Malaysian Journal of Computing, 11 (1): 2437-2446, 2026

eISSN: 2600-8238

TEXTUAL ADVERSARIAL EXAMPLE GENERATION SYSTEM

Noor Adam Noor Azmi[1*] and Haslizatul Fairuz Mohamed Hanum[2]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia
[1*]adam.azmi1519gmail.com and [2]haslizatul@uitm.edu.my

ABSTRACT

The vulnerability of Natural Language Processing (NLP) models to adversarial attacks remains a critical challenge in the field of cybersecurity and AI robustness. While deep learning models have achieved high performance in sentiment analysis, they are susceptible to subtle input perturbations that induce misclassification. This study presents the design and practical implementation of a web-based system (Proof of Concept) that automates the generation of textual adversarial examples using the Bigram Unigram-Semantic Preservation Optimization (BU-SPOF) algorithm. Rather than proposing a novel attack algorithm, our primary contribution is the architectural integration of a dual-source candidate generation strategy (WordNet and OpenHowNet) and a Probability Weighted Word Saliency (PWWS) mechanism to perturb input text while maintaining linguistic coherence. The system was evaluated against a Long Short-Term Memory (LSTM) sentiment classifier using the IMDB dataset.

Keywords: *Adversarial Examples, BU-SPOF, NLP Robustness, Probability Weighted Word Saliency, Sentiment Analysis.*

Received for review: 13-03-2025; Accepted: 27-03-2026; Published: 01-04-2026
DOI: 10.24191/mjoc.vo11i1.11057

1. Introduction

In the swiftly progressing realm of Natural Language Processing (NLP), deep learning models have achieved remarkable performance in complex tasks such as sentiment analysis, machine translation, and text classification, mirroring the success of machine learning predictors in other high-stakes domains (Noviandy et al., 2024; Shafie et al., 2024). However, despite their high accuracy, these models remain highly vulnerable to adversarial examples—crafted inputs designed with the specific intent to deceive machine learning models into making incorrect predictions. State-of-the-art adversarial attacks have demonstrated their effectiveness in triggering errors in model outputs, revealing significant security gaps in real-world AI deployments.

The primary challenge in generating textual adversarial examples lies in the discrete nature of text. In computer vision, adversarial attacks can alter pixel values imperceptibly to the human eye. However, in NLP, text is discrete; changing a single token (word) can drastically alter the grammar, meaning, or readability of a sentence. Therefore, effective textual adversarial examples must adhere to strict linguistic constraints: they must be lexically correct, syntactically sound, and semantically consistent with the original input. If an adversarial example is effective at fooling a

model but is unreadable to a human, it fails as a valid test of the model's robustness in natural language scenarios.

Currently, crafting high-quality adversarial text examples is often a manual or disjointed process, which is inefficient and inconsistent. There is a significant gap in the availability of automated systems that can systematically generate these examples while ensuring semantic preservation. This study addresses this gap by implementing a web-based adversarial generation system driven by the Bigram Unigram-Semantic Preservation Optimization (BU-SPOF) method. By automating the complex calculation of word saliency and semantic filtering, this system provides a practical framework for evaluating the resilience of sentiment analysis models against subtle, semantically consistent manipulations. Rather than proposing a novel attack algorithm or conducting large-scale empirical benchmarking, the primary contribution of this research is the system development and architectural integration of these complex calculations into a functional, web-based framework. This proof-of-concept demonstrates the feasibility of automating semantic-aware adversarial generation.

2. Literature Review

2.1 Textual Adversarial Examples

Adversarial attacks in NLP are categorized based on the granularity of the perturbation: character-level, word-level, and sentence-level.

- **Character-level attacks** involve swapping, deleting, or inserting individual characters. While effective, they often result in spelling errors that are easily detectable by spell-checkers or human readers.
- **Word-level attacks** replace words with synonyms or contextually similar terms, often requiring mechanisms to resolve word sense ambiguity as seen in recent semantic similarity studies (Asraf et al., 2018). This approach is predominant in research because it balances attack effectiveness with the preservation of semantic and syntactic integrity.
- **Sentence-level attacks** modify entire sentence structures or insert new sentences. These modifications can be substantial and risk generating unreadable text or altering the core meaning of the passage.

2.2 The BU-SPOF Algorithm

The BU-SPOF (Bigram Unigram-Semantic Preservation Optimization) algorithm, utilized in this implementation, operates as a black-box attack. Unlike white-box methods that require access to the victim model's gradients, BU-SPOF relies only on the model's output probabilities. Distinct from traditional methods, it employs both unigram (single word) and bigram (two-word phrase) substitutions to enhance the naturalness of the generated text . It integrates a Semantic Filter (SPOF) to ensure the adversarial output retains the original meaning, maximizing the Universal Sentence Encoder (USE) score between the original and perturbed text.

2.3 Comparative Analysis of Attack Methods

To contextualize the selection of BU-SPOF for this implementation, Table 1 compares various state-of-the-art adversarial methods across different perturbation levels. Character-level attacks, such as HOMOCHAR (Bajaj et al., 2023), often achieve high success rates by modifying internal token structures; however, these methods frequently result in discernible spelling errors that fail to maintain lexical correctness, rendering them easily detectable by basic spell-checkers or human

observers. On the other hand, word-level methods like the Heuristic Genetic Algorithm (HGA) and SynonymPSO prioritize readability by substituting entire tokens. While these approaches offer better syntactic coherence than character-level attacks, they often struggle with semantic drift, where the replacement word fits grammatically but alters the original sentiment or meaning of the sentence. BU-SPOF distinguishes itself from these predecessors by employing a dual-candidate strategy that leverages both WordNet and OpenHowNet. This approach allows for a more granular, semantic-aware substitution process, enabling the method to achieve superior success rates while simultaneously preserving a higher degree of semantic similarity compared to single-source substitution methods.

Table 1. Comparison of Textual Attack Method on Sentiment Analysis

Authors	**Method**	**Perturbation Level**	**Victim Models**	**Success Rate**	**Semantics Similarity**
Xiaoyan Yu et al., 2022	Prompt-Attack	Word Level	BERT (IMDB) ALBERT (IMDB) DistilRoBERTa (IMDB)	90.6% 93.5% 90.5%	49% 45% 48%
Ang Li et al., 2023	Seq2seq Stacked Auto-Encoder (SSAE)	Sentence Level	BERT (SST-2), LSTM (SST-2), TextCNN(SST-2)	94.1% 96.1% 95.7%	NA NA NA
Zhixin Shi et al., 2021	SynonymPSO	Word Level	BERT (IMDB) BiLSTM (IMDB)	95% 100%	97% 97%
Shijun Ye et al., 2021	Heuristic-word-selection Genetic Algorithm (HGA)	Word Level	TextCNN (AG'sNews) Tex-tRNN (AG's News) TextBiRNN (AG's News)	89.8% 88.9% 84.0%	89.3% 89.0% 89.3%
Ashish Bajaj et al., 2023	HOMOCHAR	Character Level	BERT (IMDB) ALBERT (IMDB) DistilBERT (IMDB)	95.9% 98.7% 97.9%	NA NA NA
Xinghao Yang et al., 2023	BU-SPOF	Word Level	CNN (IMDB) LSTM (IMDB) Bi-LSTM (IMDB)	100% 100% 100%	99.7% 99.4% 99.5%

As indicated in Table 1, BU-SPOF (Yang et al., 2023) demonstrates high efficacy in both success rate and semantic preservation, making it the optimal choice for implementation in a semantic-aware adversarial generation system. By consistently achieving a semantic similarity score above 99% while maintaining a 100% attack success rate, this algorithm provides the necessary balance between robustness and imperceptibility required for a reliable testing framework.

2.4 Sentiment Analysis and Victim Models

Sentiment analysis aims to decipher the emotional tone of textual data. Deep learning models, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, are widely used for this task due to their ability to capture long-range dependencies in sequential data, a capability recently validated in time-dependent predictive modeling (Basit et al., 2024). However, these models are susceptible to adversarial perturbations, where minor changes in input can drastically shift the probability distribution of the predicted class.

3. System Design and Architecture

This study adopts a quantitative experimental approach to implement and evaluate the BU-SPOF algorithm within a web-based framework. The methodology focuses on the algorithmic pipeline used to perturb input text while maintaining semantic consistency.

3.1 System Architecture and Workflow

The system is designed to automate the flow of adversarial generation. As illustrated in Figure 1, the workflow begins with input selection (either user-generated text or dataset sampling). The input is processed by the adversarial generation module, which interacts with the victim model to iteratively perturb the text until a misclassification occurs.

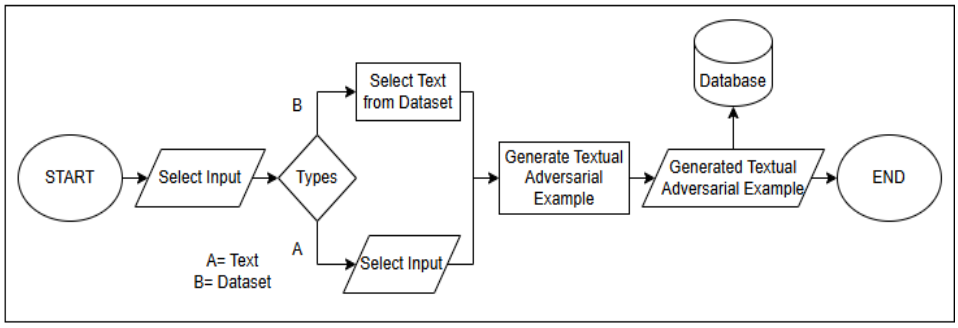

Figure 1. System Flowchart for Adversarial Text Generations

3.2 Algorithmic Implementation

The core engine of the system is built on the BU-SPOF logic, implemented using Python and the Django framework. To facilitate user interaction and automate the perturbation pipeline, the system utilizes a client-server architecture. The frontend provides a web-based interface for users to input text or select dataset samples, which are then transmitted to the Django backend. Within the Django environment, the Python-based BU-SPOF engine processes the input by querying WordNet and OpenHowNet for candidate generation. To manage processing time during these generation stages, the backend handles the semantic evaluations—including the Universal Sentence Encoder (USE) scoring and probability shift calculations against the pre-trained LSTM model—in memory. Once the halt condition is met, the backend returns the generated adversarial examples and their corresponding evaluation metrics (success status, probability shift, and USE score) back to the user interface for display and database storage. The implementation is divided into three algorithmic stages:

3.2.1 Candidate Generation Strategy

To generate valid adversarial candidates, the system employs a dual-source strategy WordNet Integration: The system queries WordNet to retrieve synonym sets (synsets) based on the part-of-speech (POS) tags of the target token.OpenHowNet Integration: To broaden the search space, the system utilizes OpenHowNet to find sememe-based candidates that share semantic structures with the original token.These candidates are combined and filtered to remove duplicates before being ranked for suitability.

3.2.2 Word Saliency and Selection (PWWS)

The system determines which words to attack using a Probability Weighted Word Saliency (PWWS) approach 12. For every word in the input text, the system calculates the "Probability Shift" ($\boldsymbol{\Delta P}$). This metric represents the difference in the model's classification probability for the true label when a specific word is removed or masked. Words that produce a higher $\boldsymbol{\Delta P}$ are deemed "salient" (influential) and are prioritized for perturbation.

3.2.3 Semantic Preservation and Filtering (Halt Condition)

A critical component of the implemented system is the Semantic Filter. As the algorithm iterates through candidates, it calculates the Universal Sentence Encoder (USE) score between the original text and the perturbed text. The system employs a "Halt Condition" function: the perturbation is accepted only if it successfully flips the model's prediction (e.g., from Positive to Negative) and maintains a high USE score. This ensures the attack is successful without altering the core meaning of the text.

4. Proof-of-Concept Evaluation Setup

To validate the functional architecture of the developed web-based system, a proof-of-concept (PoC) evaluation was conducted. This section details the experimental setup used to verify the system's ability to generate valid adversarial texts.

4.1 Evaluation Scope and Sample Size

Because the primary objective of this research is system development and architectural implementation, the evaluation was designed as a functional PoC rather than a large-scale empirical benchmark. A purposive sample of 15 textual examples was selected for this evaluation. This targeted sample size was utilized to perform an in-depth, qualitative analysis of the system's perturbation efficiency and semantic preservation capabilities.

4.2 Dataset and Victim Model

The input texts were sourced from the IMDB sentiment analysis dataset. To evaluate the system's attack capabilities, it was deployed to target a pre-trained Long Short-Term Memory (LSTM) sentiment classifier. The LSTM model was selected due to its established capability in capturing long-range dependencies in sequential text data for sentiment analysis tasks.

4.3 Evaluation Metrics

The system's operational performance on the 15 samples was measured using three primary metrics:

- **Probability Shift (ΔP):** This metric measures the change in the LSTM classifier's prediction probability for the true label after the text is perturbed.
- **Universal Sentence Encoder (USE) Score:** This metric evaluates the semantic similarity between the original text and the generated adversarial text, ensuring the core meaning of the review is retained during the attack.
- **Attack Success Rate:** This is calculated as the percentage of the 15 examples that successfully caused the LSTM model to misclassify the sentiment while simultaneously satisfying the semantic filter's USE score threshold.

5. Results and Discussion

This section presents the findings from the proof-of-concept evaluation, analyzing the system's ability to generate successful adversarial texts while maintaining semantic integrity.

5.1 Attack Effectiveness and Probability Shift

The probability shift shows the influence of the adversarial modifications towards the LSTM classifier's predictions. In our evaluation of the 15 targeted samples, the system successfully fooled the classifier in 14 instances, yielding a 93.33% functional success rate within this proof-of-concept dataset.

For the 14 successful adversarial texts as shown in Figure 2, probability shifts ranged from 0.0387 to 0.3985, indicating that higher shifts are generally linked with success. In contrast, the single failed instance (1 out of 15) demonstrated a low probability shift of 0.0387 combined with a low USE score. This data suggests that probability shifts under roughly 0.1 may decrease the effectiveness of the adversarial texts.

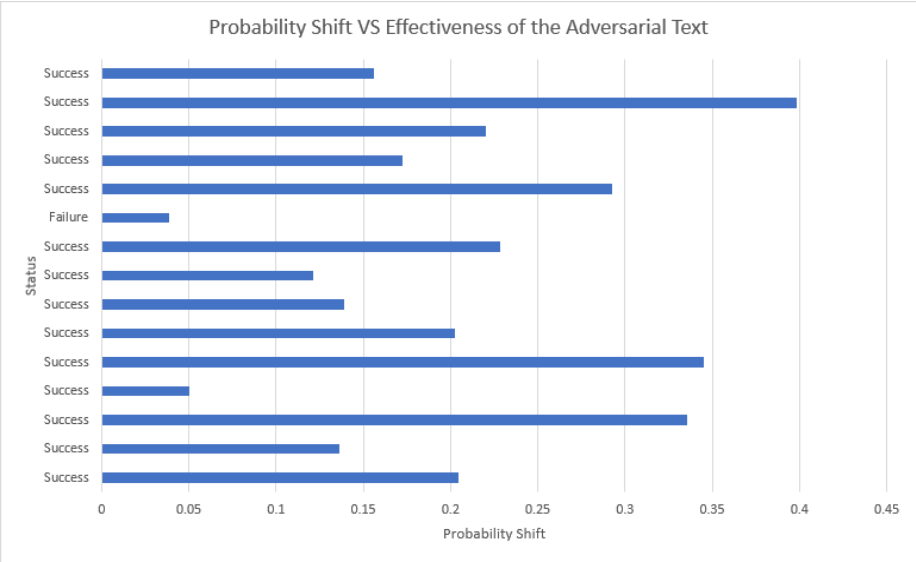

Figure 2. Chart on the Relationship between the Probability Shift and the Status of the Adversarial Texts

5.2 Perturbation Efficiency and Semantic Preservation

The system achieved these successful misclassifications with minimal alterations to the original text. Analyzing the perturbation rates of the 15 evaluated samples reveals the efficiency of the PWWS strategy in targeting salient words:

- 5 examples (33.33%) required changes to only 3 words.
- 4 examples (26.67%) required changes to 4 words.
- 2 examples (13.33%) required changes to only 1 or 2 words.

Furthermore, the Universal Sentence Encoder (USE) scores for the successful examples remained high, averaging above 0.90 (90%). This confirms that the semantic filter module successfully preserved the core meaning of the text during the attack.

5.3 Qualitative Adversarial Examples

Table 2. Qualitative Examples of Generated Adversarial Texts

Id	Original Text	Adversarial Text	Probability Shift	USE Score
1.	Many horror fans complain that horror has scarcely progressed in the last twenty years. I was inclined to agree with this until the **influx** of Asian horror films, a **trend** which has admittedly grown dull. **However**, it has produced some true classics, and A Tale of Two Sisters.	Many horror fans complain that horror has scarcely progressed in the last twenty years. I was inclined to agree with this until the ***inflow*** of Asian horror films, a ***drift*** which has admittedly grown dull. ***Nevertheless***, it has produced some true classics, and A Tale of Two Sisters.	0.3985	99.41%
2.	Three words: Piece of Art. This film is just **great**. It's beautiful, **sad, frightening** and thought-provoking at the same time. The score constantly stays in my head, the acting is wonderful…	Three words: Piece of Art. This film is just ***nifty***. It's beautiful, ***deplorable***, ***fearsome*** and thought-provoking at the same time. The score constantly stays in my head, the acting is ***marvellous***	0.2203	99.45%
3.	I first saw this movie on IFC. Which is a great network by the way to see **underground** films. I watched this movie and was thinking it was going to be **pure** drama…	I first saw this movie on IFC. Which is a great network by the way to see ***clandestine*** films. I watched this movie and was thinking it was going to be ***perfect*** drama	0.3449	99.39%
4.	The dramatic guy that tells Felix he'll "have to sacrifice my art and **go** into the movies." He's in tears.	The dramatic guy that tells Felix he'll "have to sacrifice my art and ***pass*** into the movies." He's in tears.	0.1387	99.96%

As demonstrated in Table 2, the system effectively replaces salient words with contextually appropriate synonyms derived from the dual-source candidate generation strategy. For instance, in Test Case ID 1, the system achieved a massive probability shift of 0.3985 by substituting just three

words (*influx* to *inflow*, *trend* to *drift*, and *However* to *nevertheless*). Similarly, Test Case ID 4 demonstrates that the system can induce misclassification by altering a single token (*go* to *pass*), resulting in a near-perfect semantic similarity score of 99.96%. By targeting only the most mathematically salient tokens, the system successfully flips the LSTM model's sentiment classification without rendering the sentence unreadable or grammatically broken to a human observer.

6. Limitations

While the implemented web-based system successfully automates the generation of semantically preserved adversarial texts, several limitations exist within the current proof-of-concept evaluation.

- **Sample Size and Generalizability:** The functional validation utilized a targeted proof-of-concept sample size of 15 examples. While sufficient for verifying the architectural pipeline and qualitative semantic preservation, this small sample limits the broad statistical generalizability of the attack success rates.
- **Scope of Victim Models:** Initial testing was restricted to a single pre-trained recurrent architecture (LSTM). Expanding backend support to evaluate newer transformer-based architectures, such as BERT or RoBERTa, is necessary to confirm the system's broader applicability.
- **Dataset and Baseline Comparisons:** The current evaluation is limited to the IMDB sentiment analysis dataset. Furthermore, because the primary objective of this study was system development and architectural integration, large-scale empirical benchmarking against baseline methods (e.g., TextFooler, SynonymPSO, or PWWS) was outside the current scope.

Future iterations of this system will focus on addressing these limitations through rigorous, large-scale empirical baseline comparisons across multiple datasets and victim models.

7. Conclusion

This study successfully implemented a web-based adversarial example generation system utilizing the BU-SPOF algorithm. The system automates the complex process of candidate generation, saliency calculation, and semantic filtering, making it accessible for real-time testing. With a **93.33% success rate** on the IMDB dataset, the system demonstrates that effective adversarial examples can be generated with minimal word perturbations while maintaining semantic consistency. The analysis of probability shifts further reveals that successful attacks typically require a probability deviation greater than 0.1.

This tool contributes to the field by providing a practical framework for evaluating and improving the robustness of NLP models against adversarial attacks. While the current implementation focuses on the English language and LSTM models, future work will focus on extending the system to support multiple languages and testing against a broader range of deep learning architectures such as BERT and Transformers to enhance generalizability.

Acknowledgement

The work is part of a final year project carried out at the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor.

Funding

The author(s) received no specific funding for this work.

Author Contribution

Noor Adam Bin Noor Azmi conceptualized the system, implemented the BU-SPOF algorithm, and drafted the manuscript. Haslizatul Fairuz Binti Mohamed Hanum supervised the research, validated the methodology, and reviewed the manuscript.

Conflict of Interest

The authors declare that they have no conflict of interest regarding the publication of this paper.

References

Al-Smadi, M., Hammad, M., Al-Zboon, S. A., Al-Tawalbeh, S., & Wang, Z. (2023). Gated recurrent unit with multilingual universal sentence encoder for Arabic aspect-based sentiment analysis. *Knowledge-Based Systems*, 107540.

Asraf, H., Yahaya, J. H., & Berhan, P. (2018). Word sense disambiguation using fuzzy semantic-based string similarity model. *Malaysian Journal of Computing*, 3(2), 13-26.

Bajaj, A., & Vishwakarma, D. K. (2023). HOMOCHAR: A novel adversarial attack framework for exposing the vulnerability of text-based neural sentiment classifiers. *Engineering Applications of Artificial Intelligence*, 126, 106815.

Basit, A., & Ahmad, N. (2024). Predicting COVID-19 Trends: A Deep Dive Into Time-Dependent SIRSD With Deep-Learning Technique. *Malaysian Journal of Computing*, 9(2), 1955-1978.

Chang, G., Gao, H., Zhou, Y., & Xiong, H. (2023). TextGuise: Adaptive adversarial example attacks on text classification model. *Neurocomputing*.

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. *2018 IEEE Security and Privacy Workshops (SPW)*, 50-56.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Li, A., Zhang, F., Li, S., Chen, T., Su, P., & Wang, H. (2023). Efficiently generating sentence-level textual adversarial examples with Seq2seq Stacked Auto-Encoder. *Expert Systems With Applications*, 213, 119170.

Noviandy, T. R., Idroes, G. M., & Hardi, I. (2024). Enhancing loan approval decision-making: an interpretable machine learning approach using lightgbm for digital economy development. *Malaysian Journal of Computing, 9*(1), 1734-1745. https://doi.org/10.24191/mjoc.v9i1.25691

Shi, Z., Ma, Y., & Yu, X. (2021). An Effective and Efficient Method for Word-Level Textual Adversarial Attack. *2021 IEEE Symposium on Computers and Communications (ISCC)*, 1-6.

Wang, J., Bao, R., Zhang, Z., & Zhao, H. (2022). Rethinking Textual Adversarial Defense for Pre-Trained Language Models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30*, 2526-2540.

Yang, X., Gong, Y., Liu, W., Bailey, J., Tao, D., & Liu, W. (2023). Semantic-Preserving Adversarial Text Attacks. *IEEE Transactions on Sustainable Computing, 8*(4), 583-595.

Ye, S., Zhang, P., Dong, H., & Ji, S. (2021). Heuristic-word-selection Genetic Algorithm for Generating Natural Language Adversarial Examples. *2021 IEEE International Conference on Artificial Intelligence Testing (AITest)*, 39-40.

Yu, X., Yin, Q., Shi, Z., & Ma, Y. (2022). Improving the Semantic Consistency of Textual Adversarial Attacks via Prompt. *2022 International Joint Conference on Neural Networks (IJCNN)*, 1-8.

Zhang, H., Xie, Y., Zhu, Z., Sun, J., Li, C., & Gu, Z. (2021). Attack-words Guided Sentence Generation for Textual Adversarial Attack. *2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC)*, 280-287.