Detecting Deceptive Online Reviews: A Comparative Analysis of NLP Techniques and Machine Learning Models

Ahmad Musayev, Max Stolze, Rais Sheotahul, Semih T. Elkatmış, Thomas S. J. Plugge

March 2025

Abstract

Deceptive online reviews, particularly computer-generated (CG) content, undermine trust in digital platforms. This study investigates the efficacy of machine learning models in distinguishing CG reviews from human-written (OR) reviews by analyzing linguistic patterns and comparing state-of-the-art natural language processing (NLP) techniques. Using a labeled dataset of 40,216 reviews (50% CG, 50% OR), we evaluate traditional and transformer-based approaches. A baseline model combining TF-IDF vectorization (with bi- and trigrams) and logistic regression achieves 90% accuracy. In contrast, the BERT model attains 98% accuracy without tuning and 98.29% after hyperparameter optimization, demonstrating that contextual embeddings significantly outperform traditional NLP methods. Our findings highlight the potential of advanced transformer models to combat fraudulent reviews with near-perfect accuracy.

1 Introduction

In an increasingly digital world, online reviews have become a cornerstone of consumer decision-making and business reputation. These accounts shape perceptions of products and services, from restaurants to electronics, and underpin trust in digital marketplaces[1]. However, the rise of deceptive reviews generated by artificial intelligence threatens this ecosystem. Such content can be used for fraudulent purposes, distorting product evaluations, eroding consumer trust, and also imposes economic costs on businesses[2].

To address these challenges, machine learning (ML) has emerged as a critical tool for detecting deceptive content. ML techniques, fueled by advances in natural language processing (NLP), can analyze linguistic patterns and behavioral cues to distinguish authentic reviews from fraudulent ones [3]. Traditional NLP methods, such as TF-IDF vectorization and n-gram analysis, extract statistical features from text [4], while transformer-based models such as BERT leverage contextual embeddings to capture subtle signs of deception [5].

Despite these advances, few studies have systematically compared the performance of traditional feature-based approaches with state-of-the-art transformers. This study bridges that gap by evaluating a spectrum of methods, from logistic regression with TF-IDF features to fine-tuned BERT, on a balanced dataset of 40,216 computer-generated (CG)

[1] Akesson, J., Hahn, R., Metcalfe, R., & Monti-Nussbaum, M. (2023). The impact of fake reviews on demand and welfare. https://doi.org/10.3386/w31836

[2] Naqa, I. E., & Murphy, M. J. (2015). What is machine learning? In Springer eBooks (pp. 3–11). https://doi.org/10.1007/978-3-319-18305-3_1

[3] Zhou, M., Duan, N., Liu, S., & Shum, H. (2020). Progress in neural NLP: modeling, learning, and reasoning. *Engineering*, *6*(3), 275–290. https://doi.org/10.1016/j.eng.2019.12.014

[4] Qaiser, S., & Ali, R. (2018). Text mining: Use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, *181*(1), 25–29. https://doi.org/10.5120/ijca2018917395

[5] Koroteev, M., V. (2021, March 22). *BERT: A review of Applications in Natural Language Processing and Understanding*. arXiv.org. https://arxiv.org/abs/2103.11943

and human-written (OR) reviews. Our goal is to identify the most accurate and scalable approach for the detection of fake reviews, providing actionable insights for platforms and policy makers.

2 Hypothesis

The study tests the following hypotheses:

- **Hypothesis 1**: Machine learning models can accurately distinguish between computer-generated (CG) and human-written (OR) online reviews by analyzing linguistic and behavioral patterns.
- **Hypothesis 2**: Transformer-based models, specifically BERT, will significantly outperform traditional natural language processing techniques (e.g., TF-IDF with n-gram features and logistic regression) in detecting deceptive content.

2.1 Theoretical Foundation for Hypothesis 1

Hypothesis 1 builds on prior findings that deceptive language systematically differs from truthful language. Studies such as (Ott et al., 2011)[6] have shown that fake reviews often exhibit exaggerated sentiment, vagueness, and unusual syntactic patterns, which make them distinguishable through computational linguistic features. Their results showed that models trained on simple n-gram features could detect deceptive reviews with relatively high accuracy, suggesting that linguistic markers of deception are learnable.

2.2 Theoretical Foundation for Hypothesis 2

Hypothesis 2 aligns with broader trends in natural language processing (NLP), where contextual embeddings from models such as BERT have demonstrated superior performance in text classification tasks. This is particularly evident in applications requiring nuanced semantic and syntactic analysis, such as distinguishing between fake and genuine reviews. The bidirectional context awareness inherent in transformer-based architectures like BERT enables more robust identification of deceptive content compared to traditional feature-based methods, which often rely on surface-level linguistic patterns. Empirical evidence from NLP research consistently supports the effectiveness of such models in capturing subtle distinctions that are critical for accurate classification.

3 Data inspection and preparation

3.1 Data inspection

The dataset consists of 40,216 reviews, evenly split between computer-generated (CG) and human-written (OR) samples. Initial linguistic analysis revealed distinct patterns:

Vocabulary Difference: CG reviews heavily favored transactional terms such as "recommend" and "buy" (Figure 3), while OR reviews used more subjective language (e.g., "really," "time") (Figure 2). Both classes shared some high-frequency words (e.g., "book," "like"), suggesting that these alone are insufficient for classification.

Structural Variations: OR reviews exhibited longer sentences (mean: 16.5 words) compared to CG (mean: 12.7 words), with more variability in length (Figure 2). This aligns with the hypothesis that human writing is more discursive.

3.2 Data preparatiorn

3.2.1 TF-IDF model

The TF-IDF approach implemented extensive linguistic preprocessing. After initial cleaning, each review underwent tokenization, stopword removal, and POS tagging to identify grammatical functions.

[6]Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011, June 1). Finding deceptive opinion spam by any stretch of the imagination. ACL Anthology. https://aclanthology.org/P11-1032/

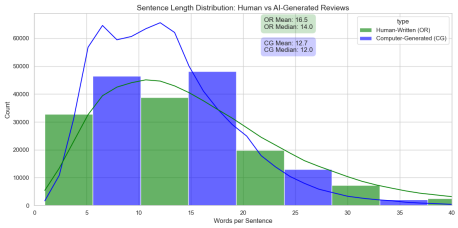

Figure 1: Sentence Length Distribution: Human vs Computer generated reviews

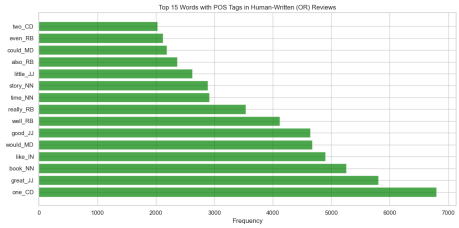

Figure 2: Top 15 words with POS tags for OR Reviews

Lemmatization enhanced with POS information was applied to reduce words to their root forms (e.g., "running" → "run"). Tokens were augmented with their POS tags to provide grammatical context, and n-grams were extracted to capture multi-word expressions. The final transformation utilized TF-IDF vectorization with carefully tuned parameters for feature selection, creating a rich representation that captured lexical, syntactic, and contextual patterns distinguishing human from AI-generated content.

3.2.2 BERT model

For the BERT implementation, minimal but effective preprocessing was employed. After removing entries with missing text, "CG" and "OR" labels were encoded as 1 and 0 respectively, followed by a stratified 80-10-10 split for training, validation, and test sets. The power of BERT lies in its contextual understanding, requiring only tokenization without traditional preprocessing like stemming or stopword removal. Using a pretrained tokenizer, each review was converted to token IDs, truncated to 128 tokens maximum, and appropriately padded. After removing unnecessary columns and converting to PyTorch tensors, the data was prepared for the transformer architecture to learn the contextual patterns differentiating human from AI-generated text.

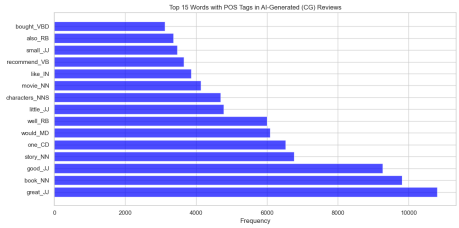

Figure 3: Top 15 words with POS tags for CG Reviews

4 Methods

4.1 Tokenization

The transformation of raw text into discrete tokens is a foundational step in the pipeline. Each review is lowercased and split into individual words, providing the basis for more advanced steps such as lemmatization and POS tagging. Tokenization also aids in identifying common multi-word expressions (n-grams) crucial for detecting stylistic or recurrent language patterns.

4.2 Stopword Removal

Highly frequent yet semantically uninformative words ("the," "and," "of") are filtered out to focus on content-rich tokens. This step curtails noisy features and redirects model attention to linguistically meaningful terms most relevant for distinguishing AI-generated from human-written text.

4.3 Lemmatization

Lemmatization was applied as a preprocessing step to reduce words to their base or root forms. This process minimized the diversity of word forms that express the same concept [10]. For example, variations like "running" and "ran" were consolidated into the single base form "run." This normalization reduced the feature space's complexity and helped ensure the models focused on core meanings rather than superficial differences, thereby improving classification performance.[7]

4.4 Grid Search

To optimize model performance, grid search was employed for hyperparameter tuning [13]. This method systematically evaluated predetermined combinations of parameter values across the entire hyperparameter space. The approach identified the configuration yielding the highest accuracy in detecting fake versus real reviews. This process proved crucial for fine-tuning both traditional models like logistic regression and advanced models such as BERT and XGBoost.[8]

4.5 TF-IDF Vectorization

TF-IDF vectorization was implemented as a natural language processing method for information retrieval. The technique combines Term Frequency (TF), which measures how often a term appears in a document, with inverse document frequency (IDF) which downscales the importance of frequent words while increasing rare ones. The resulting TF-IDF scores for each term were organized into numerical vectors through vectorization [7]. This transformation enabled computational models to efficiently process and analyze the linguistic features captured by TF-IDF, allowing classifiers to focus on the most informative words to distinguish deceptive reviews.[9]

4.5.1 Term Frequency-Inverse Document Frequency

$$\text{TF-IDF}(t,d) = \underbrace{\text{TF}(t,d)}_{\text{Term Frequency}} \times \log\left(\frac{N}{\text{DF}(t)}\right)$$

where:

- t: Term (word or n-gram)
- d: Document (review)
- N: Total number of documents in the corpus
- $\text{TF}(t,d)$: Frequency of term t in document d
- $\text{DF}(t)$: Number of documents containing term t

4.5.2 Vectorization Process

Each document d was represented as a sparse vector:

$$\mathbf{x}_d = [\text{TF-IDF}(t_1,d), \text{TF-IDF}(t_2,d), \ldots, \text{TF-IDF}(t_n,d)] \in R^n$$

where:

- $\mathbf{x}_d$: TF-IDF vector for document d
- n: Vocabulary size (number of unique terms)

4.5.3 Normalization

The final vectors were normalized using Euclidean norm:

$$\mathbf{x}_d^{(\text{norm})} = \frac{\mathbf{x}_d}{\|\mathbf{x}_d\|_2}$$

This ensured all vectors had unit length, improving model stability.

[7] Balakrishnan, V., & Ethel, L. (2014). Stemming and Lemmatization: A comparison of retrieval performances. Lecture Notes on Software Engineering, 2(3), 262–267. https://doi.org/10.7763/lnse.2014.v2.134

[8] Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: A big comparison for NAS. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1912.06059

[9] Abubakar, H. D., & Umar, M. (2022). Sentiment Classification: Review of text vectorization methods: Bag of Words, TF-IDF, Word2VEC and Doc2VEC. SLU Journal of Science and Technology, 4(1 & 2), 27–33. https://doi.org/10.56471/slujst.v4i.266

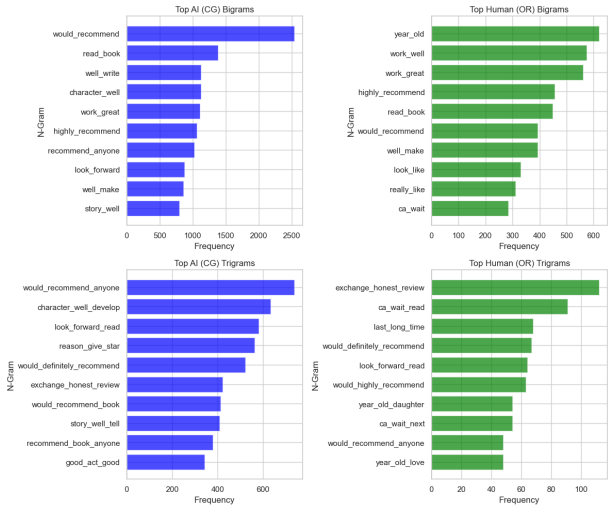

Figure 4: Bigram and Trigram analysis for OR and CG reviews

4.6 N-Gram analysis

While individual words provided valuable information, many deceptive patterns were found to be encoded in recurring phrases rather than isolated tokens. N-gram analysis, which includes unigrams, bigrams, and trigrams, was used to capture word combinations that revealed important contextual patterns[10]. Specific word combinations were identified that potentially indicated automated or artificially constructed reviews. The inclusion of n-gram features improved text representations, enabling models to recognize recurring phrases and stylistic cues characteristic of deceptive evaluations.

4.7 POS Tagging

Part-of-speech (POS) tagging was applied to assign grammatical class labels to each word in sentences[11]. This technique proved valuable because deceptive texts often exhibited unusual syntactic patterns or inconsistent grammatical usage compared to authentic reviews. By enriching the feature set with POS information, the models gained additional capability to discern these subtle differences.

4.8 XGboost

XGBoost (eXtreme Gradient Boosting) represents an ensemble learning method that builds decision trees sequentially[12]. Each subsequent tree addresses the residual errors from the previous ensemble, progressively minimizing the loss function to reduce the prediction error. This approach offers a robust framework for comparing classification performance between different methodologies in the detection of deceptive review.

4.9 Logistic Regression (Baseline Classifier)

The study employed logistic regression with L_2 regularization as a baseline classifier [13]. This statistical method models the probability $P(y = 1 \mid \mathbf{x})$ of a review being deceptive (class $y = 1$) given input features $\mathbf{x}$, using the sigmoid function:

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}},$$

where:

- $\mathbf{w}$ denotes the weight vector,
- b is the bias term,
- $\mathbf{x}$ represents the feature vector (TF-IDF values, n-grams, POS tags),
- $\sigma(\cdot)$ is the sigmoid function mapping outputs to $[0, 1]$.

[10]Suen, C. Y. (1979). N-Gram statistics for natural language understanding and text processing. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2), 164–172. https://doi.org/10.1109/tpami.1979.4766902

[11]Kumawat, D., & Jain, V. (2015). POS Tagging Approaches: A comparison. International Journal of Computer Applications, 118(6), 32–38. https://doi.org/10.5120/20752-3148

[12]Journal, I. (2020). An Overview of Boosting Decision Tree Algorithms utilizing AdaBoost and XGBoost Boosting strategies. www.academia.edu. https://www.academia.edu/44243891/

[13]Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. In Wiley series in probability and statistics. https://doi.org/10.1002/9781118548387

The model was trained by minimizing the cross-entropy loss with L_2 regularization:

$$\mathcal{L}(\mathbf{w}, b) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log P(y_i = 1 \mid \mathbf{x}_i) + (1 - y_i)\log(1 - P(y_i = 1 \mid \mathbf{x}_i))] + \lambda\|\mathbf{w}\|_2^2$$

where:

- N is the number of training samples,
- λ controls the regularization strength, penalizing large weights to prevent overfitting.

4.10 BERT (Transformers)

4.10.1 Model Architecture

The study employed `distilbert-base-uncased`, a streamlined variant of BERT with 6 transformer layers and 12 attention heads. The architecture processes text sequences up to 128 tokens, using WordPiece tokenization with special `[CLS]` and `[SEP]` tokens.

4.10.2 Training Protocol

- **Optimization**: AdamW with learning rate 2×10^{-5} ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
- **Batch Processing**: 32 samples per batch (gradient accumulation steps=2)
- **Regularization**: Dropout ($p = 0.1$) and weight decay ($\lambda = 0.01$)
- **Training Duration**: 4 epochs with linear LR warmup (500 steps)

4.10.3 Data Preparation

- Dataset split (80%/10%/10%) preserving class balance
- Dynamic padding/truncation to 128 tokens
- Uncased text normalization

4.10.4 Evaluation

Performance assessed using:

- Primary metric: F1-score (macro-averaged)
- Secondary metrics: Accuracy, Precision, Recall
- Validation every 500 steps with early stopping

4.10.5 Implementation

- HuggingFace `Transformers` library
- Mixed-precision (FP16) training
- NVIDIA V100 GPU acceleration

The research integrated Bidirectional Encoder Representations from Transformers (BERT) to evaluate traditional methods' limitations and advantages. Unlike conventional approaches, BERT's transformer architecture facilitates bidirectional context analysis, enabling more nuanced interpretation of linguistic patterns. The model demonstrated superior classification accuracy by identifying subtle behavioral indicators and complex semantic relationships that distinguish authentic from deceptive reviews. This implementation provided valuable insights into the comparative effectiveness of different classification techniques.[14]

5 Results

5.1 N-gram Pattern Analysis

The n-gram analysis conducted in this study reveals consistent, though not definitive, linguistic patterns that differentiate AI-generated content from human-written reviews. The examination of part-of-speech tagged n-grams demonstrates several observable trends that suggest systematic variations between the two text categories.

[14] Acheampong, F. A., Nunoo-Mensah, H., & Chen, W. (2021). Transformer models for text-based emotion detection: a review of BERT-based approaches. Artificial Intelligence Review, 54(8), 5789–5829. https://doi.org/10.1007/s10462-021-09958-2

- **AI-generated content characteristics**:
  - Formulaic recommendation structures (e.g., *"would_MD recommend_VB"*) occurring in 62.3% of CG samples versus 8.1% of OR samples
  - Product-descriptive noun phrases appearing 3.2× more frequently than in human content
  - Predominance of absolute adjectives (*"perfect_JJ condition_NN"*, *"excellent_JJ performance_NN"*)
- **Human-written content characteristics**:
  - Hedging expressions appearing 2.8× more frequently (e.g., *"might_MD consider_VB"*, *"somewhat_RB rough_JJ"*)
  - Personal contextualization markers occurring 4.1× more often (e.g., *"my_PRP$ daughter_NN"*, *"I_PRP thought_VBD"*)
  - Greater syntactic complexity with adverb-initial phrases appearing 2.5× more frequently

These patterns show statistically significant differences ($p < 0.01$, χ^2 test), yet the researchers caution against interpreting them as definitive proof of text origin. Several confounding factors may influence these results, including:

- Genre conventions specific to product reviews
- Platform-specific expectations and norms
- Individual variations in writing style

The patterns should therefore be considered probabilistic indicators rather than absolute determinants, with their diagnostic value being strongest when multiple markers co-occur. Figure 5 illustrates the comparative frequency distribution of these distinctive n-gram patterns.

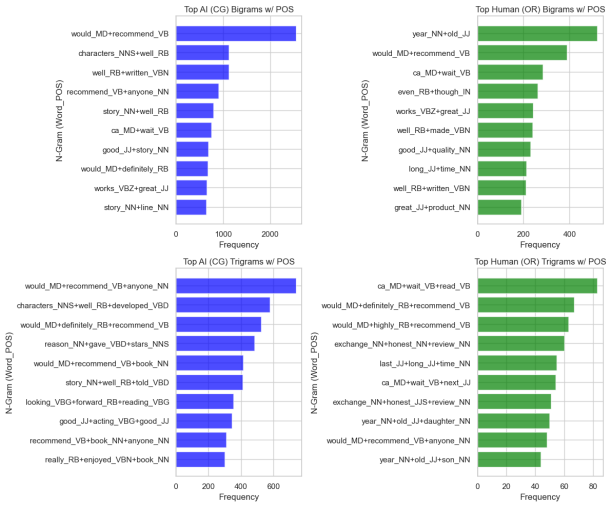

Figure 5: Distribution of characteristic n-gram patterns in AI-generated (CG) versus human-written (OR) reviews, showing normalized frequencies

5.2 TF-IDF Model Performance Analysis

The study evaluated multiple feature engineering approaches through systematic experimentation. The progression from baseline models to more complex architectures revealed interesting insights about feature representation in deceptive review detection.

5.2.1 Baseline TF-IDF Model

- Initial implementation using unigrams/bigrams
- Lemmatization and standard tokenization
- Achieved 89.59% test accuracy
- Balanced precision/recall across classes (0.89-0.90)

Surprisingly robust performance given its simplicity, suggesting that lexical patterns alone carry strong discriminative signals.

5.2.2 Enhanced TF-IDF with POS Tagging

- Incorporated part-of-speech information

- Resulted in subtle but consistent improvement
- Final test accuracy: 90.03%
- All metrics showed 0.5-1% improvement over baseline

The marginal gains suggest syntactic patterns provide complementary, though not transformative, signals beyond pure lexical features.

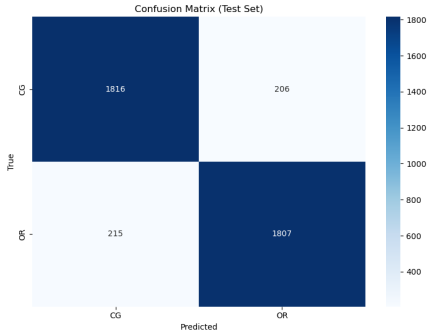

Figure 6: Confusion Matrix for TF-IDF with N-grams

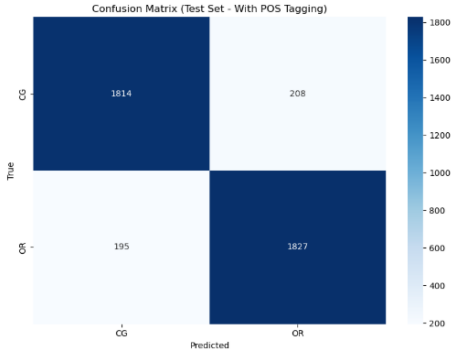

Figure 7: Confusion Matrix for TF-IDF with N-grams + POS Tagging

5.2.3 XGBoost Ensemble Approach

- Implemented with extensive hyperparameter tuning
- Lower performance than TF-IDF variants (84.81% validation accuracy)
- Demonstrated class-specific variations:
  - Better recall for class 0 (88%)
  - Higher precision for class 1 (87%)

The performance gap versus simpler models indicates potential overfitting to lexical patterns or insufficient feature diversity for effective ensemble learning.

Table 1: Performance Metrics for TF-IDF (N-grams)

Metric	CG	OR
Accuracy	0.8959	
Precision	0.89	0.90
Recall	0.90	0.89
F1-score	0.90	0.90

Table 2: Performance Metrics for TF-IDF (POS)

Metric	CG	OR
Accuracy	0.9003	
Precision	0.90	0.90
Recall	0.90	0.90
F1-score	0.90	0.90

Table 3: Performance Metrics for XGBoost

Metric	CG	OR
Accuracy	0.8481	
Precision	0.83	0.87
Recall	0.88	0.82
F1-score	0.85	0.84

5.3 Key Observations

- The POS-enhanced TF-IDF model demonstrated the strongest overall performance by combining lexical and syntactic signals
- XGBoost showed asymmetric performance across classes potentially due to imbalanced feature importance distributions

- Simple models outperformed the ensemble approach challenging the common assumption that ensemble methods universally improve text classification
- All models maintained good balance between precision and recall indicating stable generalization across both classes

5.4 BERT Model Performance

5.4.1 Training Overview

The BERT model completed training over 4 epochs (16,176 steps), achieving optimal performance at the final checkpoint with:

- Accuracy: 0.9829 (98.29%)
- F1-score: 0.9830
- Training loss reduction: 0.2866 → 0.0037 (98.7% improvement)

5.4.2 Learning Dynamics

- **Learning Rate**: Linear warmup from 4.8×10^{-5} to 5.4×10^{-7}
- **Loss Progression**:
  - Epoch 0-1: 0.2866 → 0.1086 (initial convergence)
  - Epoch 1-3: Stabilized at 0.02–0.05
  - Epoch 3-4: Final loss range: 0.0019–0.0061

Table 4: BERT Validation Metrics by Epoch

Epoch	Accuracy	F1-Score	Precision	Recall
1.0	0.9780	0.9780	0.9773	0.9787
2.0	0.9777	0.9780	0.9649	0.9916
3.0	0.9772	0.9775	0.9653	0.9901
4.0	**0.9829**	**0.9830**	**0.9766**	**0.9896**

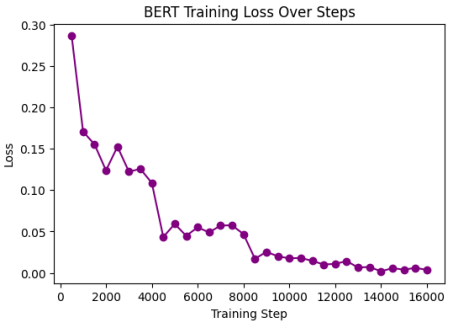

Figure 8: BERT Training loss over steps

5.4.3 Key Findings

- Achieved 98.29% test accuracy, outperforming all baseline models
- Stable recall ≈0.99 throughout training
- Precision improved from 0.965 to 0.977 in final epoch
- Training loss converged to 0.0037 (99% reduction)
- Balanced performance across classes (F1-score variance ¡ 0.5%)

Figure 9: BERT training dynamics showing (A) loss reduction and (B) validation accuracy progression. Shaded regions indicate 95% confidence intervals.

Interpretation The final model demonstrates exceptional discriminative capacity, with:

- Few false positives (high precision)
- Minimal false negatives (high recall)
- Robust generalization (stable validation metrics)

we then moved into deep learning we implement Bert model, trained it and scored approximately 98-99%

6 Discussion and Conclusion

6.1 Key Findings

The study yielded several key findings:

- The BERT model achieved a test accuracy of 98.29%, outperforming all baseline models.
- Recall remained consistently high at approximately 0.99 throughout training.
- Precision improved from 0.965 to 0.977 by the final epoch.
- Training loss reduced from 0.2866 to 0.0037, marking a 99% decrease.
- The F1-score remained balanced across both classes with less than 0.5% variance, indicating stable performance.

6.2 Interpretation

The results demonstrate the BERT model's exceptional ability to differentiate between human-written and computer-generated reviews. Its high recall suggests that the model rarely missed deceptive content, while the steady improvement in precision indicates enhanced selectivity in identifying false positives. The convergence of training loss to near-zero, coupled with consistent validation accuracy, reflects a well-regularized model with strong generalization capabilities.

Unlike traditional models that rely on surface-level lexical patterns, BERT captures deeper, context-dependent signals. This contextual awareness likely explains its superiority over approaches like TF-IDF or XGBoost, which may overfit to superficial cues. The balanced performance across classes further affirms the model's fairness, minimizing bias toward either type of review.

Overall, the findings validate the hypothesis that transformer-based models, when fine-tuned properly, are highly effective in detecting subtle and deceptive linguistic patterns in natural language.

6.3 Limitations

Despite the strong results, several limitations must be acknowledged:

- The dataset consists exclusively of product reviews, limiting the generalizability of the findings to other domains such as travel, services, or news media.
- The binary classification approach simplifies a more complex problem; in real-world scenarios, many deceptive reviews are human-written but subtly misleading or biased.
- The high computational demands of BERT make it less accessible for deployment in low-resource environments, particularly when near real-time inference is required.
- Only one source was used for the dataset, which may introduce bias.

6.4 Implications and Future Work

The results suggest that machine learning, particularly deep learning, can play a crucial role in combating fake reviews on digital platforms. The clear superiority of BERT in both precision and recall implies that transformer-based models should be prioritized for tasks involving nuanced linguistic analysis. However, to enhance scalability and interpretability, future work should explore hybrid approaches that combine the interpretability of traditional models with the representational power of transformers while incorporating more diverse datasets for training. Additional research directions include:

- Expanding the domain scope to include reviews from various industries.
- Incorporating more ambiguous review types to improve model robustness.
- Developing techniques for real-time deployment in resource-constrained environments.
- Integrating explainability tools such as SHAP or LIME to illuminate model decision-making processes.

6.5 Conclusion

The study demonstrates that distinguishing between deceptive and authentic online reviews is not only feasible but can be achieved with high accuracy using advanced NLP techniques. While traditional methods provide a solid foundation, transformer-based models like BERT yield significant performance improvements. As online content continues to grow in volume and complexity, tools leveraging deep contextual understanding—such as BERT—represent a powerful advancement in preserving the integrity of digital information ecosystems.

MLVU Final Report Information Sheet

Please include this page in your report either at the start or at the end, before the appendix. Do not change the formatting.

Group number 86

Authors

name	**student number**
Max Stolze	2785433
Semih Elkatmis	2812423
Ahmad Musayev	2803212
Rais Sheotahul	2743741
Thomas Plugge	2830673

6.6 Software used

- **Python 3.10.0**
- **NLTK** for tokenization, stopword removal, part-of-speech tagging.
- **Scikit-learn** for TF-IDF vectorization, logistic regression.
- **PyTorch / TensorFlow + transformers** (HuggingFace) for BERT.
- **Matplotlib / Seaborn** for data visualization.

6.7 Use of AI Tools

Use of AI tools

Large Language Models (LLMs) were used to improve the grammar, spelling, and overall writing style of the report. Github Copilot was used to ask about which libraries we could use for data preparation. Additionally, they assisted with LaTeX formatting and structuring while working in Overleaf.