Robustness of Transformer-Based Models Against Linguistic Noise and Adversarial Inputs in Social Media Sentiment Tasks

Benjamin IBOK – 15323543
Data Science & Computational Intelligence
Coventry University
United Kingdom

Abstract

This study investigates the robustness of Transformer-based, Deep Learning models, specifically BiLSTM and BERT, against linguistic noise in social media sentiment analysis tasks. Using Twitter datasets from SemEval (2015 and 2017), the performance of a traditional BiLSTM model trained on GloVe embeddings is compared with a pre-tuned BERT model. Both models were trained and tested using the 2017 dataset, validation was done using the SemEval-2015 dataset, then evaluated on a clean 20% test split, and evaluated on clean and noisy inputs, where the noise was introduced through character-level perturbations to simulate real-world typing errors. Results show that while BERT achieved higher accuracy and F1-score on clean data, it also demonstrated superior robustness under noise, outperforming BiLSTM significantly, which suffered a major performance drop under the same conditions. These findings underscore BERT's effectiveness and resilience for sentiment analysis in informal, noisy text environments, highlighting its suitability for deployment in real-world social media monitoring systems.

Code Link:
https://github.com/JaminUbuntu/IBOK_NLP_2-CW/blob/main/IBOK_NLP_DL_CW.ipynb

1 Introduction

Sentiment analysis on social media platforms, such as Twitter, has become a vital task in Natural Language Processing (NLP) (Omuya et al., 2023), providing valuable insights into public opinion, brand perception, and social trends. However, the informal and noisy nature of tweets, characterized by abbreviations, emojis, spelling errors, and grammatical inconsistencies, poses significant challenges for text classification models (Khan et al., 2025). Transformer-based architectures, particularly BERT, have shown state-of-the-art performance in sentiment analysis due to their ability to capture contextual semantics (Bashiri & Naderi, 2024). Despite their success on clean benchmark datasets, recent studies suggest that these models may be vulnerable to even minor perturbations in input, such as typographical errors or adversarial manipulations (Wang et al., 2021). This research aims to determine which architecture better maintains performance when exposed to realistic, linguistically degraded input. By assessing performance across standard metrics and robustness to noise, this study contributes to the broader goal of developing fault-tolerant NLP models suitable for deployment in noisy, real-world environments like Social Media Platforms or chatbots.

2 Literature Review

In a study, (Albladi et al., 2025) presents a systematic review of research on sentiment analysis using NLP models, with a specific focus on Twitter data. Various approaches and methodologies were discussed, including Machine Learning, Deep Learning, and Hybrid Models, with their advantages, challenges, and performance metrics. They also identify key NLP models commonly employed, such as transformer-based architectures like BERT and GPT.

Another study (Dang et al., 2024) proves a strong correlation between training data and the robustness of Transformer textual models. Researchers extract 13 features from input fine-tuning corpora properties and use them to predict the adversarial robustness of the models. They focus on Encoder-only Transformer models, BERT, and RoBERTa, and provide evidence that extracted features can be used with Lightweight Classifiers like Random Forest to predict attack success rates.

Sentiment Analysis (SA) is a widely used technique for extracting useful and subjective information from text-based data (Wankhade et al., 2022). However, GloVe and Word2vec embedding models have been widely used for feature extractions, but they overlook sentimental and contextual information. (Tabinda Kokab et al., 2022) proposes a generalized SA model that can handle noisy data, Out of Vocabulary Words (OOV), and sentimental and contextual loss of reviews data. The research proposes an effective Bi-directional Encoder Representation from Transformers (BERT) based Convolution Bi-directional Recurrent Neural Network (CBRNN) model for exploring syntactic and semantic information, along with sentimental and contextual analysis of the data.

3 Dataset Description

The datasets used in this study consist of Twitter sentiment data from the SemEval competition, link can be found in the appendix section. This work combines the 2017 edition of the dataset and the 2015 dataset, where the first dataset was split into 70/20 for the train and test sets, while the 2015 dataset was prepared for use as the validation set.

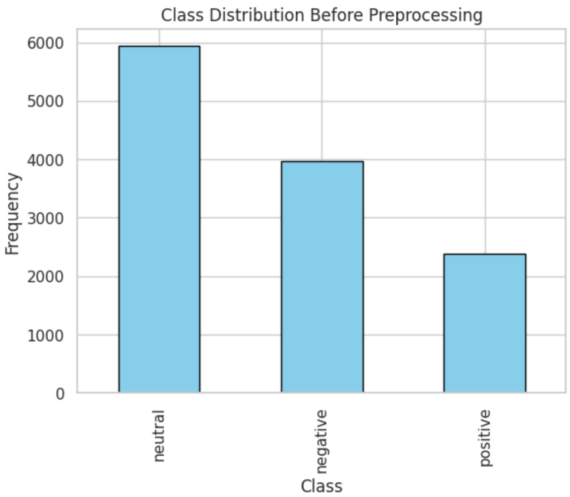

Fig. 1. Distribution of the Target Categories

The dataset contained no null values and no duplicates. All the datasets were cleaned and preprocessed uniformly, and labels were encoded using a single LabelEncoder to ensure consistent numerical mappings across sets. The distribution of the target variable categories is shown in Fig. 1.

4 Methodology

4.1 Transformer-Based Model: BERT

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art pre-trained language model developed by Google. It captures the bidirectional context of words by jointly conditioning on both the left and right surroundings in all layers. In this project, BERT is fine-tuned on the SemEval-2017 Twitter sentiment dataset to perform Target variable classification into positive, negative, or neutral sentiment categories.

BERT was selected due to its relevance to real-world sentiment classification, especially in noisy and informal social media text, and the labels from the dataset align with the motivational objective of evaluating model robustness under varying linguistic conditions.

4.2 Recurrent Neural Network: BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) networks are an extension of standard LSTM models that read input sequences both forward and backward, capturing dependencies from both directions. In this project, BiLSTM is employed as a traditional deep learning baseline for comparison with BERT. BiLSTM's inclusion is justified as a widely accepted baseline for sequence modeling, allowing us to assess the effectiveness of Transformer-based models in comparison. Its simpler structure also enables us to observe how noise and variability in social text affect traditional recurrent models, especially under low-resource and imbalanced class scenarios.

4.3 Pretrained Word Embeddings

Pretrained Word Embeddings, such as GloVe (Global Vectors), transform words into dense vector representations based on their co-occurrence in large corpora. These embeddings capture semantic relationships and are essential in traditional Deep Learning architectures like BiLSTM. In this project, they serve as static input vectors for the BiLSTM model, allowing it to learn from meaningful word representations without training embeddings from scratch. In this analysis, the glove glove.6B.100d.txt file was uploaded as the static input. The file glove.6B.100d.txt contains pre-trained GloVe (Global Vectors for Word Representation) embeddings with 100-dimensional word vectors, trained on 6 billion tokens from a Wikipedia + Gigaword corpus. It was uploaded to serve as the embedding layer for the BiLSTM model.

4.4 Tensorflow

TensorFlow, developed by Google, powers many real-world Machine Learning applications due to its scalability, performance, and broad ecosystem (Abadi et al., 2016). In this study, TensorFlow, paired with its high-level Keras API, to construct and train the BiLSTM model. Its intuitive sequential model builder, rich layer customization, and GPU support made model development efficient and reproducible.

4.5 HuggingFace Library

While BiLSTM was built using TensorFlow/Karas**,** BERT was fine-tuned and evaluated using PyTorch, via HuggingFace**,** combining the strengths of both frameworks for efficiency and flexibility. The HuggingFace library played a central role in fine-tuning the BERT model, offering a powerful and user-friendly interface for working with state-of-the-art Transformer architectures, it provided pre-trained models, tokenizers, and training utilities built on top of PyTorch, allowing for efficient implementation of text classification tasks.

4.6 PyTorch

PyTorch is a deep learning library developed by Facebook that stands out for its dynamic computation graph and Pythonic design (Jha & Pillai, 2021). PyTorch was applied through the HuggingFace Transformers library to fine-tune the BERT model for sentiment classification. It combined with HuggingFace's Trainer API to provide streamlined training, evaluation, and integration, with logging tools like Weights and Biases. It offers better control and reproducibility, especially for Transformer based architectures and real-time debugging (Sawarkar, 2022).

4.7 Class Weighting

In classification tasks, especially with real-world datasets like SemEval-2017 Twitter sentiment data, class imbalance can significantly skew model performance. Rather than synthetically altering the dataset, this study adopted Class Weight balancing**,** where misclassification penalties are adjusted during training based on class frequency. This approach differs from Over-Sampling methods like SMOTE (Synthetic Minority Oversampling Technique) or Under-Sampling methods like Random Under-Sampling (RUS)**.**

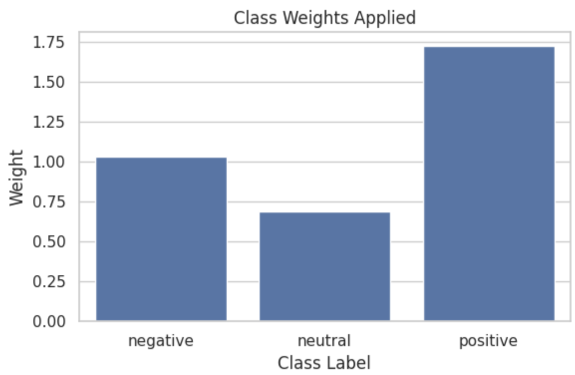

Fig. 2. Applying Class Weighting to the Target Categories

5 Experiment / Results

5.1 Text Preprocessing

During this study, a light text preprocessing was applied to clean and normalize noisy Twitter data. This included converting text to lowercase, removing URLs, user mentions, hashtags, punctuation, digits, and extra whitespaces. These steps ensured consistency and reduced noise that could mislead models. For token-based models like BERT, Hugging Face's tokenizer handled subword tokenization, while for BiLSTM, Keras tokenization and GloVe embeddings were used. Additionally, label encoding converted categorical sentiment labels into numeric format. Preprocessing was essential for improving model learning, reducing overfitting, and enhancing the robustness and fairness of evaluations across clean and noisy text scenarios. Fig. 3 shows a WordCloud of the text set with the initial noise.

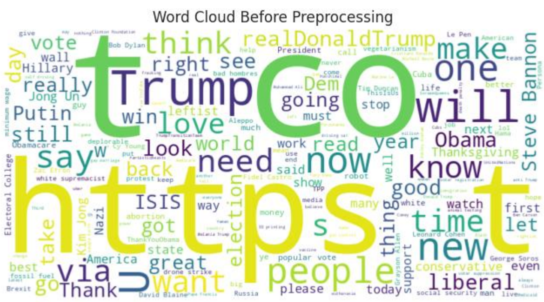

Fig. 3. Word Cloud of the Training/Testing dataset

5.2 Robustness Testing via Linguistic Noise

To evaluate the resilience of models to adversarial or real-world distortions, nlpaug linguistic noise was injected into the test set, keyboard-style typos were introduced using the keyboardAug augmenter, which simulates user typing errors common on social platforms. The noisy test set was then fed to the BERT model, and performance was re-evaluated. This step is justified as real-world NLP systems often encounter imperfect input, misspellings, typos, and slang, furthermore, the BiLSTM was subjected to other kinds of noise, such as synonyms, as a form of semantic-preserving adversarial testing, Table 1. shows the result of that analysis. Evaluating under these conditions reveals how well a model generalizes beyond clean training data, and the results showed noticeable performance drops, validating the research goal of testing model robustness, such findings are essential when deploying sentiment analysis tools in live environments where input quality is not guaranteed.

	Accuracy	Precision	Recall	F1
Clean	0.332339	0.380744	0.332339	0.344482
Typo	0.321215	0.377193	0.321215	0.329885
Syno	0.32013	0.366877	0.320130	0.332247

Table 1.Showing the results compared with synonyms.

5.3 Data Visualization

During the experiment, a series of data visualization techniques were used to uncover the inconsistencies in the data, the histogram in Fig 4. illustrates the distribution of tweet lengths in characters in the 2017 SemEval Twitter dataset. The x-axis represents tweet lengths, while the y-axis shows their frequency. The distribution is clearly right skewed, with the majority of tweets clustered between 120 and 140 characters, the historical character limit for Twitter at the time. This indicates that users often utilized the full character space available to them. And from the

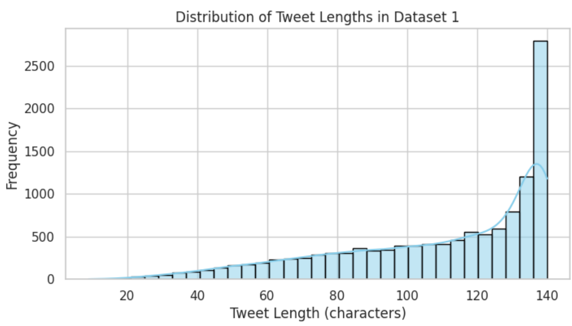

Fig. 4 Distribution of tweet lengths in characters.

plot, there is a gradual increase in frequency as tweet length increases, peaking sharply at the 140-character mark, confirming that many users wrote tweets as long as the platform allowed. The KDE (Kernel Density Estimation) line overlaid suggests a non-normal, positively skewed distribution.

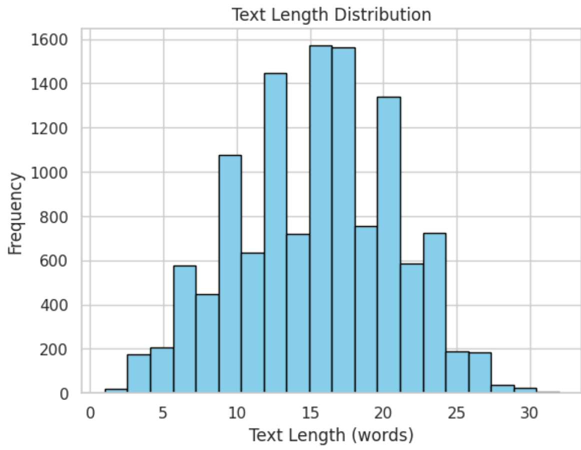

Fig. 5 Distribution of text lengths before tokenization.

The histogram in Fig. 5 displays the distribution of text lengths in words before tokenization and cleaning. The x-axis represents the number of words per tweet, while the y-axis indicates frequency.

5.4 Justification for the Approach

The adopted approach in this project, comparing a traditional BiLSTM model with a transformer-based BERT model and applying noise to both, and test for robustness, was carefully chosen to enable a comprehensive evaluation of performance and robustness in sentiment analysis on social media data. By training both models on the same datasets and subjecting the models to clean and noisy inputs, their baseline performance was assessed, as well as their resilience to linguistic distortions. The use of class weighting to address label imbalance, label encoding, and thorough data visualization ensured fairness and insights. This dual-model, noise-augmented framework provided valuable insights into the generalization capabilities and fault tolerance of static versus contextual embedding-based architectures in sentiment classification. The goal for this study is to contribute to the development of fault-tolerant NLP systems that maintain their predictive quality even when faced with messy, unpredictable user input, making AI more resilient, trustworthy, and effective in real-world social media analytics.

6 Model Architecture

6.1 The Embedding Layer (BiLSTM)

The BiLSTM model uses pre-trained static word embeddings from the GloVe text file, which provide 100-dimensional vector representations for each token. Tweets are tokenized using Keras' Tokenizer, and an embedding matrix is created to map token indices to GloVe vectors. The embedding layer is initialized with these weights and set to trainable, allowing domain adaptation during training.

Following the embedding layer, a BiLSTM layer captures both forward and backward temporal dependencies within tweet sequences. This bidirectional setup enables the model to effectively understand contextual polarity and sentiment shifts that occur within short Twitter messages. Dropout is applied to reduce overfitting.

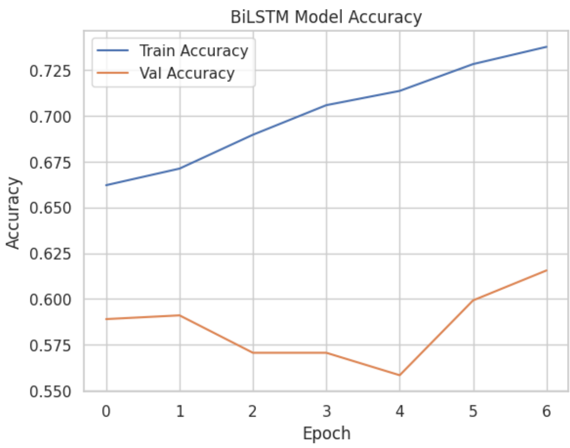

Fig. 6 Training Curve Plot of BiLSTM.

6.2 BERT Transformer

The BERT model leverages HuggingFace's BertForSequenceClassification. Using pre-trained bert-base-uncased weights, it processes tokenized input via BertTokenizer, which includes special tokens and segment embeddings. A softmax classification head is added for multi-class output. BERT's contextualized embeddings allow dynamic interpretation of token meaning based on surrounding text, key to its robustness in noisy environments.

6.3 Training Strategy

Both models are trained on the SemEval-2017 Task 4A dataset with validation on the 2015 dataset. BiLSTM training uses TensorFlow with categorical cross-entropy loss, while BERT uses HuggingFace's Trainer with Adam optimizer and standard learning rate scheduling, Class weights are applied to handle class imbalance.

6.4 Loss Curve Analysis

The graph in Fig. 7 shows a steadily decreasing training loss, indicating that the model is successfully learning from the data. The red line superimposed on the plot illustrates the overall downward trend, a positive sign of convergence, demonstrating that the BERT model fine-tuning was successful, and the loss steadily declined, fluctuations reduced over time, and the model did not overfit or diverge, also that the model is ready for evaluation and robustness testing.

Fig. 7 The Loss Curve Graph showing convergence.

6.5 Implementation Workflow

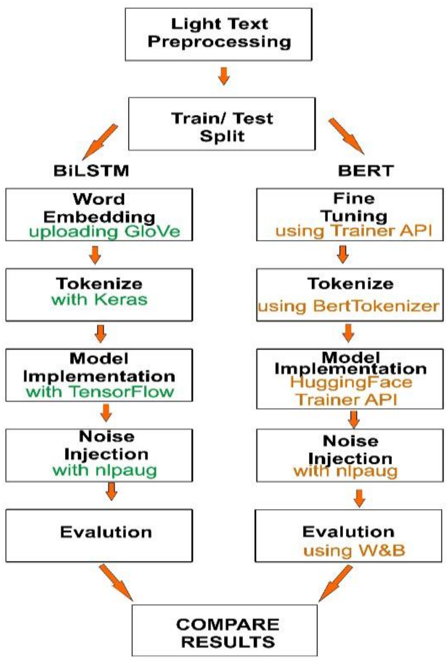

Fig 8 Implementation Framework

The implementation phase of this project was carried out using Python with a combination of TensorFlow, PyTorch, HuggingFace Transformers, and supporting libraries such as, sklearn, pandas and nlpaug on Google Colab. The notebook was structured to allow the development, training, evaluation, and robustness testing of two deep learning models: a BiLSTM network and a BERT-based classifier. The process began with the installation and configuration of the required packages. The HuggingFace Transformers Library was used to fine-tune the BERT model, while TensorFlow's Keras' API was employed for constructing and training the BiLSTM model, experiment tracking and metric logging were managed using Weights & Biases (wandb) (Fig. 9). For the BiLSTM implementation, the training data was tokenized using Keras Tokenizer, and input sequences were padded to a fixed length. The model was initialized with a pre-trained glove embedding matrix, followed by a Bidirectional LSTM layer, dropout for regularization, and a dense output layer with Softmax activation. The model was compiled with categorical cross-entropy loss and trained using the Adam optimizer. In contrast, the BERT model was implemented using the BertForSequenceClassification class from HuggingFace. Input tweets were tokenized using BertTokenizer, and formatted as PyTorch tensors to make them compatible with the HuggingFace Trainer API, and training was performed using the TrainingArguments object. Both models were evaluated using classification metrics, such as Accuracy, Precision, Recall, F1-score, and AUC. The model's performances under clean and noisy input conditions were compared and visualized using confusion matrices and classification reports, the illustration in Fig. 8 is a block diagram representation of the implementation workflow.

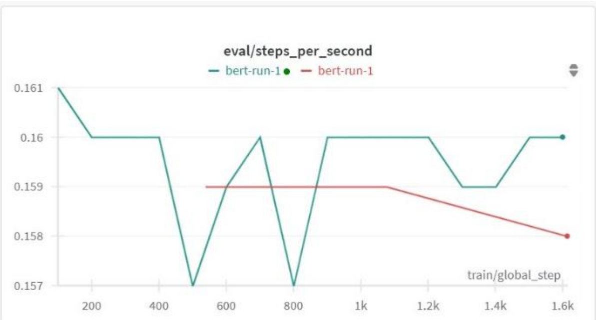

Fig 9. Metric Logging using Wandb.

7 Discussion / Analysis

In the experiment, two datasets were used to train the base model, one of the data, the BiLSTM training curve reveals overfitting after 4 epochs, with training accuracy rising while validation accuracy dips. This aligns with its lower generalization capability.
The results of this study reveal critical insights into the comparative performance and robustness of BiLSTM and BERT models for sentiment analysis on Twitter data. Initially, both models were evaluated on a clean 20% test split. The BERT model outperformed BiLSTM, achieving a higher AUC of 0.8347 vs. 0.7752 and F1-score of 0.6962 vs. 0.6021, demonstrating its superior capacity to capture semantic and contextual information from short, informal tweets.

Model	**Accuracy**	**F1**	**Precision**	**Recall**	**AUC**
BiLSTM	0.600109	0.602115	0.625013	0.600109	0.775224
BiLSTM(*)	0.321215	0.329885	0.377193	0.321215	0.768961
BERT	0.698047	0.696229	0.700973	0.698047	0.834754
BERT(*)	0.632664	0.622887	0.640962	0.632664	0.768961

Table 2.Showing the final results.

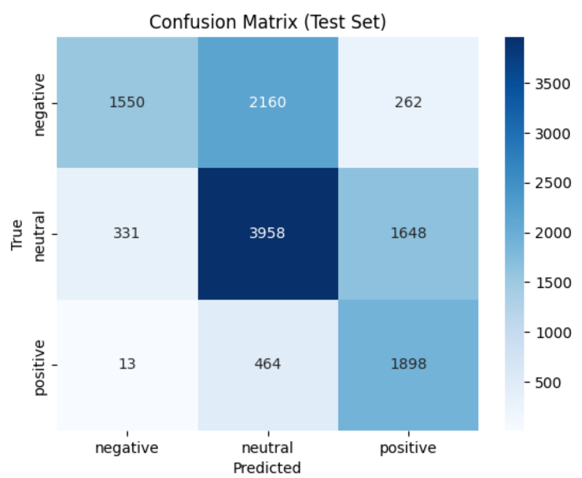

Fig. 10 Confusion Matrix of the clean BERT Model.

The confusion matrix (Fig. 10.) for BERT further confirms that on clean data, BERT predicts well across classes, but under noise, positive and neutral sentiments suffer, showing higher misclassifications, especially in the positive class. The confusion matrix for BERT in Fig. 10 on the noisy test set reveals a clear decline in classification accuracy due to linguistic distortions.

While the model correctly classified many neutral tweets, it struggled significantly with positive and negative sentiments. A large portion of negative tweets were misclassified as neutral, and a substantial number of positive tweets were incorrectly predicted as neutral or negative.

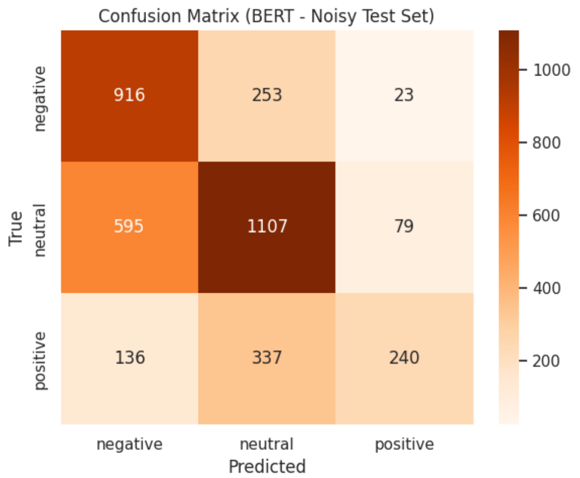

Fig. 11. Confusion Matrix of the Bert Model with noise.

7.1 Summary of overall Results

The performance comparison chart shows that BERT outperforms BiLSTM across all metrics on clean test data, with higher accuracy (~0.69 vs ~0.62), F1, and AUC scores. However, BERT's performance drops under noisy input, especially in F1 score and accuracy, nearing BiLSTM levels. The BiLSTM model, although less accurate overall, shows less degradation under noise, likely due to its simpler, more robust structure.

Fig. 12. Distribution of the overall metrics.

8 Justification for the Study

The aim of this research is to evaluate and compare the robustness of Transformer-based models, particularly BERT, against linguistic noise and adversarial inputs in social media sentiment analysis tasks. By benchmarking BERT against a traditional deep learning model (BiLSTM), the study investigates how each model performs on real-world Twitter data under both clean and distorted conditions. BERT is justified in this study as it represents the state-of-the-art in NLP, providing a robust benchmark against traditional models like BiLSTM. It serves as the core model for testing adversarial resilience, given its architectural complexity and contextual understanding.

9 Further Studies

Further analysis could explore the integration of more contextual data augmentation techniques, such as paraphrasing or back-translation, across different models to assess deeper semantic robustness. Additionally, incorporating multilingual datasets can evaluate model generalization across diverse linguistic landscapes, extending the analysis to other transformer variants like RoBERTa or DistilBERT could reveal trade-offs between robustness and computational efficiency.

10 Conclusion

This study assesses how well these models generalize beyond ideal conditions and retain their performance under linguistic perturbations. By simulating realistic noise and comparing the performance of static and contextual embedding-based models, the research seeks to identify which architecture is more reliable for deployment in real-time applications, which could also be used to guide against text attacks or perform contextual pattern matching. Applications include Chatbots, Input Validation Systems, Content Security, Phishing Detection Systems, AutoCorrect Implementations.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., … Zheng, X. (2016). *TensorFlow: A System for Large-Scale Machine Learning*.

Albladi, A., Islam, M., & Seals, C. (2025). Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review. *IEEE Access*, *13*, 30444–

30468. https://doi.org/10.1109/ACCESS.2025.3541494

Bashiri, H., & Naderi, H. (2024). Comprehensive review and comparative analysis of transformer models in sentiment analysis. *Knowledge and Information Systems*, *66*(12), 7305–7361. https://doi.org/10.1007/s10115-024-02214-3

Dang, C., Le, D. D., & Le, T. (2024). *A Curious Case of Searching for the Correlation between Training Data and Adversarial Robustness of Transformer Textual Models* (No. arXiv:2402.11469). arXiv. https://doi.org/10.48550/arXiv.2402.11469

Jha, A. R., & Pillai, G. (2021). *Mastering PyTorch: Build powerful neural network architectures using advanced PyTorch 1.x features*. Packt.

Khan, J., Ahmad, K., Jagatheesaperumal, S. K., & Sohn, K.-A. (2025). *Textual variations in social media text processing applications: Challenges, solutions, and trends*.

Omuya, E. O., Okeyo, G., & Kimwele, M. (2023). Sentiment analysis on social media tweets using dimensionality reduction and natural language processing. *Engineering Reports*, *5*(3), e12579. https://doi.org/10.1002/eng2.12579

Sawarkar, K. (2022). *Deep Learning with PyTorch Lightning: Swiftly build high-performance Artificial Intelligence (AI) models using Python* (1st ed.). Packt Publishing Limited.

Tabinda Kokab, S., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, *14*, 100157. https://doi.org/10.1016/j.array.2022.100157

Wang, B., Xu, C., Wang, S., Gan, Z., Cheng, Y., Gao, J., Awadallah, A. H., & Li, B. (2021). *Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2111.02840

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*(7), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

Appendix (Dataset link)

https://github.com/leelaylay/TweetSemEval/tree/master/dataset.