

# NLP Assignment I: SVM-based Sentiment Classification - Bag of Words

Theodora I. Zamfirache, Newnham College, tiz20@cam.ac.uk  
November 2018

## I. INTRODUCTION

The task of classifying documents by overall sentiment, e.g. determining whether a movie review is positive or negative, can be performed using bag-of-features techniques and a suite of machine learning classifiers, such as Naive Bayes and Support Vector Machines. The present work aims to replicate some of the findings of Pang et al[1], one of the most prominent papers in the domain.

## II. DATASET

The dataset that was used in the experiment consisted of 2000 movie reviews from IMDB, split equally into 1000 positive reviews and 1000 negative reviews, annotated with groundtruth<sup>1</sup>. Reviews are tokenised, and later each token is stemmed using NLTK's version of Porter Stemmer[2]. Pang et al uses a similar dataset, although no stemming is applied.

## III. METHODOLOGY

Two machine learning methods were used: Naive Bayes and SVMs. The chosen framework was that of bag-of-features, where the features  $\{f_1, \dots, f_n\}$  are unigrams, bigrams or a combination of the two. Each document is represented as a vector of size  $n$ , where the  $i^{th}$  entry represents either the occurrence frequency or simply the presence of feature  $f_i$  in the document.

Feature cutoffs were applied to the models that used frequency, with unigrams being limited to those appearing more than 3 times in the data, and bigrams more than 7 times. These values were chosen so as to maintain a reasonable amount of data while also reducing the computation time - no accuracy testing was performed, so as not to risk overfitting.

Data was split into 10 equal-sized folds, with balanced class distribution in each fold. Round-Robin cross-validation was used in calculating the accuracy of each model, where 9 of the folds were used for training and 1 fold for testing.

The models were implemented using the Scikit-learn library[3].

- The **Naive Bayes** models used *MultinomialNB* with Laplace smoothing.
- The **SVM** models used *svm.SVC* with a linear kernel.

<sup>1</sup>Data was given as part of an assignment for a course on NLP

## IV. EVALUATION

The results of the tests are displayed in Table I.

### A. Unigrams vs Bigrams vs Unigrams+Bigrams

Bigrams and Unigrams+Bigrams give higher accuracy than unigrams, indicating that bigrams help capture the context more effectively.

### B. Frequency vs. presence

Pang et al found that feature presence gives better accuracy than feature frequency, and indeed, the same applies for the present work.

### C. Stemming

The data tokens have been stemmed using Porter stemmer. Counterintuitively, very few tests where stemming is applied exceed the accuracy of their no-stemming counterpart.

### D. Naive Bayes vs. SVM

To measure statistical significance of the results, the two-tailed sign test was used, and a value of  $p = 0.1$  was set as the threshold for rejecting the Null hypothesis. However, none of the experiments resulted in a p value below this threshold, therefore the difference in accuracy between the two systems must be due to noise.

Features	Freq or Pres	Stem	NB	SVM
unigram	freq	stem	81.60%	83.60%
unigram	freq	no	81.55%	83.78%
unigram	pres	stem	82.60%	83.25%
unigram	pres	no	82.80%	85.75%
bigram	freq	stem	83.75%	82.65%
bigram	freq	no	84.00%	81.55%
bigram	pres	stem	86.65%	83.60%
bigram	pres	no	85.85%	84.05%
unigram+bigram	freq	stem	83.75%	85.00%
unigram+bigram	freq	no	83.95%	84.90%
unigram+bigram	pres	stem	85.90%	87.60%
unigram+bigram	pres	no	85.50%	<b>88.00%</b>

TABLE I: Mean accuracy for each of the models

## V. CONCLUSIONS

Despite the simplicity of the model, Naive Bayes does well in this task, as there is no statistical significant difference between it and SVMs. Accuracy-wise, the SVM with unigram+bigram and feature presence performed best, achieving 88.00%.

## REFERENCES

- [1] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.
- [2] <http://www.nltk.org/howto/stem.html>, Accessed 9 November 2018
- [3] <https://scikit-learn.org/>, Accessed 9 November 2018