NLP Report 1 — A. Koc (ak2023) — 495 words

1 Introduction

The purpose of this report is to replicate efforts done by Pang et al[1]. Data used comes from an NLP course and consists of movie reviews. Reproductions do not cover the whole paper and we experiment with some additional pre-processing.

2 Background

This replication is based on Pang et als paper (*the paper*), who apply machine learning techniques to classify sentiment. We focus only on Naive Bayes (*NB*) and Support Vector Machines (*SVM*). Additionally, we experiment with stemming - our assumption was that stemming increases accuracy because words will achieve more significant frequencies.

3 Method

We represented sentiment as vectors specifying words' presence or frequency with dimensions equal to unique word count across reviews. They served as input to models developed with Python sklearn package.

NB was implemented using MultinomialNB with Laplace smoothing and SVM using SVC with a linear kernel. The kernel is linear to mimic SVM-light. Different kernels caused drops in accuracy.

We used a corpus larger than the paper's: 2000 instead of 1400 reviews. Cross-validation was performed on 10 folds (instead of 3). Unlike the paper we used Porter stemming but didn't investigate negation.

4 Results and discussion

	No stemming			Stemming		
Features	# features	NB	SVM	# features	NB	SVM
unigrams frequency	38467	81.60%	82.50%	28319	81.20%	82.60%
unigrams presence	38467	82.55%	84.90%	28319	82.20%	83.65%
unigrams+bigrams	498451	85.80%	87.85%	446198	85.60%	87.05%
bigrams	459984	85.60%	82.75%	417879	85.40%	83.20%

Table 1: Accuracy of models without feature cut-off.

As a whole, accuracies clearly surpass the paper. The best performing model reached nearly 5% higher (87.85%) than the paper's best model (82.9%). The datasets, cross-validation techniques and models used might explain this difference.

	No stemming			Stemming		
Features	# features	NB	SVM	# features	NB	SVM
unigrams frequency	14610	82.65	84.45	10931	82.45	83.4
unigrams presence	14610	82.65	84.45	10931	82.45	83.4
unigrams+bigrams	54382.1	85.2	87.3	52238	85.45	86.4
bigrams	39772.1	84.5	82.1	41306.9	85.15	82.85

Table 2: Accuracy of models with feature cut-off=4

Similarly to the paper, higher accuracies were achieved utilizing feature presence not feature frequency. This might be due to duality of word semantics.

Our earlier hypothesis on stemming has not been proven. Stemming seems to have little, mixed effects on accuracies. Its only significant benefit is in SVM binomial models. Feature presence explains this phenomenon: when not stemming, frequent words are noted multiple times if they occur inflected. Moreover, we used a very basic stemming technique.

Looking at line 3 in tables 1 and 2, unigrams+bigrams present best performance whereas the paper gets best performance for unigrams. As mentioned above, this might be due to different experiment contexts.

	No frequency cut-off		Frequency cut-off=4	
Features	No stemming	Stemming	No stemming	Stemming
unigrams frequency	67.07%	61.49%	65.46%	75.40%
unigrams presence	63.48%	73.30%	65.46%	75.40%
unigrams+bigrams	69.35%	72.24%	67.74%	77.52%
bigrams	60.77%	68.26%	62.25%	63.70%

Table 3: Prob. NB is statistically the same as the SVM.

An interesting result is that feature cut-off has negative influence on all models but NB with unigrams. The improvement reflects the difference in approaches to classification of the two methods in question. The number of features analysed drops down by nearly a third (line 1 table 3), removing a lot of noise coming from infrequent, insignificant words.

Overall, features without pre-processing yield best results. SVMs had consistently higher accuracies except for bigrams. The two models, however, are not significantly different (table 2). In every experiment the Null Hypothesis was not rejected most probably caused by lack of sufficient data. Having said that, the best results came from an SVM running on non-pre-processed unigrams+bigrams reaching 87.85% accuracy.

[1] Pang, B., Lee, L. & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (p./pp. 79–86), .