# 의견 어구 추출을 위한 생성 모델과 분류 모델을 결합한 부분 지도 학습 방법

남상협 O, 나승훈, 이예하, 이용훈, 김준기, 이종혁

포항공과대학교 전자컴퓨터공학부 컴퓨터공학과

{namsang O, nsh1979, sion, yhlee95, jhlee}@postech.ac.kr

# Semi-Supervised Learning for Sentiment Phrase Extraction by Combining Generative Model and Discriminative Model

Sang-Hyob Nam O  Seung-Hoon Na  Yeha Lee

Yong-Hun Lee Jungi Kim  Jong-Hyeok Lee

Department of Computer Science and Engineering
Division of Electrical and Computer Engineering
Pohang University of Science and Technology

**요 약**

의견(Opinion) 분석은 도전적인 분야로 언어 자원 구축, 문서의 Sentiment 분류, 문장 내의 의견 어구 추출 등의 다양한 문제를 다룬다. 이 중 의견 어구 추출문제는 단순히 문장이나 문서 단위로 분류하는 수준을 뛰어 넘는 문장 내 의견 어구를 추출하는 문제로 최근 많은 관심을 받고 있는 연구 주제이다. 그러나 의견 어구 추출에 대한 기존 연구는 문장 내 의견 어구 부분이 태깅(tagging)된 학습 데이터와 의견 어휘 자원을 이용한 지도(Supervised)학습을 이용한 접근이 대부분으로 실제 적용 상의 한계를 갖는다. 본 논문은 문장 내 의견 어구 부분이 태깅된 학습 데이터와 의견 어휘 자원이 없는 환경에서도 문장단위의 극성 정보를 이용하여 의견 어구를 추출하는 부분 지도(Semi-Supervised)학습 방법을 제안한다. 본 논문의 방법은 Baseline 에 비하여 정확률(Precision)은 33%, F-Measure 는 14% 가량 높은 성능을 냈다.

## 1. 서론

인터넷 사용이 점차 활발해 짐에 따라, 많은 사람들이 인터넷에서 블로그, 위키와 같은 매체를 통해서 자신의 의견을 표현하고 있는 추세이다. 또한, 특정한 정보의 가치를 평가할 때, 이러한 다른 사람들이 인터넷상에 올려놓은 의견정보를 참조하고자 하는 수요도 높아지고 있다. 이 때문에, 사용자들의 의견을 자동으로 추출하는 분야에 대한 관심도 높아졌고, 이 분야에 대한 연구도 활발 하게 이루어지고 있다.

의견을 추출하기 위해서는 전체 문서 중에서 의견이 포함된 문서를 일차적으로 분류하고, 의견이 포함된 문서 안에서 다시 의견이 포함된 문장을 추출하고, 추출된 문장 내에서 의견인 부분을 추출하는 단계로 이루어진다.

이 전체 과정에서는 의견 어휘 자원이 사용되고 이 자원을 구축하고자 하는 다양한 시도가 있었다[1,2,3]. 먼저, 의견 어휘 자원 구축 연구에는 형용사의 접속 관계를 사용한 방법 및 PMI 정보를 활용하여 극성을 지닌 단어가 문서 내 같이 등장하는 경향을 이용한 방법들이 제시되었다 [4,5]. 또한, 유사한 극성을 지닌 단어들은 비슷한 주석(gloss) 정보를 지닌다고 가정하고, 주석 정보를 이용하여 극성 단어를 군집화하는 방법도 제안되었다[6].

의견 문서 극성 분류 문제에 대해서는 극성이 분류된 문서 집합을 학습 데이터로 하여, minimum cuts algorithm과 SVM과 같은 클러스터링 또는 기계 학습 방법이 사용되었고 [7,8], 문서 내의 특정 어구와 seed 단어 간의 PMI 정보를 이용하여 어구의 극성을 파악한 후 문서를 분류하는 시도도 있었다[9].

의견 문서나 문장을 분류하고 나서 비교적 최근에 문장 내에서 실제 의견에 해당하는 문자열이나 어구를 추출하는 문제가 새롭게 연구되고 있다[10,11,12]. 그러나, 현재까지 연구된 기존의 연구는 다음의 측면에서 몇 가지 한계를 갖는다.

첫째, 기존의 연구는 의견 어구가 태깅된 학습 데이터에 기반하여, 의견 어구 추출 모델을 학습하는 지도학습방식이다. 그러나, 지도학습방식을 적용하기 위해 필요한 실제 의견성 어구가 태깅된 학습 데이터는 문서단위로 찬반이 부여되는 리뷰 문장과 달리 웹에서 쉽게 얻을 수 없기 때문에, 만만치 않은 데이터 구축 비용이 소요된다는 점에서 적용상 한계를 갖는다. 따라서, 학습 데이터 구축 비용을 최소화하여 의견성

어구를 태깅할 수 있는 부분지도학습 방법 또는 비지도(Unsupervised)학습방법론의 필요성이 제기된다.

둘째, 기존의 방식은 영어권의 SentiWordNet[1]과 같은 의견성 어구에 대한 기본 사전 정보에 기반을 두고 있는데, 이 역시 한국어와 같이 리소스가 없는 경우에는 적용하기 어렵다. 이러한 리소스 구축을 위해서, [13]에서와 같이 리소스가 있는 언어로부터 대상언어로 리소스를 매핑하는 교차 언어 투사(cross-lingual projection) 방법도 고려해볼 수 있겠으나, 투사 방법 과정 상에서 어휘 중의성, 다어절 어휘 매핑 문제등 애매성이 많아, 높은 정확률을 기대하기 힘들다. 따라서, 기본 사전 정보가 없는 경우에도 효과적으로 의견성 어구를 태깅할 수 있는 모델이 연구되어야 한다.

이를 위해서, 본 논문은 웹에서 쉽게 얻을 수 있는 리뷰 문장 데이터만 가지고, 효과적으로 의견성 어구 추출을 위한 부분지도학습 방법을 제안한다. 먼저, 리뷰 문장데이터는 의견성 어구 태깅보다 구축하기가 더 용이할 뿐 아니라, 잘 알려진 도메인의 경우에는 웹에서 쉽게 얻을 수 있기 때문에, 실제 어구 태깅된 학습 데이터에 대한 효과적인 대안책이라 할 수 있겠다. 특히 리뷰 데이터의 경우 극성까지 부착되어 있다는 점에서, 추출된 어구에 극성 정보까지 분류하고자 하는 우리의 목적에 적합하며, 또한, 리뷰 데이터는 SentiWordNet과 같은 의견 어휘 자원을 사용하지 않고도, 기본적인 통계적 리소스를 구축하기 위한 자료로 활용될 수 있을 것이다.

결국, 우리는 리뷰 문장 데이터로부터, 실제 해당 단어열이 의견성인지 아닌지를 판별하는 추출 모델을 개발하고자 한다. 제안 추출 모델은 생성 방법(Generative Model)과 분류 방법(Discriminative Model)의 하이브리드 방법(Hybrid Model)으로, 생성 모델 (HMM)로 초기에 어구 추출을 수행하고, 추출된 어구는 분류 모델 (CRF)에 학습 데이터로 활용이 될 수 있도록 하여, 최종적으로 학습된 분류 모델을 얻는 방법이다. 이러한 하이브리드 방식은 분류 모델의 높은 정확률과 생성 모델의 높은 재현율(Recall)을 장점으로 취해, 실제 추출을 위한 학습데이터가 없이 리뷰 데이터만 가지고도 고성능 추출 모델을 얻을 수 있게 해 준다. 실제 실험 결과, 본 하이브리드 모델은 HMM 모델만 사용한 방법보다 Precision 은 9.56%, F-Measure 는 0.87% 높은 성능을 보였고, CRF 모델만 사용한 방법 보다 Precision 은 5.51% 떨어졌지만 Recall 37.35%, F-Measure 37.37% 더 높은 성능을 보였다.

## 2. 접근 방법

2.1 에서 설명하는 바와 같이 극성 정보가 부착된 전체 문장을 이용하여 각 어구의 극성 정보를 예측한다. 예측한 어구의 극성 정보를 이용하여 그림 1 과 같이 각 문장 내 어구의 극성을 태깅한 후 HMM 학습에 이용하고, 이 HMM 모델을 이용하여 그림 2 와 같은 과정을 거쳐 좀더 나은 모델을 만들게 된다. 이 모델을 사용하여 그림 3과 같이 극성이 태깅되 어구 학습데이터를 만든 후 CRF 모델을 학습하게 된다.

### 2.1 어구의 극성 정보 예측

본 논문의 실험 데이터는 1~10점 사이의 점수가 있으며, 10점에 가까울수록 긍정이다.

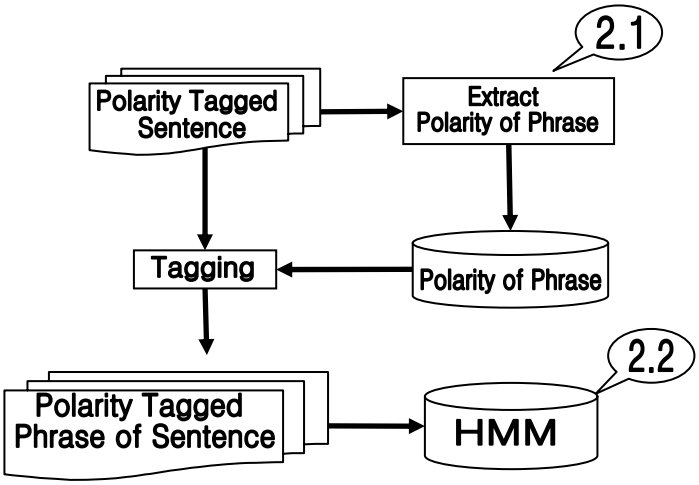


그림 1. 문장 극성 정보로부터 어구의 극성 정보 예측 및 이를 이용한 HMM 모델 학습

$$Pos = \{s_7, s_8, s_9, s_{10}\}, Neg = \{s_1, s_2, s_3, s_4\} \quad (1)$$

$$N = Pos \cup Neg \quad (2)$$

$$P(s_i \mid t) = \frac{P(s_i, t)}{P(t)} = \frac{Freq(t \in s_i)}{Freq(t)} \quad (3)$$

$$S(s_i) = \begin{cases} 5 - s_i \ (s_i \in Neg) \\ s_i - 6 \ (s_i \in Pos) \end{cases} \quad (4)$$

$$SO(t) = \sum_{s_i \in Pos} S(s_i) \times P(s_i \mid t) - \sum_{s_i \in Neg} S(s_i) \times P(s_i \mid t) \quad (5)$$

(1)에서는 1~10 점의 점수로 나누어져서 그룹을 이루는 영화 리뷰 문장 집합 $s_1, s_2, s_3, s_4, s_7, s_8, s_9, s_{10}$ 들이 긍정(Pos), 부정(Neg) 집합에 포함됨을 보여준다. (2)에서 N 은 전체 문장 집합을 의미하고 이 전체 문장 집합은 Pos 집합과 Neg 집합으로 구성된다. (3)에서 $P(s_i \mid t)$ 는 어구(t) 가 주어졌을 때 해당 어구가 특정 점수 문장 집합에서 나타날 확률이다. (4)의 $S(p_i)$ 는 Scaling Factor로서 의견의 강도를 표현하기 위한 값이다. 예를 들어서 가장 강한 긍정적, 부정적인 $s_{10}, s_1$ 문장 집합은 가장 높은 값인 4점을 지니고, 가장 약한 긍정적, 부정적인 $s_7, s_4$ 문장 집합은 1점을

[1] http://sentiwordnet.isti.cnr.it

지니게 된다.

(5) 의 SO(Semantic Orientation) 는 해당 어구(t) 의 극성을 의미한다. 가장 긍정적인 어구는 4점, 가장 부정적인 어구는 -4점을 지니게 된다. 이 방법을 이용하여 각 어구의 극성을 구한다. SO 값이 4~1 인 어구는 긍정(positive), 1~-1 은 중립(neutral), -1~-4 는 부정(negative) 어구로 결정 하였다.

자동적으로 생성한 어구의 극성 정보는 HMM 모델의 학습데이터를 생성하는데 사용된다.

긍정, 부정 어구는 문장 문맥에 따라 정 반대의 의미도 지닐 수 있기 때문에 의견 어휘 자원의 극성을 바로 적용하지 않고 문장 극성을 따르도록 할 수 있다. 예를 들어 (1) 에서 Pos 그룹의 문장에서 긍정, 부정 어구는 모두 긍정 어구로 초기화 하고, Neg 그룹의 문장에서 긍정, 부정 어구는 모두 부정으로 초기화 할 수 있다. 이 같이 문장(Sentence)의 극성을 따르는 학습 데이터를 사용한 모델은 모델 명 앞에"S-" 표시를 하였다. 3.6 에서 실제 성능상에 어떤 영향이 있는지 결과를 통해서 비교해 보았다.

### 2.2 Hidden Markov Model

본 논문에서는 의견 어구 추출을 위하여 Hidden Markov Model 을 사용한다. 각 어구들은 HMM 모델에서 observation 에 해당되고, 각 observation 은 3가지 상태를 가진다(긍정, 부정, 중립). 어구의 단위 및 예는 3.1에서 자세히 설명하겠다. 우리는 자동적으로 예측한 어구의 극성 정보 자원을 이용하여 어구를 긍정, 부정, 중립으로 태깅 한다. 이때도 해당 문장의 극성에 따라 주관적인 어구(긍정, 부정)에 대하여 그 문장의 극성을 부여하였다. 이렇게 태깅된 데이터를 바탕으로 initial probability, emission probability, transition probability 을 구한다. 이 확률과 Viterbi algorithm 을 이용하여 어구들의 극성 정보를 추론한다. 좀더 나은 language Model 을 만들기 위해서 그림 2 와 같이 Multiple Pass Decoding 을 수행 하였다(MP-HMM).

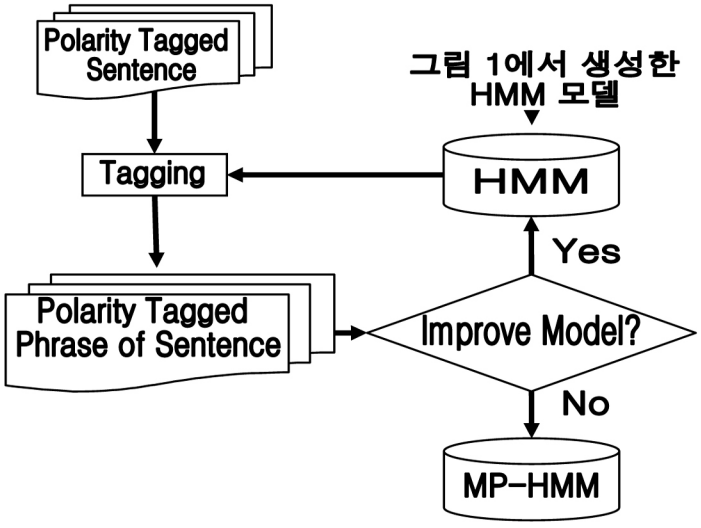


그림 2. Multiple Pass Decoding 을 수행하는 모습

### 2.3 Conditional Random Field

의견 어구를 찾는 방법으로 CRF를 사용한 기존 연구가 있었다[10,11]. 그 연구에서는 의견 어구를 찾는 문제를 태깅 문제로 바라 보았고 본 논문에서도 의견 어구와 그 극성을 찾는 문제를 마찬가지로 태깅 문제로 바라보았다.

$x = x_1 x_2 ... x_n$ 이라는 순차적인 어구들이 존재하는 경우에 $y = y_1 y_2 ... y_n$ 라는 극성 태그를 생성하게 된다. 어구의 단위 및 설명은 3.1에서 하겠다.

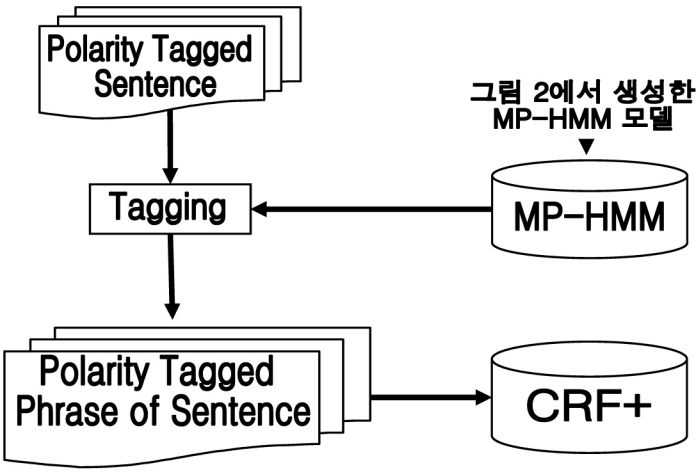


그림 3. CRF+ 모델의 학습 과정

이때 극성 태그는 3가지 종류 긍정, 부정, 중립을 가지게 된다. CRF에 대한 자세한 설명은 [14]에서 볼 수 있다. 우리의 순차적인 어구 태깅 문제에서, $G = (V, E)$ 이고 여기서 V 는 random variable $Y = \{Y_i \mid 1 \le i \le n\}$ 집합으로서 입력 문장에 존재하는 n 개 어구의 극성에 해당되는 변수이다. $E = \{(Y_{i-1}, Y_i \mid 1 < i \le n\}$ 은 linear chain 을 형성하는 $n-1$ 개의 edge 이다.

$$\exp(\sum_{k=1}^{K} \lambda_k f_k(y_{i-1}, y_i, x)) \ (8) \quad \exp(\sum_{k=1}^{K'} \lambda_k' f_k'(y_i, x)) \ (9)$$

문장 $x$ 에서 각 node에 대하여 non-negative clique potential (8)과 edge 에 대하여 (9)를 정의한다. 각 node에 대하여 $f_k(...)$ 는 binary feature indicator 함수 이고, $\lambda_k$ 는 각 feature 함수에 할당되는 weight 이고, $K$ 와 $K'$ 는 edge 와 node에 각각 정의된 feature 수 이다. [14]에 따라서, 주어진 $x$ 토큰들에 대하여

순차적인 극성 정보 $y$ 가 가지는 조건 확률은 다음과 같이 된다.

$$P(y \mid x) = \frac{1}{Z_x} \exp(\sum_{i,k} \lambda_k f_k(y_{i-1}, y_i, x) + \sum_{i,k} \lambda_k' f_k'(y_i, x))$$

$$Z_x = \sum_{y} \exp(\sum_{i,k} \lambda_k f_k(y_{i-1}, y_i, x) + \sum_{i,k} \lambda_k' f_k'(y_i, x))$$

여기서 $Z_x$ 는 각각의 $x$ 에 대하여 normalization 상수이다. 주어진 학습 데이터 $D$ 에 대하여 문장내의 각 어구들은 그것에 해당되는 긍정, 부정, 중립 극성과 짝을 이루게 된다. 파라미터들은 conditional log-likelihood $\prod_{(x,y)\in D} P(y \mid x)$ 를 최대화 하도록 학습된다. 학습된 모델로 원하는 극성 태깅 정보를 추론하기 위해서 주어진 문장의 각 어구 $x$ 에 대하여 순차적인 태깅 결과 $y$ 는 $\arg\max_{y'} P(y' \mid x)$ 을 만족하는 값으로 추론하게 된다. CRF 는 이미 구현된 Toolkit[2]을 사용할 수 있다.

위 CRF 방법은 Supervised 학습 방법이기 때문에 정확하지 않고 예측 값인 2.1 에서 생성한 어구의 극성 정보만으로는 만족할 만한 성능을 내기 어렵다. 이러한 한계를 극복하기 위하여 그림 3 과 같이 우리는 CRF의 학습 데이터를 생성하기 위하여 Multiple Pass Decoding 을 수행한 HMM 모델(MP-HMM)을 사용한다(CRF+).

## 3. 실험

### 3.1 어구의 구분 단위

어구 구분 단위 선택은 형태소 분석이나 N-Gram 모델을 사용할 수 있으며, 본 논문에서는 인터넷 과 같이 정확하게 표기되지 않은 많은 문장을 포함하는 데이터를 대상으로 의견 어구를 추출하기 위해서 N-Gram 모델을 선택하였다. Character N-Gram 모델이 한국어 어구의 분할에 적절하기 때문에 본 논문에서 제시하는 N-Gram 모델은 Character N-Gram 을 사용하였다. 어구의 극성을 부여 하는 단위는 모두 Bigram으로 부여하였으며, 이 Bigram에 극성을 부여하는 자질은 다음과 같은 자질을 사용하였다.

N-Gram 모델을 이용하여 의견 어구의 자질을 선택하는 경우에는 문서의 극성을 기계 학습 방법으로 분류하였던 연구를[7][8] 통해서 Unigram 과 Bigram 이 좋은 자질임을 알 수 있고, 이는 의견 어구를 구분하는 자질로도 사용될 수 있다.

Character Unigram 은 일반적인 Unigram 과 달리 독자적으로 의미를 지니기 어렵기 대문에 본 논문에서는 Character Unigram 자질은 사용하지 않았고 Character Bigram, Character Trigram 을 사용하였다.

"$c_i c_{i+1} c_{i+2}$ $c_j c_{j+1} c_{j+2}$ (6)" "$c_{i+1} c_{i+2}$ $c_j c_{j+1}$ (7)"

이 외에도 (6)과 같은 어구에서 (7)과 같은 자질을 Bigram*으로 정의하여 사용하였다. 예를 들면 "이렇게 재미있는" 이라는 어구에서 "렇게 재미" 라는 어구를 의미한다. 실제 실험에서도 Bigram* 의 자질이 성능 향상에 Trigram보다 더 큰 향상을 가져왔다.

이러한 N-Gram 의 연속이 모여서 하나의 의견 어구를 형성하게 된다.

### 3.2 학습 데이터

본 실험에서 사용하는 데이터는 1~10 점으로 평가된 영화 평가 문장들이다. 이 실험 데이터는 http://movie.naver.com 에서 추출한 데이터로서 사용자들에 의해서 실제로 평가된 문장들이다. 그 중에서 각 어구들의 극성을 학습하는 데이터로는 부정에 가까운 1,2,3,4 점의 영화평과 긍정에 가까운 7,8,9,10 에 가까운 영화평 데이터만 사용하였다. 이 데이터는 각 2000 문장씩 총 16000 문장으로 이루어져 있다. 이 데이터는 오류가 일부 존재할 수 있는 실제 환경의 데이터이다. 의견 어구 태깅 학습 데이터 생성에서는 극성이 뚜렷하지 않은 4,7 점 점수 문장들은 제외 하고, 1,2,3점의 부정 문장들과 8,9,10점의 긍정 문장들 사용하여 의견 어구에 극성이 부여된 학습 데이터를 생성한다. HMM, CRF 모델은 이 데이터를 이용하여 학습을 하게 된다.

### 3.3 평가 데이터

평가 데이터는 http://movie.naver.com 에서 추출하였으며, 학습 데이터와는 다른 데이터로서 여기서도 극성이 좀 더 명확한 1,2,3,8,9,10점의 점수만을 사용한다. 각 300문장씩 총 1800개의 문장이다. 2명의 대학원 생이 각 문장 내에서 긍정, 부정, 중립 부분을 Bigram 단위로 태깅하였다. 주석자에 따라서 "스릴 넘치는" 부분을 긍정 어구로 표시하거나, "스릴넘치는" 부분을 긍정 어구로 표시하였다. 해당 문자열의 Bigram 표현인 "스릴 넘치 치는" 과 "스릴 릴넘 넘치 치는"을 직접적으로 비교하게 되면 부분적으로 일치하는 태깅도 일치하지 않는 것으로 판단되는 어려움이 따른다. 따라서 직접적으로 일치도를 계산하는 대신에 우리는 CRF모델을 사용하여 주석자가 단 데이터의 일치도를 계산 하였고, 동시에 주석자가 얼마나 일관되게 주석을 달았는지 일관성 정도도 계산하였다.

주석자의 데이터가 얼마나 일관성이 있는지 여부는 각 주석자들이 태깅을 한 데이터를 각자 자신의 CRF 모델에 적용한 결과를 통해서 알 수 있다. 정확률, 재현률이 긍정적/부정적 어구 및 주관적 어구 구분 모두에서 95%~100%로서 충분히 일관성을

[2] 본 논문에서는 CRF Toolkit으로 CRF++ 을 사용하였다. http://crfpp.sourceforge.net
–c 파라미터는 1로 정하였다.

지니고 있었다. 주석자들의 주석 데이터를 상대 주석자들의 CRF 모델에 테스트한 결과는 긍정적/부정적 어구 구분에서는 재현율이 76~83%, 정확률이 85~88%에 이르렀고, 주관적 어구 구분에서는 재현율이 76~83%, 정확률이 85~89%에 이르렀다. 이는 주석자들이 공통적으로 각 어구의 극성에 대한 감각을 공유하고 있음을 보여주고, 테스트 데이터로 사용하기에 적절함을 알 수 있다. 본 논문에서는 그 두 명의 주석자가 만든 정답 데이터 중 하나를 선택하여 정답으로 사용하였다.

### 3.4 평가

정확률, 재현율, F-measure를 사용하여 본 논문의 성능을 측정하였다. 평가대상은 중립 극성을 제외한 긍정, 부정 어구이다. 극성 데이터는 모두 Character 단위로 부여 되었다.

시스템은 "대단한 수작"이라는 구절에서 "대단한"라는 구절만 찾는 경우도 있다. 이 같은 경우도 감안하여 좀더 완화된 방식의 평가를 하였다 하였다[11]. 이 평가는 Overlap으로 지칭하였다.

### 3.5 Baseline

Baseline으로는 임의로 어구의 극성을 선택하는 모델을 사용하였다. 각 문장의 어구에 대해서 긍정, 부정, 중립 극성을 임의로 부여하게 된다. 임의로 극성을 선택할 때에는 보통 극성이 띄어쓰기를 기준으로 하여 달라지기 때문에 띄어쓰기를 기준으로 하여 극성을 임의로 선택한다. 주관성(Subjectivity)을 구분하는 baseline 은 위 baseline 의 긍정, 부정 극성을 주관성으로 묶어서 baseline으로 사용하였다.

### 3.6 Results

2.1에서 설명한 문장(Sentence)의 극성을 이용한 모델들은 모두 모델명 앞에 "S-"를 붙여 표시하였다.

| | Overlap | | |
|---|---|---|---|
| Method | Rec | Pre | F |
| S-Bigram | 46.40 | 54.77 | 50.24 |
| S-Trigram | 23.58 | **63.00** | 34.31 |
| S-Bigram* | 19.96 | 60.01 | 29.96 |
| S-Bigram+Trigram | 49.30 | 54.44 | 51.75 |
| S-Bigram+Bigram* | 52.59 | 54.35 | 53.46 |
| S-All | **54.86** | 54.25 | **54.55** |

표 1. N-Gram 성능 비교

표 1 을 보면 Bigram이 성능에는 가장 큰 영향을 끼치고 있음을 볼 수 있으며, 그 외에 Bigram*과 Trigram 이 Bigram의 recall 을 향상 시키는데 도움을 주고 있음을 확인 할 수 있다. 이는 Bigram*과 Trigram이 Bigram보다 더 넓은 범위의 어구를 반영하기 때문에 Bigram 만으로 극성을 여부를 판단하지 못하는 부분에 대해서 극성을 판단하기 때문이다. 특히 흥미로운 점은 Bigram*이 Trigram 보다 Bigram의 recall 성능 향상에 더 기여를 한다는 점이다. 이는 한국어의 특성상 Bigram으로 반영 하지 못하는 부분을 Bigram*이 Trigram 보다 좀더 잘 반영하기 때문이다.

표 2 에서 SO resource는 2.1 에서 구축한 어구의 극성 정보로 의견 어구를 추출하는 모델이다.

| | Overlap | | |
|---|---|---|---|
| Method | Rec | Pre | F |
| Baseline | 52.63 | 36.79 | 43.30 |
| S-All(SO resource) | **54.86** | 54.25 | 54.55 |
| S-HMM | 34.95 | 72.11 | 47.08 |
| S-MP-HMM | 53.47 | 59.73 | 56.43 |
| S-CRF | 11.49 | **74.80** | 19.93 |
| CRF+ | 48.55 | 67.74 | 56.56 |
| S-CRF+ | 48.84 | 69.29 | **57.30** |

표 2. 여러 모델에서 긍정, 부정 어구 추출 성능 비교

S-HMM은 이 SO resource를 사용하여 어구의 극성을 초기화한 학습데이터를 이용하여 생성된 모델이며, S-MP-HMM 는 S-HMM 모델이 Multiple Pass Decoding을 수행한 모델이다. S-CRF 는 SO resource로 어구의 극성을 초기화한 학습 데이터를 사용한 모델이고, CRF+, S-CRF+은 S-MP-HMM으로 초기화한 학습 데이터를 사용한 모델이다.

S-MP-HMM, CRF+, S-CRF+ 는 의견 어휘 자원만을 사용한 모델보다 성능을 향상 시켰으며, 특히 정확률에서 성능을 크게 향상 시켰다. S-CRF+ 과 CRF+ 는 Overlap 평가의 정확률, F-Measure 에서 S-HMM, S-MP-HMM, S-CRF 보다 높은 성능을 보였다. 이를 통해서 S-CRF+, CRF+ 가 S-MP-HMM 만을 사용한 모델보다 더 나은 모델이 되었음을 확인할 수 있다.

S-CRF+는 평가에서 CRF+ 보다 높은 성능을 보였는데, 이는 2.1에서 제시한 각 어구의 긍정, 부정 극성 정보는 문장의 극성 정보를 따르게 하였던 가정이 유효함을 보여준다.

| | Overlap | | |
|---|---|---|---|
| Method | Rec | Pre | F |
| Baseline | **77.77** | 54.90 | 64.37 |
| S-All(SO resource) | 60.42 | 59.74 | 60.08 |
| S-HMM | 34.12 | 76.92 | 47.27 |
| S-MP-HMM | 62.95 | 70.32 | **66.43** |
| S-CRF | 12.38 | **80.55** | 21.46 |
| CRF+ | 55.75 | 77.79 | 64.95 |
| S-CRF+ | 55.49 | 78.73 | 65.10 |

표 3. 여러 모델의 주관적 어구 추출 성능 비교

CRF, S-CRF+, CRF+ 의 정확률은 높지만, 재현률은 낮다. 이는 해당 모델들의 학습 데이터가 완전히 정확한 데이터가 아니라 의견 어구의 극성을 예측하는

방법으로 구축된 데이터 이기 때문이다. 의견 어휘 자원만을 사용한 모델의 재현률은 54.86%에 불과 하고, 이 재현률은 CRF 모델들의 성능에 영향을 끼쳤고, 54.25%의 정확률 역시 CRF 모델의 정확률에 영향을 끼쳤다. S-CRF+ 와 CRF+는 이와 같은 재현률이 낮은 약점을 S-MP-HMM 을 통하여 극복하였다.

표 3은 각 모델이 주관적 어구를 찾아 내는 문제에서의 성능을 보여준다. 이 문제는 전체 문장 중에서 의견이 들어 있는 부분을 찾는 문제이다. 즉 긍정, 부정 극성을 지닌 어구를 찾는 문제보다는 좀더 간단한 문제이다. 이 데이터에서 Baseline 의 재현률과 F-measure가 높은 이유는 테스트로 사용한 데이터가 한 문장에 주관적 구절이 하나 이상씩 포함된 문장인 요인이 크게 작용한 결과 이다. 그런데 여기서 흥미로운 점은 이렇게 좀더 간단한 문제임에도 불구하고 긍정, 부정 극성의 어구를 찾는 문제에서의 정확률과의 차이가 10% 이내라는 점이다. 실험 결과로 미루어 보았을 때에 주관적 어구를 구분하는 부분이 전체 의견 어구 추출 성능에서 가장 큰 영향을 끼치고 있음을 알 수 있다.

## 5. 요약 및 결론

본 논문에서는 문장내의 긍정, 부정 어구를 찾기 위한 부분 지도 학습 방법을 제안 한다. 본 논문의 방법은 어구의 극성이 부여된 학습 데이터가 많지 않은 현실을 극복하기 위해 문장 단위로 극성이 부여된 데이터를 이용하여 모델을 학습한다. 이 방법은 언어 독립적이며 문장 단위의 데이터만을 이용하여 모델을 학습하므로 다른 언어(비 영어권 언어)에 쉽게 확장될 수 있다.

제안된 방법을 통해 구축된 의견 어휘 자원이 기존 방법만큼 잘 정제된 자원이 될 수 는 없고 어느 정도 오류를 내포하고 있고 이 자원을 그대로 사용하기에는 무리가 따른다. 따라서 본 논문에서는 자동으로 구축된 어휘 자원을 HMM, CRF 모델을 이용하여 정제하여 사용하는 방법을 제안했다. 우리는 또한 긍정, 부정 어구를 찾아 내는 문제에 있어서 S-CRF, S-HMM, S-MP-HMM 보다 더 나은 성능을 보이는 S-CRF+ 모델을 제안하였다.

본 논문에서는 언어 독립적인 자질만을 이용하였으나, 언어 종속적인 자원과 다양한 자질의 활용은 의견 어휘 추출의 성능을 향상시킬 수 있을 것으로 기대된다.

## 감사의 글

본 논문은 2008년도 두뇌한국21사업의 지원을 받았고 지식경제부 및 정보통신 진흥연구원의 정보통신선도기반기술개발사업의 연구결과로 수행되었습니다.

## Reference

[1] Janyce M. Wiebe, Learning Subjective Adjectives from Corpora. In Proceedings of AAAI, pages 735~740, 2000

[2] Ellen Riloff, Janyce Wiebe, and Theresa Wilson, Learning Subjective Nouns using Extraction Pattern Bootstrapping. In Proceedings of CoNLL, pages 25~32. 2003

[3] Thomas Hofmann, Jan Puzicha and Michael I. Jordan, Unsupervised learning from dyadic data, In Advances in Neural Information Processing Systems,volume 11, 1999

[4] Hatzivassiloglog, vasileios and Kathleen R. McKeown, Predicting the semantic orientationation of Adjectivies, 35nd ACL, pages 174~181. 1997

[5] Turney,Peter D. and Michael L.Littman, Measuring praise and criticism:Inference of semantic orientationation from association, ACM Transactions on Information Systems Vol.21, No.4 pages 315~346 , 2003

[6] Esuli, Andrea and Fabrizio Sebastiani, Determining the semantic orientationation of Terms through Gloss Classification, CIKM, 2005

[7] Pang, Bo and Lillian Lee, A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, 42nd ACL pages 271~278, 2004

[8] Pang, Bo and Lillian Lee, Thumbs up? Sentiment Cl assification using Machine Learning Techniques, In Proceedings of EMNLP, page 79~86. 2002

[9] Turney,Peter D., Measuring praise and criticism:Inference of semantic orientationation from association, ACM Transactions on Information Systems Vol.21, No.4, pages 315--346, 2003

[10] Choi, Yejin and Claire Cardie, Ellen Riloff and Siddharth Patwardhan, Identifying Sources of Opinions with Conditional Random Fields and Extraction Pattern, HLT/EMNLP, page 355~362. 2005

[11] Breck,Eric and Yejin Choi and Claire Cardie, Identifying Expressions of Opinion in Contenxt, In Proceedings of the Twentieth International Join Conference on Artificial Intelligence, 2007

[12] Theresa Wilson, Janyce Wiebe, Paul Hoffmann, Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, HLT/EMNLP, pages 347~354, 2005

[13] Mihalcea, Rada and Carmen Banea, Janyce Wiebe, Learning Multilingual Subjective Language via Cross-Lingual Projections, 45nd ACL, pages 976~983. 2007

[14] Lafferty, John , Andrew McCallum, Fernando Pereira, Conditional Random Fields:Probabilistic Models for Segmenting and Labeling Sequence Data, In Proceedings of 18th International Conference on Machine Learning, 2001