NLP Assignment I: SVM-based Sentiment Classification

Maja Trebacz, Newnham College, mt675@cam.ac.uk
November 2018

I. Introduction

Sentiment Classification is a popular classification task in which one can apply Bag-of-Words based machine learning methods. The pioneering and highly-cited paper in this field was written by Pang et al. in 2002 [1]. They used movie reviews as a dataset and found that machine learning methods perform much better than the human-produced baseline.

The purpose of this experiment is to replicate the findings from this paper (Pang et al., 2002) and compare the performance of Naive Bayes and Support Vector Machines.

II. Data

Provided dataset consists of IMDd movie reviews, and contains 1000 positive and 1000 negative reviews. The data underwent tokenization and I applied stemming using the Porter Stemmer implementation [2] (original paper is not using stemming).

III. Methodology

I experienced with two standard algorithms: Naive Bayes and Support Vector Machines. I used the implementations from sklearn Python library [3]. For NaiveBayes, I used the MultinomialNB [4] module with Laplace smoothing. For Support Vector Machines, I used the SVC module [5] (with a linear kernel).

Similarly as in the paper, I applied the standard bag-of-words approach and used the vectors of word counts as the features. I compared it with bigrams and combination of unigrams + bigrams. I repeated the measurements with using word presence instead of the counts (a binary variable indicating presence or absence of a unigram or bigram).

Following the paper, I also tried feature cutoff technique. In order to avoid overfitting, I choose the cutoff just once (discard the features with a frequency lower than 5) and did not experiment with different count cutoffs.

To calculate the accuracies I used a 10-fold cross-validation technique with stratified, Round-robin splitting. Each time I trained the classifier on nine folds (1800 reviews) and tested on one fold (200 reviews).

IV. Results

My measurements confirm the findings from the paper that the machine learning techniques perform well on the Sentiment Classification task. Every experiment achieved mean accuracies above 80%. The classification accuracies resulting from using only unigrams as features are shown in line 1 of Table 1. Measurements on SVMs showed higher accuracy, but a two-tailed sign test (performed on the combined predictions from all the folds) showed no statistically significant difference at the p=0.62 significance level (Plus: 203 Minus: 183 Null: 1617).

Features	# of features	freq. or pres.	NB	SVM
1. unigrams	38386	frequency	**81.45%**	**83.50%**
2. unigrams	"	presence	81.90%	85.25%
3. bigrams	461492	frequency	84.00%	81.55%
4. bigrams	"	presence	85.65%	82.35%
5. unigrams+bigrams	499878	frequency	83.55%	84.50%
6. unigrams+bigrams	"	presence	85.50%	**87.85%**
7. uni. & feature cutoff	12547	frequency	82.40%	82.95%
8. uni. & feature cutoff	"	presence	82.75%	85.15%

TABLE I
Average 10-fold cross-validation accuracies, in percent

Lines 5 and 6 of the results table shows that adding bigrams information as the improves the achieved accuracy. The highest mean accuracy of $87,85\%$ was observed with using both unigrams and bigrams and considering just word presence. However, the improvement between the baseline Naive Bayes using unigrams and this result is not statistically significant at the p=0.70 level.

Lines 3 and 4 of the tables show that relying just on bigrams as the features decrease the performance of SVM.

In general, a two-tailed sign test showed no statistically significant difference between any of the stated results.

V. Conclusions

Naive Bayes despite being a simplistic model performs very well in the sentiment classification task. SVM showed higher accuracy but the improvement is not statistically significant.

Appendix

- Word count: 492
- Code pointer: https://github.com/hey-now/NPL-assignment1

References

[1] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan (2002). *Thumbs up? Sentiment Classification using Machine Learning Techniques*. Proceedings of EMNLP.
[2] https://tartarus.org/martin/PorterStemmer/
[3] Scikit learn library for Python, https://scikit-learn.org/
[4] Naive Bayes classifier for multinomial models, *sklearn.naive_bayes.MultinomialNB*, https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
[5] C-Support Vector Classification implementation based on libsvm, *sklearn.svm.SVC*, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html