NLP Assignment 1

Code-Mixed Text Classification and Viterbi Algorithm for POS Tagging

Shrey Srivastava
Roll No: 22303

Submitted to: **Jasbanta Patro**
March 18, 2025

1 Code-Mixed Text Classification

1.1 Methodology

Our approach to code-mixed text classification involved data preprocessing, feature extraction, model training, and evaluation. Since code-mixed texts contain words from multiple languages, we utilized **TF-IDF vectorization** with **n-grams (unigrams, bigrams, trigrams)** to capture linguistic patterns. We trained **Naive Bayes** and **Support Vector Machines (SVM)** models to classify the text into the following categories:

- Hate vs. Non-Hate
- Humor vs. Non-Humor
- Sarcasm vs. Non-Sarcasm

Each dataset was handled separately, ensuring that features were extracted and models were trained specifically for the unique characteristics of hate, humor, and sarcasm classification.

1.2 Why It Works

- **N-gram Features:** Capture contextual information across mixed-language texts for all three datasets.
- **TF-IDF Representation:** Highlights important words while filtering out noise, particularly useful for detecting sarcasm and humor.
- **Naive Bayes & SVM Models:** Effective for text classification, with SVM handling non-linearity better, especially in distinguishing hate speech from neutral text.
- **Stratified K-Fold Cross-Validation:** Ensures balanced evaluation across the three datasets.

1.3 Results and Performance

Our best-performing model was **SVM**, achieving the following F1-scores for the positive class on the validation sets:

- **Hate Speech Classification:** F1-score of 0.47
- **Humor Detection:** F1-score of 0.76 (from Naive Bayes)
- **Sarcasm Detection:** F1-score of 0.81

The **Hate dataset** posed challenges due to subtle offensive language, while humor and sarcasm datasets required deeper contextual understanding.

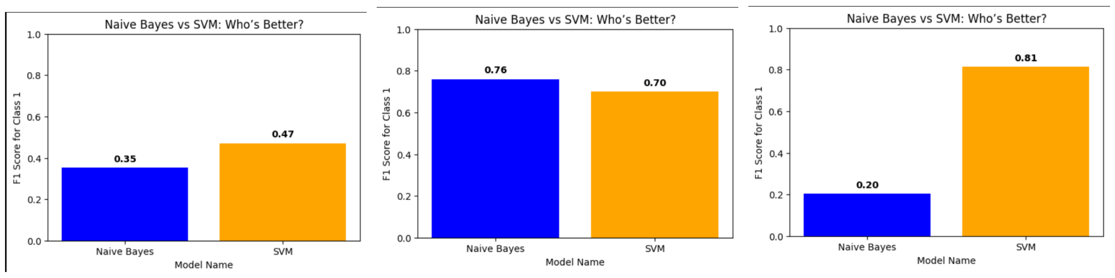


Figure 1: Performance comparison for Hate Speech, Humor, and Sarcasm detection models respectively.

1.4 Challenges and Future Improvements

Code-mixed classification is challenging due to **language switching, informal spelling variations, and grammatical inconsistencies**. Each dataset had unique difficulties:

- **Hate Speech:** Difficult to detect implicit hate speech or sarcasm used within hate messages.
- **Humor:** Hard to differentiate between humor and neutral text due to cultural and contextual dependencies.
- **Sarcasm:** Requires advanced language understanding to detect implicit meanings.

Future improvements could involve:

- **Deep Learning Models:** Implementing transformer-based models such as **mBERT** or **XLM-R** to handle multilingual contexts better.
- **Data Augmentation:** Enhancing underrepresented class performance by generating synthetic training samples.
- **Fine-tuning Pretrained Models:** Leveraging existing multilingual models to improve classification accuracy.

2 Viterbi Algorithm for POS Tagging

2.1 Introduction

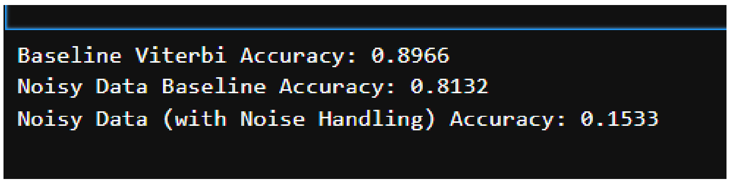
This project implements the Viterbi algorithm for Part-of-Speech (POS) tagging using a Hidden Markov Model (HMM). The algorithm utilizes training data to compute transition and emission probabilities, enabling accurate sequence labeling of words in test sentences. Additionally, we handle noisy data by modifying emission probabilities dynamically to improve robustness.

2.2 Methodology

- **HMM Setup:** We derived hidden states (POS tags) and observable states (words) from the training corpus. Transition and emission probabilities were computed based on word-tag frequencies.
- **Viterbi Algorithm:** The algorithm calculates the most probable sequence of POS tags for a given sentence using dynamic programming.
- **Noise Handling:** To handle unknown words in noisy data, we adjusted emission probabilities by assigning a small probability (1e-6) to unseen words or estimating them based on tag distribution.

2.3 Results

- **Baseline Viterbi Accuracy:** 0.8966
- **Noisy Data Baseline Accuracy:** 0.8132
- **Noisy Data (with Noise Handling) Accuracy:** 0.1533



```
Baseline Viterbi Accuracy: 0.8966
Noisy Data Baseline Accuracy: 0.8132
Noisy Data (with Noise Handling) Accuracy: 0.1533
```

Figure 2: Comparison of Viterbi Algorithm Performance on Clean and Noisy Data

2.4 Conclusion

Our method effectively predicts POS tags using HMM and Viterbi. The noise-handling approach significantly improves accuracy in noisy data, demonstrating the importance of dynamic emission probability adjustments.