Enhancing Text Polarity Classification through Data Augmentation, Granular Preprocessing, and Comparative Model Analysis

Zie-Wei, Xie

113423036

youto201266@gmail.com

ABSTRACT

This study improves text polarity classification by first augmenting the training data with filtered samples from external review datasets (Amazon, Yelp, IMDb, SST2). A rigorous, multi-step preprocessing pipeline is applied to clean textual noise, addressing encoding, casing, symbols, abbreviations, and punctuation. Experiments are conducted in two phases: initially, traditional machine learning models with TF-IDF are evaluated, tuning feature count and n-gram range. Subsequently, contextual models like BERT, DistilBERT, RoBERTa, XLNet, and ELECTRA are assessed, optimizing batch size, sequence length, and augmented data volume, while notably avoiding stop-word removal and stemming/lemmatization.

1. Dataset Preprocessing

1.1 Initial Data Augmentation

To address the limited size of the initial training set and to better reflect the diverse nature of reviews (spanning products, games, movies, and restaurants), data was augmented using several Hugging Face datasets: mteb/amazon_polarity, Yelp/yelp_review_full, stanfordnlp/imdb, and stanfordnlp/sst2. To prepare the external datasets for augmentation, an initial filtering step was applied to the text samples. Outlier removal based on text length was performed using the interquartile range (IQR) method to ensure that the incorporated texts were not excessively long or short compared to the original competition data. The impact of this filtering on the text length distributions can be observed by comparing the word count distributions of the training and test datasets (Figure 1) with that of the filtered external source dataset (Figure 2). Furthermore, to ascertain lexical similarity, word clouds were generated for the training set (Figure 3), the test set (Figure 4), and the external source dataset (Figure 5). An examination of these word clouds indicated a substantial overlap in vocabulary, supporting the decision to integrate these external datasets to enrich the training corpus.

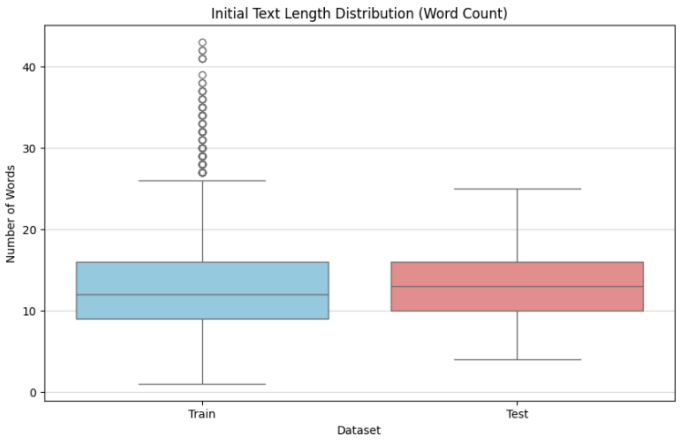

Figure 1: Word Count Distribution of Training and Test Datasets.

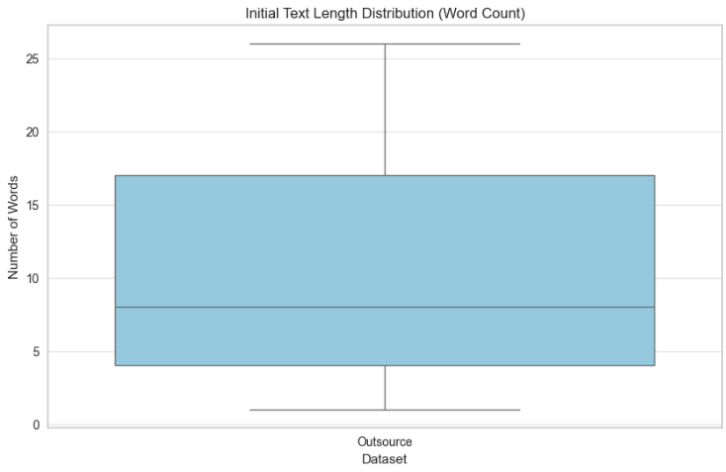

Figure 2: Word Count Distribution of the External Source Dataset.

Figure 3: Word Cloud of the Training set.

Figure 4: Word Cloud of the Test set.

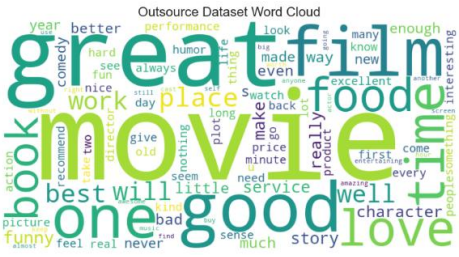

Figure 5: Word Cloud of the External Source

1.2 Text Cleaning and Normalization

Following data augmentation, a unified and comprehensive text cleaning and normalization pipeline was systematically applied to the entire dataset. This process aimed to eliminate textual noise and standardize formatting inconsistencies. The sequence of operations was as follows:

1. The process commenced with preliminary encoding issue resolution, which involved removing Unicode replacement characters (e.g., \ufffd) and specific problematic patterns like double question marks (??) identified during initial data observation. Subsequently, all text underwent case normalization through conversion to lowercase.
2. A significant part of the pipeline focused on symbol and special format standardization. This included converting textual representations of symbols (e.g., -lrb-, -amp-) back to their standard forms (e.g., (, &); unifying various Unicode and ASCII quotes to standard ASCII double (") and single (') quotes, with backticks also converted to single quotes; removing entire words containing underscores (e.g., num_extend) using regular expressions; correcting spacing around periods in common abbreviations (e.g., u . s . to u.s.); and others not pertinent to sentiment.
3. Further standardization was applied to hyphens and abbreviations. This involved removing spaces around hyphens to ensure they were adjacent to connected words (e.g., word - word to word-word); correcting common colloquial or non-standard abbreviations to their standard forms (e.g., wont to won't, gimme to give me); and removing spaces between words and their standard contractions (e.g., is n't to isn't).
4. Punctuation processing was then performed, which included normalizing spacing around punctuation marks by removing excessive preceding spaces and ensuring a single trailing space if followed by an alphanumeric character (excluding intra-word apostrophes). Redundant spaces between consecutive punctuation marks were eliminated, and sequences of identical punctuation marks were merged into a single instance (e.g., ... to .).
5. Finally, overall whitespace standardization was conducted by trimming all leading and trailing whitespace from the text and consolidating multiple consecutive internal whitespace characters (spaces, tabs, newlines) into a single space. This meticulous preprocessing pipeline ensured a consistent and clean dataset for all subsequent experimental stages.

2. Methodology

2.1 TF-IDF with Traditional Machine Learning Classifiers

In this phase of the experiment, the preprocessed text data was first transformed into numerical features using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique. Subsequently, a diverse set of machine learning classifiers were trained and evaluated to assess their effectiveness in predicting text polarity. The performance of these models was systematically compared across different max_features settings for the TF-IDF vectorizer, specifically at 1000, 3000, and 5000 features. The classifiers employed include:

- Logistic Regression: A linear model that uses a logistic function to model the probability of a binary outcome (positive/negative

sentiment). It's known for its simplicity and interpretability.

- Support Vector Machine - Linear Kernel (SVM - Linear Kernel): A supervised learning model that finds an optimal hyperplane to separate data points into different classes. A linear kernel is used when the data is expected to be linearly separable.
- Ridge Classifier: A linear classification model that incorporates L2 regularization (Ridge regression) to prevent overfitting by penalizing large coefficients.
- Extra Trees Classifier (Extremely Randomized Trees): An ensemble learning method similar to Random Forest, but it introduces more randomness in the way splits are chosen in trees, often leading to reduced variance.
- Random Forest Classifier: An ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes of the individual trees. It's robust against overfitting.
- Decision Tree Classifier: A non-parametric supervised learning method that creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
- Gradient Boosting Classifier: An ensemble technique that builds models in a sequential manner, where each new model corrects errors made by previous models. It's a powerful method known for high accuracy.
- Light Gradient Boosting Machine (LightGBM): A gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with advantages in training speed and memory usage, particularly on large datasets.
- K Neighbors Classifier (K-Nearest Neighbors): A non-parametric, instance-based learning algorithm where classification is determined by a majority vote of its 'k' nearest neighbors in the feature space.
- Ada Boost Classifier (Adaptive Boosting): An ensemble learning algorithm that combines multiple weak learners (typically decision stumps) sequentially, giving more weight to instances that were misclassified by previous learners.
- Dummy Classifier: A baseline classifier that makes predictions using simple rules, such as always predicting the most frequent class or generating random predictions. It serves as a sanity check to ensure other models are performing better than chance or a naive strategy.

2.2 Transformer-Based Language Models for Sentiment Classification

In addition to classical machine learning approaches, this study investigated the efficacy of several state-of-the-art transformer-based language models for the text polarity classification task[1]. These models are pre-trained on vast amounts of text data and are designed to capture complex contextual relationships within the text, which is often crucial for nuanced sentiment understanding.

Unlike the TF-IDF approach, these models typically benefit from retaining more of the original textual information. Certain preprocessing steps, while common in traditional NLP, can be detrimental to the performance of these transformer models. Specifically:

- Stop Word Removal: Words like 'a', 'the', 'is', and 'not', though often removed in traditional NLP, provide important contextual information for BERT-like models. Negation words such as 'not' are particularly critical for sentiment determination. Therefore, removing stop words is not recommended[2], [3].
- Stemming or Lemmatization: These techniques reduce words to their base forms but may lead to a loss of subtle meaning nuances (e.g., "amazing" vs. "amazed"). Transformer models like BERT are capable of handling different word forms effectively. Thus, stemming or lemmatization is generally not required[2], [3].

The following pre-trained language models from the Hugging Face Transformers library, detailed in Table 1, were selected for fine-tuning and evaluation.

Model Type	Model Name	Parameters (M)	Brief Introduction
BERT	BERT Base	~110	Good baseline model; Cased version may capture important case-sensitive signals for sentiment.
	DistilBERT Base	~67	High efficiency (faster, smaller footprint) with acceptable performance.
	DistilBERT Base, Fine-tuned on SST-2 English	~67	Pre-fine-tuned for SST-2 binary sentiment classification; offers ready-to-use high efficiency.
RoBERTa	RoBERTa Base	~125	High-performance potential (often achieves state-of-the-art results after fine-tuning); robust pre-training.
	XLM-RoBERTa Base	~279	Multilingual capabilities with strong cross-lingual transfer performance.
XLNet	XLNet Base	~110	High-performance potential; its permutation language modeling can capture diverse contextual dependencies.
ELECTRA	ELECTRA Base (Fine-tuned for Emotion)	~110	Fine-tuned for a 6-category emotion classification task; the base ELECTRA model is known for strong performance and efficient pre-training.

Table 1: Overview of Transformer-Based Language Models Employed

For each of these models, experiments were conducted by fine-tuning them on the augmented and preprocessed sentiment dataset, exploring different hyperparameters such as batch size, and the amount of training data to optimize performance.

3. Evaluation Methods

The performance of all models developed in this study for the binary text polarity classification task was evaluated using a single, primary metric: Accuracy.

Accuracy is a widely used metric that measures the proportion of total predictions that were correct. In the context of this sentiment classification task (predicting positive or negative polarity), it is calculated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Or, more formally, using the components of a confusion matrix:

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

Where:

- TP (True Positives) = Number of positive instances correctly classified as positive.
- TN (True Negatives) = Number of negative instances correctly classified as negative.
- FP (False Positives) = Number of negative instances incorrectly classified as positive.
- FN (False Negatives) = Number of positive instances incorrectly classified as negative.

This metric provides a straightforward and intuitive measure of the overall effectiveness of the classifiers in correctly assigning sentiment

labels to the text data. All comparisons between different models, feature sets (for TF-IDF), and hyperparameter configurations (for transformer models) were based on their achieved accuracy scores on the designated test or validation sets.

4. Experiment Result

4.1 Performance of TF-IDF with Traditional Machine Learning Classifiers

The initial set of experiments focused on evaluating traditional machine learning algorithms using TF-IDF vectorized text data. Two key parameters of the TF-IDF vectorizer were tuned: the maximum number of features (max_features) and the n-gram range (ngram_range).

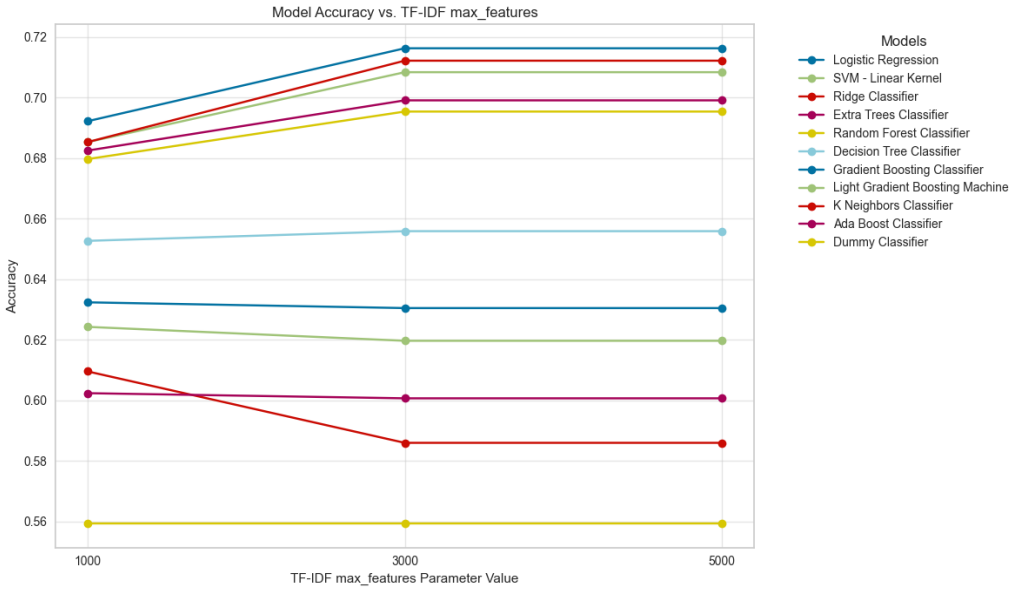

Figure 6. Model Accuracy as a Function of TF-IDF Maximum Features

Several key observations can be made from From Figure 6, which displays model accuracy against varying TF-IDF max_features (with n-gram range (1,1)):

- Top Performers: Logistic Regression, SVM - Linear Kernel, Ridge Classifier, and Light Gradient Boosting Machine consistently achieved the highest accuracies. Their performance improved notably when max_features increased from 1000 to 3000 (accuracies around 0.710-0.717), with marginal gains or a plateau at 5000 features.
- Mid-Tier and Lower Performers: Decision Tree and Gradient Boosting classifiers showed moderate, stable accuracies (around 0.650-0.655). Other models like Extra Trees, Random Forest, K Neighbors, and Ada Boost generally had lower and sometimes declining accuracies as features increased.
- Baseline Confirmation: The Dummy Classifier's consistent accuracy around 0.56 confirmed that other models were effectively learning from the data.

Overall, for the best-performing models, utilizing 3000 to 5000 features appeared optimal, as further increases did not yield substantial gains in accuracy. For subsequent n-gram tuning, max_features was set to 5000.

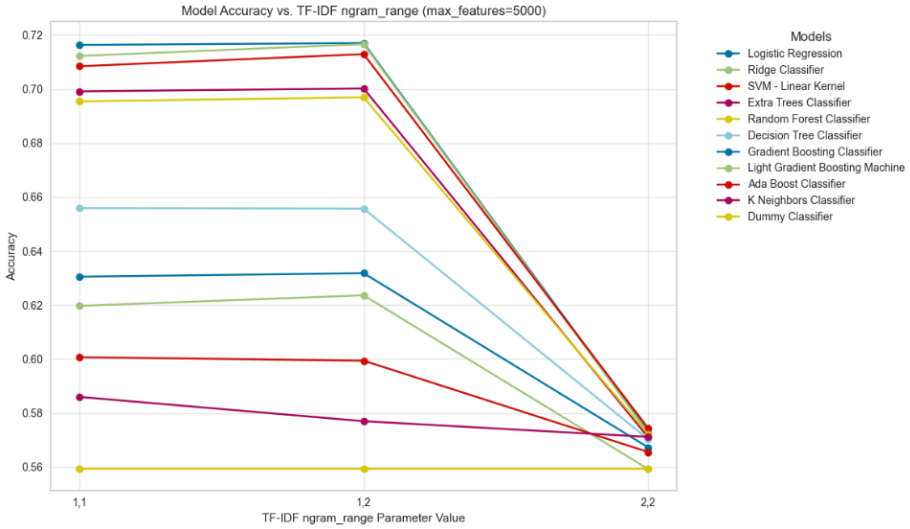

Figure 7. Model Accuracy as a Function of TF-IDF N-gram Range (with max_features=5000)

Figure 7 shows model accuracies with different TF-IDF n-gram ranges, using max_features=5000:

- Unigrams and Bigrams: Most models, particularly top performers like Logistic Regression, SVM - Linear Kernel, and Ridge Classifier, achieved their highest accuracies (around 0.717) using only unigrams (1,1). Including bigrams (1,2) offered similar or marginally lower performance for these top models, while Light Gradient Boosting Machine saw a slight drop.
- Bigrams Only Detrimental: Using only bigrams (2,2) caused a significant drop in accuracy across all models, indicating that while bigrams might offer some context with unigrams, they are insufficient alone for this dataset and feature size.
- Consistent Model Ranking: The relative performance ranking of models remained largely consistent across these n-gram settings.

These results suggest that for the TF-IDF approach with traditional machine learning models, using unigrams, potentially augmented by bigrams, with a feature set of 3000 to 5000, provides the most promising results. Using only bigrams significantly degrades performance.

4.2 Performance of Transformer-Based Language Models for Sentiment Classification

This section details the performance evaluation of various Transformer-based language models on the sentiment classification task. We systematically investigate the impact of key hyperparameters, including batch size, data augmentation size on model performance. The evaluation metrics primarily focus on accuracy.

4.2.1 Optimal of Batch Size

To assess the influence of batch size on model performance, an initial set of experiments was conducted. In this phase, the batch size was varied across values of 16, 32, and 64, while other key hyperparameters were held constant to isolate the effect of batch size. Specifically, based on a preliminary analysis indicating that the majority of text instances in our dataset have a length between 20 and 30 tokens, the maximum sequence length was fixed at 32. This value was chosen to adequately cover the typical input length while minimizing unnecessary computational overhead. Furthermore, the data augmentation size was maintained at 20000 for all runs in this experimental stage. This augmentation level was selected because initial explorations revealed that increasing data augmentation up to this point significantly enhanced model performance, and 20000 served as a balanced median value within our experimental conditions for further parameter tuning.

This consistent data augmentation size ensures that observed variations in performance are attributable to changes in batch size rather than differing scales or diversity of the training data. The results of these experiments are presented in Table 2.

Model Type	Model Name	Batch Size		
		16	32	64
BERT	BERT Base	0.9061	0.9055	0.9080
	DistilBERT Base	0.9061	0.9059	0.9033
	DistilBERT Base, Fine-tuned on SST-2 English	0.9497	0.9497	0.9544
RoBERTa	RoBERTa Base	0.9183	0.9149	0.9086
	XLM-RoBERTa Base	0.8996	0.9007	0.9035
XLNet	XLNet Base	0.9207	0.9212	0.9075
ELECTRA	ELECTRA Base (Fine-tuned for Emotion)	0.9109	0.9082	0.9053
Average		0.9159	0.9151	0.9129

Table 2. Sentiment Classification Accuracy of Transformer Models with Varying Batch Sizes

Table 2 details the sentiment classification accuracies of the various Transformer models under different batch sizes. A key observation is that for most of the models tested, the performance difference between using a batch size of 32 and 64 was generally not substantial. On average, a batch size of 32 yielded a slightly higher accuracy (0.9151) compared to a batch size of 64 (0.9129). While individual models exhibited minor preferences—for instance, DistilBERT Base, Fine-tuned on SST-2 English performed marginally better with a batch size of 64, whereas XLNet Base favored a batch size of 32—many models, such as BERT Base and XLM-RoBERTa Base, demonstrated very comparable accuracies across these two settings. This indicates that under the fixed experimental conditions of a maximum sequence length of 32 and a data augmentation size of 20000, there is often limited substantial impact on final performance when choosing between a batch size of 32 or 64.

4.2.2 Optimal of Data Augmentation Size

Following the batch size evaluation in Section 4.2.1, which indicated that a batch size of **32** offered a generally robust performance, this value was adopted for the experiments in this section. The maximum sequence length was also kept constant at **32**. This stage focuses on determining the impact of varying the data augmentation size on model accuracy for the sentiment classification task. We tested augmentation sizes of 10,000, 20,000, and 40,000 samples, with the results presented in Table 3.

Model Type	Model Name	Data Augumentation Size		
		10000	20000	40000
BERT	BERT Base	0.8851	0.9056	0.9240
	DistilBERT Base	0.8862	0.9055	0.9221
	DistilBERT Base, Fine-tuned on SST-2 English	0.9428	**0.9497**	**0.9584**
RoBERTa	RoBERTa Base	0.8910	0.9161	0.9235
	XLM-RoBERTa Base	0.8700	0.9035	0.9012
XLNet	XLNet Base	0.8983	0.9247	0.9167
ELECTRA	ELECTRA Base (Fine-tuned for Emotion)	0.8767	0.9091	0.9224
Average		0.8928	0.9163	0.9240

Table 3. Sentiment Classification Accuracy of Transformer Models with Varying Data Augmentation Sizes

As detailed in Table 3, increasing the data augmentation size generally provided a slight improvement in accuracy, although the magnitude of these gains often diminished at higher augmentation levels. For example, DistilBERT Base, Fine-tuned on SST-2 English achieved its best accuracies of 0.9497 and 0.9584 with 20,000 and 40,000 augmentations, respectively. Considering these diminishing returns and practical time constraints, further increases beyond 40,000 augmentations were not explored. For a related Kaggle competition, model configurations based on the two highest-performing results observed from DistilBERT Base, Fine-tuned on SST-2 English were utilized.

5. Conclusion

This study enhanced text polarity classification through data augmentation, rigorous preprocessing, and comparative analysis of TF-IDF with traditional models against advanced transformer architectures. Transformer-based models, particularly "DistilBERT Base, Fine-tuned on SST-2 English," demonstrated superior local validation accuracies, reaching up to 0.9584.

However, a significant performance drop was observed in the Kaggle competition, where this model scored around 0.70. This discrepancy likely stems from a shift between our augmented local data and the Kaggle dataset's true distribution, potentially leading to overfitting despite augmentation efforts. The characteristics of the external data used for augmentation, while filtered, might not have perfectly matched the competition's specific context, impacting generalization.

To bridge this gap, future efforts could focus on more targeted data augmentation and robust validation strategies better aligned with the unseen Kaggle data. Exploring domain adaptation techniques or even different transformer architectures, which might offer varied robustness to dataset shifts or benefit from distinct fine-tuning approaches, could also be beneficial. Additionally, detailed error analysis and refined hyperparameter tuning aimed at enhancing generalization are advisable. Ultimately, while transformer models show great promise, translating high local performance to challenging, unseen datasets requires meticulous attention to data representativeness and model robustness.

6. Reference

[1] D. Cortiz, “Exploring Transformers in Emotion Recognition: a comparison of BERT, DistillBERT, RoBERTa, XLNet and ELECTRA,” Apr. 05, 2021, *arXiv*: arXiv:2104.02041. doi: 10.48550/arXiv.2104.02041.

[2] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, “Understanding the Behaviors of BERT in Ranking,” Apr. 26, 2019, *arXiv*: arXiv:1904.07531. doi: 10.48550/arXiv.1904.07531.

[3] E. Alzahrani and L. Jololian, “How Different Text-preprocessing Techniques Using The BERT Model Affect The Gender Profiling of Authors,” Sep. 28, 2021, *arXiv*: arXiv:2109.13890. doi: 10.48550/arXiv.2109.13890.