

# 미국 내 한식당 온라인 리뷰의 감성분석을 위한 특징벡터와 분류기법의 비교 연구

장한솔, 유성준\*, 한동일, 김경중  
세종대학교 컴퓨터공학과  
e-mail : [hsjang@sju.ac.kr](mailto:hsjang@sju.ac.kr), [sjyoo@sejong.ac.kr](mailto:sjyoo@sejong.ac.kr)

## Comparative Study of Feature Vectors and Classification Methods for Sentiment Analysis for Online Review of Korean Restaurant in U.S.A.

Hansol Jang, Seongjoon Yoo\*, Dongil Han, Kyungjoong Kim  
Department of Computer Engineering  
Sejong University

### Abstract

In this paper, we did sentiment analysis for online review of Korean restaurant in Yelp using machine learning methods. We adapted various classification algorithms to compare the performance. Also, we changed type and size of feature vectors to compare the performance. We used Adaboost, ANN, Decision Tree, kNN, Naive Bayes, SVM as classification algorithm and extracted feature vectors with PCFG Parser and POS Tagger. On average, SVM with POS Tagger showed better performance.

### I. 서론

자신의 의견을 인터넷 상에 등록하는 것이 대중화됨에 따라, 여러 분야의 다양한 의견들이 인터넷 상에 대규모로 게재되고 있다. 음식점을 방문하고 나서 맛, 서비스, 분위기 등에 대한 의견 또한 트위터나 블로그와 같은 SNS(Social Network Service)와 텡스푼, Yelp와 같은 전문 리뷰 사이트에 활발하게 등록되고 있다. 일반적으로 작성자가 매긴 별점을 통해 온라인에 등록된 음식점에 대한 의견들의 전반적인 분위기를 짐작하게

된다. 하지만, 별점들을 분석해보면 몇 가지 문제점을 발견하게 된다. 1) 대부분의 작성자들은 최악이 아니라면, 2~4개의 별점을 준다. 2) 일부 음식점 주인이 홍보성으로 의견을 작성하면서 5개의 별점을 준다. 3) 광고와 같이 작성자의 의견과 무관한 임의의 별점을 주는 경우도 있다. 따라서, 단순히 별점을 근거로 전반적인 분위기를 짐작하는 것이 아니라, 긍정/부정 분류와 같은 감성 분석을 통해 요약 및 정리를 하게 되면, 전반적인 분위기를 짐작할 수 있게 된다. 이러한 관점으로 음식점 리뷰에 대해 많은 선행 연구들이 있었다. Lee와 Graf[1]는 POS Tagger와 N-gram을 이용하여 감성 분석을 하였다. Zhang과 Ye 등[2]은 중국어로 작성된 리뷰에 대해 N-gram을 이용하여 감성 분석을 하였다. Kang과 Yoo 등[3]은 Senti-lexicon과 향상된 Naive Bayes 알고리즘을 이용하여 감성 분석을 하였다. 기존 연구는 하나의 특징벡터 혹은 하나의 분류 알고리즘으로만 실험하여 특징벡터의 종류 및 분류 알고리즘에 따른 성능의 비교를 할 수 없는 문제가 있어서 우리는 그런 문제를 해결하기 위해 Yelp에 올라와있는 대량의 미국 내 한식당 온라인 리뷰에 대해 다양한 분류 알고리즘을 적용하여 성능을 비교하였다. 아울러, PCFG Parser와 POS Tagger를 이용하여 특징 벡터의 종류 및 사이즈 변화에 따른 분류 성능의 차이도 비교하였다. 혼련/분류 알고리즘으로 Adaboost[4], ANN[5], Decision Tree[6], kNN[7], Naive Bayes[8], SVM[9]을 사용했는데 평균적으로 POS Tagger를 이용하여 특징벡터를 추출하고 SVM으로 분류했을 때의 성능이 우수한 편이라는 결과가 얻어졌다.

\* 교신저자 : 유성준(세종대학교 컴퓨터공학과)

## II. 본론

### 2.1 일반적인 문서 분류

일반적인 문서 분류 실험에서는 그림 1과 같이 먼저 훈련 과정을 통해 모델을 구축하고, 분류 과정에서 얻어진 결과물을 모델에 적용하여 최종 결과물을 도출하게 된다. 이때 사용되는 훈련/분류 알고리즘에는 ANN, Decision Tree, kNN, Naïve Bayes, SVM 등이 있고, 평가 방법에는 10-fold Cross Validation 등이 있다.



그림 1. 일반적인 문서 분류 실험의 순서도

### 2.2 WEKA를 이용한 문서 분류

그림 2와 같이 WEKA[10]를 이용한 문서 분류 실험에서는 일반적인 문서 분류 실험에서 훈련/분류 알고리즘을 개발해야 하는 과정대신 ARFF 포맷으로 변환하여 프로그램에 적용하는 것으로써, 손쉽게 적용하여 결과를 도출할 수 있다. 더불어, 통계 및 분석 결과도 볼 수 있다는 장점이 있다.

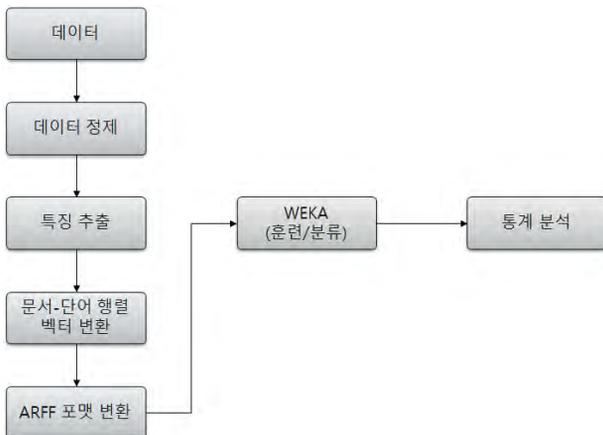


그림 2. WEKA를 이용할 경우 문서 분류 실험의 순서도

## III. 구현

본 논문에서는 일반적인 문서 분류 실험 방법보다 간편하게 결과물을 도출할 수 있는 WEKA를 이용한 문서 분류 실험 방법을 기반으로 그림 3과 같이 분류 실험을 진행하였다.

### 3.1 감성 분석 실험

본 논문에서는 Yelp에 올라와있는 대량의 미국 한식당 온라인 리뷰들을 요약 및 정리하는 관점으로 기계학습 기법을 이용하여 긍정, 부정으로 분류하였다.

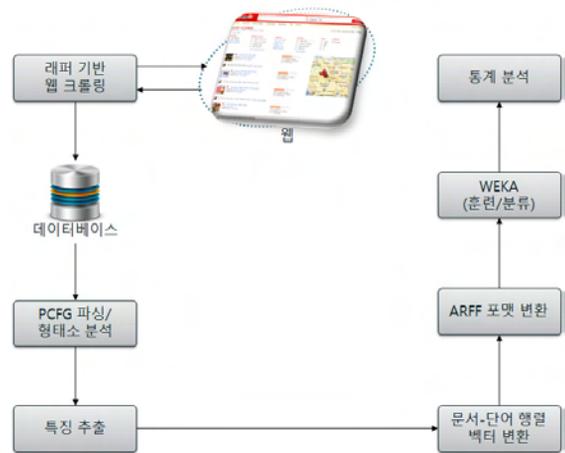


그림 3. 본 논문의 분류 실험 구조도

먼저, 레퍼 기반 크롤러를 이용하여 Yelp에 올라와있는 미국에 있는 한식당 온라인 리뷰 36,535개를 수집하여 데이터베이스에 저장하였다. HTML 태그와 같이 작성자의 의견과 상관없는 내용들을 문서에서 정제하였다. 모델을 생성할 때, 기계학습 기법의 단점인 수작업으로 긍정, 부정으로 분류하는 데 많은 시간이 소요되어, 별점을 분류 기준으로 사용하여 실험하였다. 별점이 3 이상이면 긍정, 그렇지 않으면 부정으로 분류하여, 각 5,000개씩 총 10,000개의 샘플 문서들을 추출하여 실험하였다. 실험은 특징벡터의 종류에 따른 분류의 성능을 보기 위해 총 2세트로 진행이 되었는데, 1세트에서는 PCFG Parser를 이용하여 특징벡터를 추출하여 실험하였고, 2세트에서는 POS Tagger를 이용하여 특징벡터를 추출하여 실험을 하였다.

### 3.2 특징벡터

#### 3.2.1 PCFG Parser를 이용한 특징 추출

Stanford PCFG Parser[11]를 이용하여 Type Dependency를 추출하였다. 추출된 단어들을 빈도수 기반으로 정렬

하여 상위 단어 집합을 생성하였다. 그림 4와 같이 상위 단어 집합을 통합하여 특징벡터로 사용하였는데, 특징벡터의 사이즈 변화에 따른 분류의 성능을 보기 위해, 특징벡터의 사이즈를 3가지로 정하였다. 첫째는, 총 10개의 단어로 이루어진 집합을, 둘째는, 총 20개의 단어로 이루어진 집합을, 마지막으로 총 30개의 단어로 이루어진 집합을 사용하였다. 또한, 특징벡터의 정제 여부에 따른 분류의 성능을 보기 위해, 그림 5와 같이 상위 단어 집합에서 리뷰의 전반적인 분위기를 나타낼 수 있는 단어들만을 추출한 집합을 사용하였다. 샘플로 추출된 문서들을 특징벡터와 비교하여 문서-단어 행렬을 생성하였고, 각 문서에 특징벡터가 포함되는 경우에는 1로 가중치를 주었다. attribute에 특징벡터를 나열하고, data에 문서-단어 행렬을 나열함을 통해 ARFF 포맷으로 변환하여 WEKA를 통해 실험하였다. 다양한 훈련/분류 알고리즘을 사용하여 성능을 비교하기 위하여 6가지 알고리즘 (Adaboost, ANN, Decision Tree, kNN, Naive Bayes, SVM)을 사용하였고, 평가 방법으로는 10-fold Cross Validation을 사용하였다.

```

amod(food, chinese)    amod(Sushi, Best)      amod(night, last)
amod(place, great)    neg(had, never) neg(good, not)
amod(food, good)      neg(, not) amod(food, good)
amod(I've, best)      amod(time, first) amod(restaurant, favorite)
amod(place, Good)     neg(sure, not) amod(soup, Sour)
amod(food, great)     amod(place, Best)     amod(time, last)
amod(place, favorite) amod(food, best)      amod(fan, Big)
amod(, good)          amod(sushi, great)    amod(food, Korean)
amod(Sushi, Good)     amod(thing, only)    amod(times, few)
amod(restaurant, Chinese) amod(places, other) amod(food, Chinese)
    
```

그림 4. 1세트에서 사용한 특징벡터의 예

```

amod(place, great)    amod(place, Best)      amod(hour, happy)
amod(food, good)      amod(food, best)      amod(service, great)
amod(I've, best)      amod(sushi, great)    neg(go, not)
amod(place, Good)     neg(good, not) amod(service, good)
amod(food, great)     amod(food, good)      amod(service, friendly)
amod(place, favorite) amod(restaurant, favorite) neg(worth, not)
amod(, good)          neg(bad, not) amod(food, Great)
amod(Sushi, Good)     amod(place, nice)      amod(restaurant, best)
amod(Sushi, Best)     amod(service, Good)    neg(go, never)
neg(had, never) amod(spot, great)      amod(food, decent)
    
```

그림 5. 1세트에서 사용한 정제된 특징벡터의 예

### 3.2.2 POS Tagger를 이용한 특징 추출

Stanford POS Tagger[12]를 이용하여 형용사를 추출하였다. 1세트와 마찬가지로 그림 6과 같이 상위 단어 집합을 통합하여 특징벡터로 사용하였다. 또한, 특징벡터의 정제 여부에 따른 분류의 성능을 보기 위해, 그림 7과 같이 상위 단어 집합에서 리뷰의 전반적인 분위기를 나타낼 수 있는 단어들만을 추출한 집합을 사용하였다. 이후 실험 방법은 1세트와 동일하게 진행하였다.

```

good more favorite
great hot bad
chinese tasty much
other better last
best few next
little small big
nice first special
delicious friendly amazing
fresh many only
korean sweet new
    
```

그림 6. 2세트에서 사용한 특징벡터의 예

```

good bad worth
great special fantastic
best amazing clean
nice decent reasonable
delicious excellent yummy
fresh awesome okay
tasty super flavorful
better happy fine
friendly perfect wonderful
favorite ok interesting
    
```

그림 7. 2세트에서 사용한 정제된 특징벡터의 예

### 3.3 실험 결과

WEKA에서 제공하는 통계 분석 결과 중에서 알고리즘들의 성능을 비교하는 척도로 F-Measure 값을 사용하였다.

	AdaBoost	ANN	Decision Tree	kNN	Naïve Bayes	SVM
P	0.005	0.08	0.058	0.082	0.076	0.068
N	0.667	0.667	0.665	0.667	0.667	0.668
Avg	0.336	0.373	0.362	0.374	0.371	0.368

그림 8. 상위 10개 특징벡터 적용 결과(1세트)

	AdaBoost	ANN	Decision Tree	kNN	Naïve Bayes	SVM
P	0.169	0.098	0.393	0.11	0.666	0.666
N	0.644	0.666	0.583	0.665	0.081	0.076
Avg	0.406	0.382	0.488	0.388	0.373	0.371

그림 9. 상위 20개 특징벡터 적용 결과(1세트)

	AdaBoost	ANN	Decision Tree	kNN	Naïve Bayes	SVM
P	0.034	0.124	0.096	0.146	0.142	0.11
N	0.67	0.671	0.668	0.668	0.669	0.672
Avg	0.352	0.397	0.382	0.407	0.405	0.391

그림 10. 상위 30개 특징벡터 적용 결과(1세트)

그림 8 ~ 10은 PCFG Parser로 추출한 특징 벡터와 샘플 문서들을 대상으로 실험한 결과물 중 F-Measure 값을 나타낸 것이다. 평균적인 값으로 볼 때, kNN의 성능이 가장 좋게 나왔다.

	AdaBoost	ANN	Decision Tree	kNN	Naïve Bayes	SVM
P	0.013	0.062	0.048	0.064	0.062	0.055
N	0.667	0.666	0.666	0.666	0.666	0.666
Avg	0.34	0.364	0.357	0.365	0.364	0.361

그림 11. 정제된 상위 10개 특징벡터 적용 결과(1세트)

	AdaBoost	ANN	Decision Tree	kNN	Naïve Bayes	SVM
P	0.667	0.074	0.048	0.076	0.071	0.209
N	0.006	0.667	0.666	0.667	0.666	0.642
Avg	0.337	0.37	0.357	0.371	0.369	0.426

그림 12. 정제된 상위 20개 특징벡터 적용 결과(1세트)

	AdaBoost	ANN	Decision Tree	kNN	Naïve Bayes	SVM
P	0.667	0.08	0.048	0.554	0.216	0.648
N	0.006	0.668	0.666	0.455	0.643	0.203
Avg	0.337	0.374	0.357	0.504	0.43	0.425

그림 13. 정제된 상위 30개 특징벡터 적용 결과(1세트)

그림 11 ~ 13은 PCFG Parser로 추출한 정제된 특징 벡터와 샘플 문서들을 대상으로 실험한 결과물 중

F-Measure 값을 나타낸 것이다. 평균적인 값으로 볼 때, kNN의 성능이 가장 좋게 나왔다.

	AdaBoost	ANN	Decision Tree	kNN	Naive Bayes	SVM
P	0.64	0.681	0.707	0.7	0.681	0.682
N	0.763	0.774	0.772	0.762	0.773	0.774
Avg	0.701	0.728	0.739	0.731	0.727	0.728

그림 14. 상위 10개 특징벡터 적용 결과(2세트)

	AdaBoost	ANN	Decision Tree	kNN	Naive Bayes	SVM
P	0.675	0.708	0.71	0.69	0.708	0.682
N	0.779	0.764	0.768	0.727	0.765	0.775
Avg	0.727	0.736	0.739	0.709	0.737	0.728

그림 15. 상위 20개 특징벡터 적용 결과(2세트)

	AdaBoost	ANN	Decision Tree	kNN	Naive Bayes	SVM
P	0.676	0.737	0.718	0.686	0.738	0.708
N	0.779	0.742	0.763	0.712	0.77	0.776
Avg	0.727	0.739	0.741	0.699	0.754	0.742

그림 16. 상위 30개 특징벡터 적용 결과(2세트)

그림 14 ~16은 POS Tagger로 추출한 특징 벡터와 샘플 문서들을 대상으로 실험한 결과물 중 F-Measure 값을 나타낸 것이다. 평균적인 값으로 볼 때, Decision Tree의 성능이 가장 좋게 나왔다.

	AdaBoost	ANN	Decision Tree	kNN	Naive Bayes	SVM
P	0.592	0.592	0.63	0.627	0.571	0.596
N	0.665	0.669	0.663	0.654	0.679	0.672
Avg	0.629	0.63	0.647	0.641	0.625	0.634

그림 17. 정제된 상위 10개 특징벡터 적용 결과(2세트)

	AdaBoost	ANN	Decision Tree	kNN	Naive Bayes	SVM
P	0.544	0.656	0.685	0.672	0.698	0.693
N	0.688	0.707	0.705	0.686	0.706	0.715
Avg	0.616	0.682	0.695	0.679	0.702	0.704

그림 18. 정제된 상위 20개 특징벡터 적용 결과(2세트)

	AdaBoost	ANN	Decision Tree	kNN	Naive Bayes	SVM
P	0.542	0.664	0.696	0.681	0.717	0.712
N	0.688	0.717	0.713	0.691	0.716	0.728
Avg	0.615	0.691	0.705	0.686	0.716	0.72

그림 19. 정제된 상위 30개 특징벡터 적용 결과(2세트)

그림 17 ~ 19는 POS Tagger로 추출한 정제된 특징 벡터와 샘플 문서들을 대상으로 실험한 결과물 중 F-Measure 값을 나타낸 것이다. 평균적인 값으로 볼 때, SVM의 성능이 가장 좋게 나왔다.

결과적으로, 1세트보다 2세트의 성능이 더 좋게 나왔다. PCFG Parser를 이용하여 추출한 특징벡터들의 빈도가 POS Tagger를 이용한 경우보다 상대적으로 적어서, 정확하게 분류할 수 있는 규칙을 만들지 못한 것으로 판단된다. 또한, 정제된 특징벡터를 이용한 경

우 성능이 낮아지는 것을 볼 수 있는데, 이 또한, 특징 벡터의 빈도가 영향을 준 것으로 판단된다.

#### IV. 결론 및 향후 연구 방향

본 논문에서는 대량의 데이터를 요약 및 정리하는 관점으로 기계학습 기법을 이용하여 긍정, 부정으로 분류하였다. 여기서는 다양한 분류 알고리즘을 적용하여 성능을 비교하였다. 아울러, 특징 벡터의 종류 및 사이즈 변화에 따른 분류 성능의 차이도 비교하였다. 본 논문에서는 Adaboost, ANN, Decision Tree, kNN, Naive Bayes, SVM을 사용했는데 평균적으로 POS Tagger를 이용하여 특징벡터를 추출하고 SVM으로 분류했을 때의 성능이 우수한 편이라는 결과가 얻어졌다. 기계학습 기법의 단점인 수작업으로 긍정, 부정으로 분류하는 데 많은 시간이 소요되어, 별점을 분류 기준으로 사용하여 실험하였기 때문에, 향후에는 여러 사람을 통해서 수작업으로 분류된 결과의 평균을 가지고 실험을 해 볼 필요가 있다.

#### 감사의 글

이 논문은 한국연구재단의 기초연구사업 (No. 2012-007498)의 지원을 받아 수행된 연구임.

#### 참고문헌

- [1] Moontae Lee et al., "Multiclass Sentiment Analysis with Restaurant Reviews", Final Projects from CS 224N / Ling 284 for Spring 2009/2010.
- [2] Ziqiong Zhang et al., "Sentiment classification of Internet restaurant reviews written in Cantonese", Expert Systems with Applications 38, 2011, pp. 7674-7682
- [3] Hanhoon Kang et al., "Senti-lexicon and improved Naive Bayes algorithms for sentiment analysis of restaurant reviews", Expert Systems with Applications 39, 2012, pp. 6000-6010
- [4] <http://en.wikipedia.org/wiki/AdaBoost>
- [5] [http://en.wikipedia.org/wiki/Artificial\\_neural\\_network](http://en.wikipedia.org/wiki/Artificial_neural_network)
- [6] [http://en.wikipedia.org/wiki/Decision\\_tree](http://en.wikipedia.org/wiki/Decision_tree)
- [7] <http://en.wikipedia.org/wiki/KNN>
- [8] [http://en.wikipedia.org/wiki/Naive\\_bayes](http://en.wikipedia.org/wiki/Naive_bayes)
- [9] [http://en.wikipedia.org/wiki/Support\\_vector\\_machine](http://en.wikipedia.org/wiki/Support_vector_machine)
- [10] <http://www.cs.waikato.ac.nz/ml/weka/>
- [11] <http://nlp.stanford.edu/software/lex-parser.shtml>
- [12] <http://nlp.stanford.edu/software/tagger.shtml>