ACM Symposium on Applied Computing ACM*, pp. 703-710, 2013.

[6] J. H. Yeon, D. J. Lee, J. H. Shim, and S. G. Lee, "Product Review Data and Sentiment Analytical Processing Modeling," *The Journal of Society for e-Business Studies*, vol. 16, no. 4, pp. 125-137, 2011.

[7] H. J. Yune, H. J. Kim, and J. Y. Chang, "An Eficient Search Method of Product Reviews using Opinion Mining Technique," *The Journal of KIISE*, vol. 16, no. 2, pp. 222-226, 2010.

[8] C. CORTES, V. VAPNIK, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.

[9] K. M. Kim, J. D. Lee, and J. H. Lee, "Sentiment Classification using Extracted Rationale Words by Genetic Algorithm," *In Proceeding of the 14th International Symposium on Advanced Intelligent Systems*, pp. 36-43, 2013.

[10] H. G. Yeom, S. M. Park, J. J. Park, and K. B. Sim, "Superiority Demonstration of Variance-Considered Machines by Comparing Error Rate with Support Vector Machines," *International Journal of Control, Automation, and Systems*, vol. 9, no. 3, pp. 595-600, 2011.

[11] H. J. Lee, H. J. Shin, S. Z. Cho, and D. MacLachlan, "Semi-supervised response modeling," *Journal of Interactive Marketing*, vol. 24, no. 1, pp. 42-54, 2010.

[12] K. Soranaka, M. Matsushita, "Relationship Between Emotional Words and Emoticons in Tweets," *In Proceeding of Technologies and Application of Artificial Intelligence*, pp.262-265, 2012.

[13] C. Li, K. Liu, and H. Wang, "The incremental learning algorithm with support vector machine based on hyperplane-distance," *Applied Intelligence*, pp.19-27, 2011.

[14] Yun, "Evolution of big data - The future of IT services to resemble a human," Available: http://cfono1.tistory.com/704, 2013, [Accessed: August 1, 2014].

저 자 소 개

홍소라 (Sola Hong)
2011년: 백석대학교 정보보호학과 공학사
2011년~ 현재: 성균관대학교 전자전기컴퓨터공학과 석사과정

관심분야 : 기계학습, 데이터 마이닝
Phone : +82-31-290-7987
E-mail: plancute@skku.edu

정연오 (Yeounoh Chung)
2008년: Cornell Univ. 전자공학과 학사
2009년: Cornell Univ. 전산학과 석사
2012년~2014년: 성균관대학교 컴퓨터공학과 연구원

관심분야 : 통계학습이론, 기계학습
Phone : +82-31-290-7987
E-mail: yeounohster@gmail.com

이지형 (Jee-Hyong Lee)
1993년: 한국과학기술원 전산학과 학사
1995년: 한국과학기술원 전산학과 석사
1999년: 한국과학기술원 전산학과 박사
2002년~현재: 성균관대 정보통신공학부 교수

관심분야: 지능시스템, 기계학습, 데이터마이닝
Phone : +82-31-290-7154
E-mail: john@skku.edu