Binary Sentiment Classification

Adewale-Young Adenle

January 2025

Overview. We experiment with two distinct approaches for binary (positive vs. negative) sentiment classification on a movie review dataset, discarding neutral examples. First, we train a logistic regression (LR) model with TF-IDF features to establish a high-performing baseline. Second, we introduce a Deep Averaging Network (DAN) that averages pretrained word embeddings and feeds them into a multi-layer perceptron. We also explore mini-batching for faster neural-network training.

1. Logistic Regression

Implementation. We convert each sentence into a TF-IDF weighted vector (unigrams only). We then train LR with stochastic gradient descent, tuning the regularization term and learning rate via the development set. Additionally:

- *Bigram features* (pairs of adjacent tokens) increased feature sparsity, reducing performance.
- *Better features* (unigrams + bigrams + sentiment lexicon counts) improved dev accuracy.

Results. Table 1 (left) summarizes our LR-based evaluations. Unigrams alone reached a dev accuracy of ∼78%. Adding only bigrams dropped dev accuracy to 65%. Including curated sentiment lexicons boosted performance to 75%, highlighting that feature engineering can significantly improve LR classifiers.

2. Deep Averaging Network + Batching

Model. We average pretrained GloVe embeddings (300d) for each sentence (masking out pad tokens). A small feedforward network (two ReLU layers + dropout) classifies the resulting vector via a final `LogSoftmax` layer.

Batching. We compare two regimes:

- *Batch size = 1:* Achieves ∼80% dev accuracy, though training is slower.
- *Mini-batching:* Speeds training substantially, yielding a similar 78–79% dev accuracy.

3. Combined Results and Conclusion

Table 1 compares logistic regression and DAN approaches. Logistic regression proves highly effective with carefully engineered features. The Deep Averaging Network, meanwhile, benefits from pretrained embeddings and can train efficiently via batching. Both methods comfortably exceed 77% dev accuracy, making them solid choices for sentiment analysis tasks.

LR Feature Set	Train Acc.	Dev Acc.
Unigrams	0.890	0.780
Bigrams Only	0.610	0.570
Better (Uni+Bi+Lex)	0.710	0.700

Method	Batching	Dev Acc.	Time
DAN	No	0.7993	∼173s
DAN	Yes	0.7878	∼126s

Table 1: Left: Logistic Regression variations. Right: DAN with/without batching.