Advanced Natural Language Processing

Assignment 1: Twitter hate speech detection using Naive Bayes and Logistic Regression

Lucia Welther, Matrikelnummer: 835106

November 16, 2025

1 Smoothing Parameter Analysis

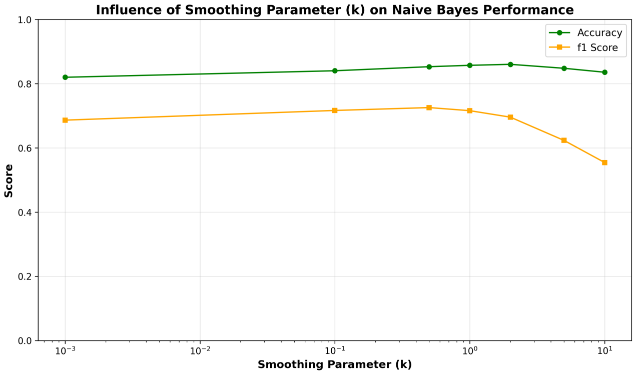

Figure 1: Accuracy and F1-score as functions of smoothing parameter k

Observations from Figure 1:

- Very small smoothing values (e.g. $k = 0.001$) lead to overfitting: the model depends too much on exact training frequencies and handles unseen words poorly.
- Moderate smoothing (e.g. $k = 0.5$) gives the best performance by balancing observed counts with reasonable estimates for unseen words.
- Very large smoothing values (e.g. $k = 10$) over-smooth the probabilities, making word distinctions weak and reducing accuracy.

2 Feature Engineering

2.1 Feature Set 1: Stopword Removal

Removing stopwords reduces noise by eliminating very frequent and non-informative words. Stopwords occur in both classes and do not help distinguish offensive from non-offensive tweets. After removing them, the model focuses on more meaningful content words, improving accuracy and F1-score.

2.2 Feature Set 2: Bigrams

Bigrams capture word order and context, allowing the model to distinguish phrases like “not good” from “good”. Offensive language often contains specific multi-word expressions that bigrams can identify. By combining unigrams and bigrams, the model learns both individual words and contextual word pairs.

Table 1: Performance comparison of feature engineering approaches

Feature Set	Accuracy	F1-Score
Baseline	0.8572	0.7159
Stopword Removal (Feature 1)	0.8597	0.7209
Bigrams (Feature 2)	0.8548	0.6672

As seen in Table 1, Feature 1 (stopword removal) slightly improves both metrics (accuracy: 0.8597, F1: 0.7209), demonstrating that removing non-discriminative words helps focus on meaningful content. However, Feature 2 (bigrams) decreases performance (accuracy: 0.8548, F1: 0.6672), likely due to vocabulary expansion causing data sparsity issues. The increased feature space introduces many rare bigrams with unreliable probability estimates, leading to overfitting. This suggests that for this dataset size, vocabulary reduction is more effective than feature expansion.

3 Model Evaluation Results

The Naive Bayes classifier with optimal smoothing parameter $k = 0.5$ and the Logistic Regression classifier with learning rate $\eta = 0.01$, 10 epochs, and L2 regularization parameter $C = 0.1$ achieved:

Table 2: Performance comparison of Naive Bayes vs. Logistic Regression

Model	Accuracy	F1-Score
Naive Bayes	0.8572	0.7159
Logistic Regression	0.8148	0.4506

As shown in Table 2, Naive Bayes significantly outperforms Logistic Regression (F1: 0.7159 vs. 0.4506). Despite Logistic Regression's flexibility, Naive Bayes excels due to appropriate independence assumptions, resistance to overfitting on small data, and sufficient training compared to Logistic Regression's limited 10 epochs. This demonstrates that simpler models can be more effective with limited training data.

4 Bonus: Discussion Questions

- **Effect of Smoothing Parameter:** The smoothing parameter k prevents zero probabilities for unseen words by adding pseudocounts; very small k may cause overfitting, while very large k makes word probabilities nearly uniform and harms model performance.
- **Word Order Shuffling:** Shuffling word order does not affect Naive Bayes or bag-of-words logistic regression because both treat documents as unordered collections of words.
- **Purpose of Softmax Function:** Softmax converts raw class scores into normalized probabilities that sum to 1, enabling meaningful comparison between classes.
- **Decision Threshold Tuning:** The decision threshold can be adjusted after training based on validation or test predictions, so no retraining is required.

4.1 GitHub Repository

GitHub Repository: `https://github.com/luzecode/aNLP-Assignment1-Hate-Speech-Detection.git`

Last commit hash: `First Commit: NLP Assignment1`