Beyond Subwords: Systematic Evaluation of Tokenization Methods Across Sentiment Analysis Paradigms

Sadhika Singh
IIT Hyderabad
me22btech11047@iith.ac.in

Abstract

This study presents a systematic evaluation of six tokenization methods—SentencePiece BPE, SentencePiece Unigram, Standalone BPE, Standalone Unigram, WordPiece, and tiktoken—for sentiment analysis on the SST-5 and IMDB datasets using a range of machine learning models. Our extensive evaluation, comprising over 1,160 experimental configurations with cross-validation, reveals consistent performance hierarchies across tasks while highlighting task-specific optimization strategies. SentencePiece Unigram emerges as the most robust tokenizer across all tasks, achieving optimal performance when paired with ANN. To further validate these findings, we implemented transformer architectures both from-scratch and using the HuggingFace Transformers library, confirming the stability of tokenizer rankings across implementations. The study establishes crucial baselines for tokenization research and provides evidence-based recommendations for practitioners working on sentiment analysis applications. This work is ongoing and results should be considered preliminary, with further refinements and experiments planned.

1 Introduction

Tokenization represents a fundamental preprocessing step in NLP pipelines, directly influencing downstream task performance through its impact on feature representation and model understanding. While significant attention has been devoted to advanced architectures, the systematic evaluation of tokenization methods across diverse sentiment analysis tasks remains largely unexplored. The rise of subword tokenization techniques such as BPE, Unigram, WordPiece, and optimized implementations like tiktoken has produced an intricate setting in which practitioners lack clear guidelines for method selection.

This study addresses this gap by providing the first comprehensive evaluation of tokenization methods across binary classification, multi-class classification, and regression tasks. In addition, we validate these findings using transformer architectures implemented both from-scratch and through the HuggingFace library. This dual evaluation confirms that tokenizer choice remains the dominant factor, independent of implementation details.

2 Background

2.1 Tokenization Algorithms

Tokenization plays a foundational role in natural language processing pipelines, and recent advances have led to the development of several subword-based algorithms that offer varying trade-offs in efficiency, linguistic coverage, and downstream performance. This section outlines the six tokenization methods evaluated in our study:

- **Byte-Pair Encoding (BPE):** Introduced by Sennrich et al. (2016), BPE merges the most frequent character pairs iteratively to construct subword vocabularies. It effectively handles out-of-vocabulary words while maintaining a balance between character-level and word-level representations.

- **SentencePiece BPE:** Developed by Kudo and Richardson (2018), SentencePiece extends the BPE algorithm within a language-agnostic framework that removes the reliance on whitespace-based preprocessing. It allows for consistent tokenization across diverse languages.

- **SentencePiece Unigram:** Also proposed by Kudo (2018), the Unigram Language Model is a probabilistic approach that begins with an overcomplete vocabulary and prunes tokens based on their likelihood contribution, offering strong performance in linguistically complex scenarios.

- **Standalone BPE:** A direct implementation of the original BPE algorithm outside of the SentencePiece framework. It operates on raw text and serves as a baseline for evaluating the impact of implementation-specific variations.

- **Standalone Unigram:** Similar in principle to SentencePiece Unigram, this variant implements the Unigram Language Model independently, enabling an isolated assessment of its behavior and effectiveness across tasks.

- **WordPiece:** Developed by Schuster and Nakajima (2012) and adopted in BERT (Devlin et al., 2019), WordPiece uses a likelihood-based strategy for merging tokens, producing vocabularies tailored to optimize performance on downstream NLP tasks.

- **tiktoken:** Designed by OpenAI, tiktoken builds on BPE principles with optimizations for speed and memory usage, making it well-suited for large-scale transformer models used in modern LLMs.

2.2 Tokenization Evaluation Studies

The systematic evaluation of tokenization methods has gained increasing attention as the field recognizes the critical impact of tokenization choices on downstream performance. Choo and Kim (2023) [1] conducted a comparative study evaluating SentencePiece against Mecab-Ko for Korean sentiment analysis using smartphone review data, demonstrating the superior performance of subword tokenization approaches over morpheme-based methods, particularly for handling out-of-vocabulary words and domain-specific terminology.

While prior studies have compared tokenization methods, most are limited to single tasks or small tokenizer sets. This study extends the scope by offering a systematic, multi-task evaluation framework that highlights cross-task performance patterns.

2.3 Sentiment Analysis

Sentiment analysis has progressed through several paradigmatic shifts, from rule-based lexicon approaches to sophisticated machine learning models. Initial work primarily addressed binary polarity classification, exemplified by large-scale datasets like IMDB, which offer balanced positive and negative sentiment labels for coarse-grained analysis. More recent research has expanded to include fine-grained sentiment understanding and regression-based formulations, aiming to capture the subtle gradations of emotional intensity.

The introduction of the Stanford Sentiment Treebank (Socher et al.,2013) represented a key development, enabling multiclass and continuous sentiment modeling through phrase-level annotations. However, despite such advances, much of the literature continues to treat sentiment as a discrete classification problem, neglecting the continuous spectrum of emotional expression. Traditional categorical approaches struggle to capture nuanced distinctions, for example, between ’excellent’ and ’outstanding’ that carry important sentiment intensity cues.

3 Methodology

3.1 Tasks

- **Fine Grained Classification (SST-5):** Predicts one of five sentiment classes (Very Negative to Very Positive) from sentence-level data.
- **Ordinal Regression (SST-5):** Models sentiment as a continuous ordinal variable (0.0 to 4.0), capturing gradations of sentiment intensity.
- **Binary Classification (IMDB):** Classifies full-length movie reviews as either positive or negative sentiment, offering a large-scale benchmark with minimal class imbalance.

3.2 Datasets

Stanford Sentiment Treebank (SST-5): The Stanford Sentiment Treebank, comprising 11,855 movie review sentences with fine-grained sentiment labels, serves as the foundation for two of our experiments. The dataset exhibits natural class imbalance with the following distribution:

- Very Negative: 1,510 samples (12.7%)
- Negative: 3,140 samples (26.5%)
- Neutral: 2,242 samples (18.9%)
- Positive: 3,111 samples (26.2%)
- Very Positive: 1,852 samples (15.6%)

For the regression experiment, we treat these labels as continuous values (0.0, 1.0, 2.0, 3.0, 4.0), enabling models to learn ordinal relationships and predict intermediate values. For the classification experiment, we maintain discrete class labels while investigating the impact of different balancing strategies.

IMDB Movie Reviews:

The IMDB Large Movie Review Dataset provides 50,000 movie reviews with binary sentiment labels, offering balanced representation with 25,000 positive and 25,000 negative reviews. This dataset enables evaluation of tokenization methods on large-scale binary sentiment classification with minimal class imbalance concerns.

3.3 Models

Our evaluation encompasses multiple machine learning paradigms to ensure comprehensive assessment of tokenization impact:

- **Naive Bayes (NB):** Multinomial Naive Bayes with Laplace smoothing represents probabilistic approaches to text classification. This model provides baseline performance while offering computational efficiency and interpretability.
- **k-Nearest Neighbors (KNN):** Instance-based learning with cosine similarity metrics enables assessment of tokenization impact on distance-based classification. KNN models are particularly sensitive to feature representation quality, making them ideal for tokenization evaluation.
- **Support Vector Machines (SVM):** Both linear and RBF kernel SVMs are evaluated across experiments, with class weighting employed to address imbalanced datasets. SVMs provide robust performance across diverse feature spaces and offer insights into tokenization effects on margin-based classification.

- **Artificial Neural Networks (ANN):** Multi-layer perceptrons with dropout regularization and early stopping represent non-linear approaches to sentiment analysis. ANNs consistently demonstrate superior performance across tokenization methods, making them crucial for identifying optimal configurations.

- **Ensemble Methods:** Random Forest and XGBoost models provide ensemble-based evaluation, offering insights into tokenization effects on tree-based approaches. These methods complement neural approaches while providing robust baseline performance.

- **Long Short-Term Memory (LSTM):** Recurrent neural networks enable assessment of tokenization impact on sequence modeling approaches. LSTMs are particularly relevant for understanding how different tokenization strategies affect temporal pattern recognition in sentiment analysis.

3.4 Evaluation Method

Cross-Validation: All experiments employ 10-fold cross-validation with multiple random seeds to ensure statistical reliability. This approach provides robust performance estimates while enabling significance testing across tokenization methods. Stratified sampling ensures consistent class distribution across folds, maintaining experimental validity.

Hyperparameter Optimization: RandomizedSearchCV with 3-fold cross-validation optimizes hyperparameters across all models. Search spaces are carefully designed to balance comprehensive exploration with computational feasibility, ensuring fair comparison across tokenization methods while maintaining practical constraints.

Evaluation Metrics:

For **Binary Classification (IMDB Dataset)** we employ:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\tag{6}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

For **Multi-class Classification (SST-5 Dataset)**, we adopt macro-averaged metrics:

$$\text{Macro-Precision} = \frac{1}{n} \sum_{i} \frac{TP_i}{TP_i + FP_i} \tag{8}$$

$$\tag{9}$$

$$\text{Macro-Recall} = \frac{1}{n} \sum_{i} \frac{TP_i}{TP_i + FN_i} \tag{10}$$

$$\tag{11}$$

$$\text{Macro-F1} = 2 \times \frac{\text{Macro-Precision} \times \text{Macro-Recall}}{\text{Macro-Precision} + \text{Macro-Recall}} \tag{12}$$

where $n = 5$ for the SST-5 dataset.

For **Regression (SST-5 Regression)**, we employ:

- Root Mean Square Error (RMSE)

- Mean Absolute Error (MAE)
- R^2: coefficient of determination

4 Experiments

4.1 Experiment 1: SST-5 Sentiment Regression

We formulate SST-5 as regression with continuous labels (0.0–4.0). A novel resampling strategy balances training while preserving regression at inference. Vocabulary sizes of 3K, 5K, and 8K are tested. A naive baseline predictor outputs the dataset mean (2.0553), yielding RMSE=1.2879.

4.2 Experiment 2: SST-5 Fine-Grained Classification

Two phases:

- Phase 1 (Undersampling): All classes undersampled to 1,510. Linear SVM used.
- Phase 2 (Oversampling): Random oversampling; RBF SVM with class weights.

This design allows comparison of balancing strategies.

4.3 Experiment 3: IMDB Binary Classification

IMDB dataset (50,000 reviews) used for large-scale binary classification. Vocabulary sizes 10K and 20K tested. Phase 1 included LSTM; Phase 2 excluded LSTM due to computational cost.

5 Results

5.1 SST-5 Regression Results

Best configuration: SentencePiece Unigram + ANN (8K vocab) with RMSE=1.106, $R^2 = 0.261$. Baseline RMSE=1.288. Transformers library improved RMSE to 1.0248 (+1.1%).

5.2 SST-5 Classification Results

Phase 1: Best F1=0.393 (SP-Unigram + ANN, 5K). Phase 2: Best F1=0.369 (SP-Unigram + ANN, 5K). Transformer implementation slightly improved F1 from 0.384 to 0.391.

5.3 IMDB Binary Classification Results

Peak accuracy=91.24% (Standalone Unigram + ANN). Tokenizer ranking: SP-Unigram, WordPiece, Standalone Unigram $\gg$ BPE methods.

5.4 Transformer vs From-Scratch Comparison

- Identical tokenizer rankings across implementations.
- Library yielded modest 1–2% improvements.
- Gains from CUDA kernel optimization and memory management.
- Development: from-scratch = 2 weeks; library = 2 days.

5.5 Cross-Experiment Analysis

- SP-Unigram consistently best across tasks.
- Neural models (ANN, Transformers) outperform traditional ML.
- Optimal vocab sizes: 8K (regression), 5K (SST-5 classification), 20K (IMDB).
- Tokenization improvements larger in classification than regression.

6 Discussion

6.1 Tokenizer Performance Patterns

Unigram-based tokenizers consistently outperform frequency-based methods. Rankings are robust across architectures, tasks, and implementations.

6.2 Task-Specific Insights

Binary classification shows largest tokenization effect. Regression gains are modest but consistent. Class balancing strategies matter more than tokenizer choice in SST-5 classification, but relative rankings remain stable.

6.3 Model and Implementation Effects

ANN and Transformers dominate performance, showing greatest sensitivity to tokenizer choice. Traditional ML shows limited gains. Transformers library improves stability and consistency but does not alter core research conclusions.

7 Conclusion

We present the first comprehensive evaluation of modern tokenization methods for sentiment analysis across classification and regression tasks, validated through both traditional ML and transformer architectures. SentencePiece Unigram consistently outperforms alternatives, particularly when paired with neural models. Transformer experiments confirm that tokenizer choice dominates over implementation details, with HuggingFace libraries offering modest gains and practical advantages. Our findings establish robust baselines, methodological tools, and practical guidelines for tokenization research, with implications for scaling across domains, languages, and architectures. As this study is still in progress, the reported results represent preliminary findings that may be extended with further experimentation and analysis, confirming the stability of tokenizer rankings across implementations.

References

[1] Choo, S. and Kim, W. (2023). A study on the evaluation of tokenizer performance in natural language processing.