Evaluating the Robustness of LNPL Under Adversarial Text Perturbations: An Empirical and Interpretability Driven Analysis

Thesis Report

Saarthak Solomon Seshadri (29113535)

Data Science (B.Sc.)

Supervisor: Dr. Usman Akhtar

Acknowledgment

I would like to express my deepest gratitude to my family for their unwavering patience, understanding, and support throughout this journey. Their encouragement gave me the strength to stay focused during the most demanding phases of this thesis.

I am also sincerely thankful to my supervisor, whose guidance, insights, and constructive feedback were invaluable at every stage of this research. His direction helped shape this work and kept it aligned with academic rigor.

Abstract

Adversarial robustness has become a critical concern in Natural Language Processing (NLP), particularly as models are deployed in real world settings characterized by noisy, informal, and adversarial inputs. This thesis presents a systematic robustness audit of the Learning with Noisy and Pseudo Labels (LNPL) framework, originally designed to mitigate label noise, by evaluating its performance under adversarial perturbations in text classification.

We reimplement LNPL with a BERT base encoder and train it on a balanced 50,000 example subset of the Yelp Polarity dataset. To evaluate robustness, the model is subjected to five representative adversarial attacks spanning character level, word level, and context preserving perturbations: TextFooler, DeepWordBug, TextBugger, and two BERT Attack variants. These attacks generate semantically equivalent but lexically altered inputs aimed at misleading the classifier.

To diagnose internal failures, we complement performance metrics with SHAP (SHapley Additive exPlanations) visualizations. These token level attributions reveal unstable decision behavior under attack, including overreliance on specific tokens, vulnerability to benign synonyms, and misalignment of contextual salience. Our experiments show that while LNPL performs well on clean data, it experiences substantial degradation under adversarial conditions, with Attack Success Rates exceeding 80% in several cases.

This thesis contributes both empirical evidence and diagnostic insight into LNPL's limitations under adversarial pressure. By categorizing failure modes and analyzing attribution shifts, we propose a framework for improving model robustness through adversarial training, semantic consistency objectives, and interpretability aware learning strategies.

Keywords

adversarial NLP, robustness, LNPL, BERT, sentiment classification, text perturbation, SHAP, TextFooler, DeepWordBug, TextBugger, BERT Attack

Table of Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing
LNPL	Learning with Noisy and Pseudo Labels
PT	Positive Training
NT	Negative Training
BERT	Bidirectional Encoder Representations from Transformers
SHAP	SHapley Additive exPlanations
ASR	Attack Success Rate
FGSM	Fast Gradient Sign Method
MLM	Masked Language Modeling
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
SOTA	State of the Art
KL	Kullback–Leibler (Divergence)
TOC	Table of Contents
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Chapter 1: Introduction

1.1 Background and Motivation

Transformer based models such as BERT have brought about a paradigm shift in natural language processing (NLP), achieving state of the art performance on tasks ranging from sentiment analysis to question answering *(Devlin et al., 2019)*. However, these models are often trained under idealized assumptions most notably, that the training data is clean and representative of the real world distribution.

In practical scenarios, especially with large scale text data, labels are frequently noisy. They may be introduced via weak supervision, heuristic rules, or crowd sourced labelling *(Song et al., 2022; Ratner et al., 2020)*. Recent advances like the LNPL (Learning with Noisy and Pseudo Labels) framework aim to tackle this problem by combining Positive Training (PT) with Negative Training (NT) to improve robustness under label corruption. This thesis builds upon the LNPL framework proposed by Zhu et al. (2022), which was originally designed to address robustness in the presence of label noise and pseudo labelling strategies.

Simultaneously, another threat to NLP model reliability is the vulnerability to adversarial examples inputs that are semantically equivalent to the original but crafted with small perturbations that fool the model into making incorrect predictions *(Jin et al., 2020; Li et al., 2020)*. Despite LNPL's effectiveness against noisy labels, its behaviour under such adversarial conditions has not been examined.

This gap between robust learning with noisy supervision and input level adversarial robustness motivates our investigation. We aim to assess whether the LNPL framework also offers resilience to adversarial perturbations and where it fails to generalize under such attacks.

1.2 Research Objectives

The primary goal of this thesis is to systematically evaluate the adversarial robustness of the LNPL framework and analyse its limitations. To achieve this, we define the following objectives:

- Re implement the LNPL training procedure as defined in the original paper using a BERT based classifier.
- Evaluate the model's resilience against multiple adversarial attacks (TextFooler, DeepWordBug, TextBugger, and BERT Attack).
- Use model explainability tools (SHAP) to interpret prediction shifts and attribute model behaviour.
- Identify and categorize failure modes based on empirical evidence.
- Develop a comprehensive testing framework for robustness evaluation under both clean and perturbed input settings.

1.3 Research Questions

The core question addressed in this work is:

RQ: How robust is LNPL to adversarial text perturbations, and where does it fail internally?

To further refine our investigation, we also ask:

- Does LNPL improve robustness over standard BERT models under adversarial conditions?
- Which categories of perturbation (syntactic, semantic, character level) most frequently lead to failure?
- Can SHAP based interpretation help localize tokens responsible for prediction flips?

1.4 Scope and Limitations

This study focuses exclusively on evaluating the ***LNPL (Learning with Noisy and Pseudo Labels)*** framework in the context of binary sentiment classification using the **Yelp Polarity** dataset. While the original LNPL paper was designed to address robustness against label noise, our emphasis shifts to a different challenge: input level perturbations introduced via adversarial attacks. Specifically, we aim to test whether LNPL's learning paradigm generalizes to scenarios where the input text is subtly manipulated while the label remains unchanged.

Several constraints apply to this investigation:

- ***Dataset Generalizability:*** All training and evaluation are conducted on a single domain specific dataset Yelp Polarity with binary sentiment labels. Results may not directly generalize to multi class or domain adaptive settings.

- ***Attack Coverage:*** We restrict our adversarial evaluation to five representative attack methods: **TextFooler** (Jin et al., 2020), **DeepWordBug** (Gao et al., 2018), **TextBugger** (Li et al., 2019), and two versions of **BERT Attack** (Li et al., 2020). Gradient based white box attacks such as **HotFlip** (Ebrahimi et al., 2018) are excluded due to implementation complexity and runtime costs.

- ***No Adversarial Defences:*** Our model is not retrained with any defence mechanisms such as adversarial training (Miyato et al., 2017) or noise aware fine tuning. The goal is to evaluate **LNPL's default robustness** in isolation.

- ***No Hyperparameter Optimization:*** The LNPL training hyperparameters specifically the **NT margin (0.5)** and **λ (0.04)** were adopted from the original paper without ablation. While stable in practice, these settings may not be optimal for adversarial contexts.

- ***No Calibration Control:*** The model's prediction confidence was not explicitly calibrated using techniques such as temperature scaling (Guo et al., 2017). This may contribute to the sharp confidence drops and high confidence misclassifications observed under attack.

1.5 Research Contributions

This thesis makes the following original contributions:

1. **Reproduction of LNPL with Clean Modular Design**
   We implement the LNPL framework using HuggingFace Transformers and PyTorch, adhering strictly to the paper's structure and loss formulations.
2. **Adversarial Robustness Evaluation**
   This is the first work to evaluate LNPL under multiple adversarial attack scenarios, including black box and semantic preserving perturbations.

3. **Interpretable Failure Analysis using SHAP**
   We use SHAP explanations, we trace attribution changes between original and adversarial inputs to understand internal failure behaviours of the model.
4. **End to End Framework**
   We build a complete experimental pipeline for robust training, adversarial evaluation, visualization, and interpretability, which can be extended to other models.

1.6 Thesis Structure

The rest of the thesis is organized as follows:

- **Chapter 2 Related Work:**
  Reviews literature on adversarial robustness in NLP, noise tolerant training, and interpretability techniques.
- **Chapter 3 Methodology:**
  Describes dataset construction, model setup, training process, and attack configurations used in our experiments.
- **Chapter 4 Implementation and Experimental Results:**
  Presents model performance under clean and adversarial settings, including statistical metrics and robustness gaps.
- **Chapter 5 Discussion:**
  Explores why LNPL fails under certain attacks, analyses patterns in failure, and reflects on implications.
- **Chapter 6 Conclusion and Future Work:**
  Summarizes findings and proposes future research directions like integrating adversarial training or testing on more diverse datasets.

Chapter 2: Related Work

2.1 Overview of Existing Studies

Research in robust NLP has evolved to address multiple axes of model reliability including robustness to label noise, adversarial input perturbations, and generalization under distributional shift. This section explores key literature in the domain of adversarial

robustness in NLP. This section synthesizes literature across three core dimensions: the handling of label noise in text classification, adversarial attacks on NLP systems, and strategies for training models with enhanced robustness.

2.1.1 Robust Learning Under Label Noise

Label noise is a widespread issue in real world NLP datasets, particularly those derived from crowdsourced or weakly supervised sources. Early methods addressing this challenge include noise transition matrices (Sukhbaatar et al., 2015), co teaching (Han et al., 2018), and bootstrapping techniques (Reed et al., 2014). More recent methods have shifted toward robust representation learning.

Liu et al. (2020) proposed a selective augmentation approach that utilizes confidence scores to manage noisy sentiment labels. Similarly, Li et al. (2021) introduced DivideMix, which separates clean and noisy instances using per sample loss analysis, followed by a refined training phase using MixMatch. The LNPL framework by Zhu et al. (2022) combines positive and negative training to penalize overconfidence on incorrect labels, improving noise robustness.

These approaches vary in effectiveness across domains. Table 1 summarizes key methodologies addressing label noise and their respective datasets, techniques, and contributions.

Table 1: Comparison of Adversarial Attack Configurations for LNPL

Author	Method	Dataset(s) Use	Key Technique	Contribution
Han et al. (2018)	Co teaching	CIFAR 10, AG News	Peer training with disagreement	Reduces confirmation bias
Li et al. (2021)	DivideMix	CIFAR 10/100, Yelp	Probabilistic label refinement	Strong separation of clean/noisy
Liu et al. (2020)	Confident Learning	Yelp, Twitter	Confidence based sample filtering	Label correction + selection
Zhu et al. (2022)	LNPL	AG News, DBpedia	PT + NT hybrid training	SOTA results in noise settings

2.1.2 Adversarial Attacks on Text Classifiers

Adversarial robustness in NLP has gained momentum since the introduction of gradient free methods like TextFooler (Jin et al., 2020), which aim to reduce model performance using minimal semantic preserving edits. Follow up work extended this idea in multiple directions.

DeepWordBug (Gao et al., 2018) introduced character level perturbations via insertions, deletions, and swaps, enabling effective black box attacks. Building upon this, TextBugger (Li et al., 2019) combined character and word level attacks for sentiment and toxicity classification. A more recent approach, BERT Attack (Li et al., 2020), employs masked language modelling to replace salient tokens with fluent alternatives, making it highly effective in generating misleading inputs.

These methods differ in their perturbation strategies, fluency retention, and computational cost. Table 2 outlines a comparative view of widely used attack frameworks.

Table 2: Overview of Adversarial Attack Configurations for LNPL Testing

Attack Method	Perturbation Type	Target Model(s)	Key Features	Reference
TextFooler	Word level, greedy	BERT, BiLSTM	Word importance + synonym swap	Jin et al.(2020)
DeepWordBug	Character level	CNN, LSTM	Black box, gradient free	Gao et al.(2018)
TextBugger	Hybrid (char + word)	CNN, BERT	Joint search, black box	Li et al.(2019)
BERT Attack	MLM guided word swap	BERT	Leverages contextual MLM for attack	Li et al.(2020)

2.1.3 Robustness Aware Training and Defences

To counter the susceptibility of NLP models to adversarial perturbations, researchers have developed a variety of defence strategies aimed at enhancing robustness. These methods differ in how they intervene during model training ranging from embedding level noise injection to output regularization and hybrid objective functions. This section presents key robustness aware training strategies, discussing their underlying principles, advantages, and limitations, with a particular emphasis on their relevance to adversarial scenarios.

Adversarial training, introduced by Miyato et al. (2017) and later adapted for NLP specific tasks by Jin et al. (2020), is one of the earliest and most widely adopted defence mechanisms. It introduces small perturbations typically generated using the Fast Gradient Sign Method (FGSM) directly into the input embeddings during training. This encourages the model to learn smoother decision boundaries and enhances its resilience to minor input shifts. However, while effective in continuous embedding space, adversarial training is computationally expensive and performs poorly against discrete input manipulations such as synonym substitutions or character level noise, which are common in real world attacks like TextFooler or DeepWordBug.

An extension of this idea is FreeLB (Zhu et al., 2020), which performs multi step adversarial optimization using accumulated gradient noise in embedding space. By iteratively refining perturbations during training, FreeLB captures stronger adversarial signals than its single step counterparts and has achieved state of the art results on adversarial benchmarks like GLUE. Nevertheless, FreeLB remains limited to continuous perturbations and does not directly address discrete or semantic preserving input alterations.

Another technique, Negative Training, was proposed by Liang et al. (2020) to explicitly penalize overconfidence in incorrect predictions. By incorporating misclassified examples and applying a margin based penalty on their associated confidence scores, Negative Training improves model calibration and robustness under label noise. However, this benefit comes at the cost of reduced accuracy on clean inputs and potential instability if the majority of training samples are misclassified.

LNPL (Zhu et al., 2022) extends this idea further by combining Positive Training (PT), which reinforces correct predictions, with Negative Training (NT), which discourages incorrect overconfident outputs. The LNPL loss function integrates both components into a joint optimization strategy, making it suitable for learning under noisy or pseudo labelled settings. LNPL achieves state of the art performance in weak supervision tasks; however, its robustness has only been tested against synthetic label corruption, not against adversarial perturbations at the input level a critical gap this thesis directly investigates.

The SMART algorithm (Jiang et al., 2020) adopts a different approach by introducing smoothness inducing adversarial regularization in the output space using Kullback Leibler (KL) divergence. By enforcing consistency in model logits under small perturbations,

SMART enhances generalization in tasks like question answering and natural language inference. Its primary drawback lies in its computational cost and limited ability to defend against surface level perturbations, as it operates indirectly on the output distributions rather than the input tokens.

Lastly, MixText (Chen et al., 2020) applies mixup style data augmentation within the hidden layers of transformer models. This technique interpolates between different examples in the latent space, improving generalization and robustness, particularly in semi supervised contexts. However, MixText was not originally evaluated under adversarial benchmarks, and its efficacy against targeted token level attacks remains limited.

The following table summarizes these methods, outlining the core idea behind each defense, its target vulnerability, and the associated trade offs in terms of performance and applicability.

Table 3 : ***Comparison of robustness aware training methods in NLP***

Method	Key Idea	Defence Target	Strengths	Limitations
Adversarial Training	Adds adversarial noise to input embeddings (e.g., FGSM)	Token embedding shift	Improves local smoothness and general robustness	Fails against discrete word level edits
FreeLB	Multi step adversarial optimization using accumulated gradients	Embedding space	Captures stronger adversarial signals than FGSM	Still limited to continuous (not token level) noise
Negative Training	Penalizes overconfident wrong predictions with a margin loss	Label overconfidence	Improves uncertainty calibration	Can hurt accuracy on clean data
LNPL	Joint PT (CE loss) and NT (contrastive	Noisy/pseudo labels	Handles weak supervision and noisy settings well	Not evaluated on adversarial perturbations

	penalty on wrong labels)			
SMART	Enforces output consistency using KL divergence	Output distributions	Encourages smoother output logits under perturbations	High training cost, indirect input robustness
MixText	Applies mixup style data interpolation in hidden layers	Semi supervised representation	Boosts generalization and label efficiency	Weak against targeted attacks on tokens

2.2 Terminology and Definitions

This section provides formal definitions and explanations for all key technical concepts and methodologies used throughout the thesis. It establishes a shared vocabulary and clarifies the structural, functional, and theoretical foundations of this study.

2.2.1 Adversarial Examples in NLP

Adversarial examples in natural language processing refer to synthetically modified text inputs designed to mislead a model's prediction while preserving the original semantic meaning and grammatical correctness (Zhang et al., 2020). These examples exploit vulnerabilities in a model's decision boundaries without producing any perceptible anomaly for human readers. Unlike random noise, which can degrade both human and machine understanding, adversarial perturbations are carefully crafted to deceive models while remaining natural and contextually appropriate.

For instance, consider the original sentence: "The food was great and the service was excellent." A minimally altered version "The food was fine and the service was decent" retains a generally positive sentiment from a human perspective. Yet, such modifications may cause a sentiment classifier to shift its prediction from positive to neutral or even negative, depending on the model's token salience and learned associations. This subtle manipulation, illustrated in Figure 1, highlights how small lexical shifts can result in substantial changes in model output despite the preservation of sentence level intent.

In the context of this thesis, adversarial examples serve a diagnostic role. They are employed to stress test the LNPL (Learning with Noisy and Pseudo Labels) framework and evaluate its robustness to surface level manipulations. By examining model behaviour under adversarial conditions, we can better understand the internal weaknesses that arise not from label noise, but from input level semantic perturbations that the original LNPL formulation was not explicitly designed to handle.

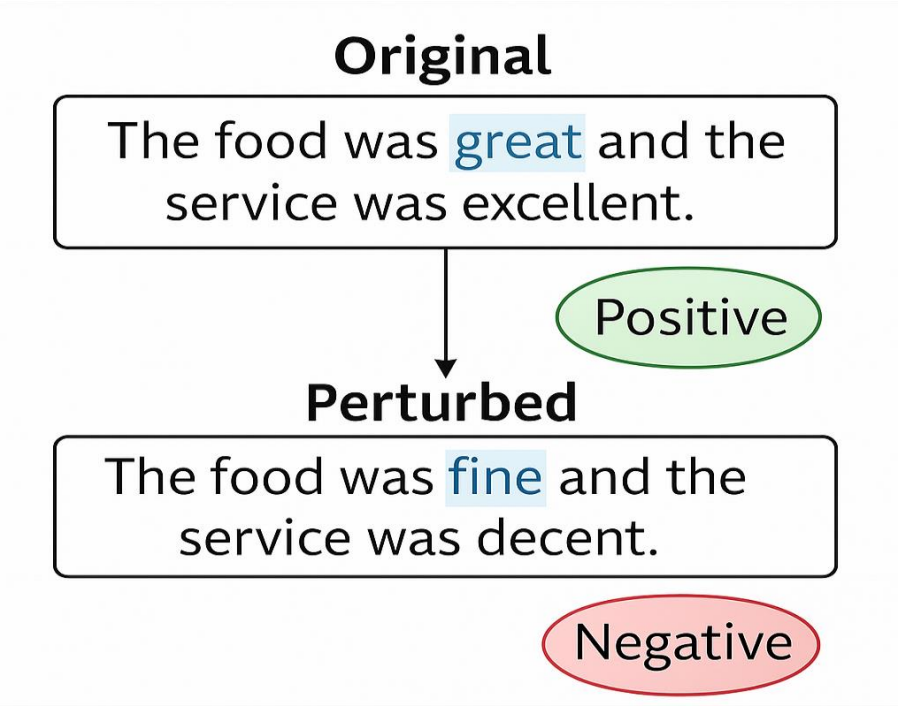


Figure 1: Example of semantic preserving adversarial attack flipping sentiment classification

2.2.2 Label Noise and Weak Supervision

Label noise refers to inaccuracies or inconsistencies in the labels assigned to training data, which can significantly impair model performance and generalization. As noted by Song et al. (2022), such noise often appears from various practical sources, including human annotation errors, ambiguous labelling criteria, and the use of automated or crowdsourced labelling processes. For instance, large scale datasets built via platforms like Amazon Mechanical Turk may suffer from variability in annotator quality, while labels derived from heuristic rules such as keyword based sentiment tagging can introduce systematic bias. Other sources include optical character recognition (OCR) errors in

digitized text and domain mismatch, where labels from one context are incorrectly applied to another.

The concept of weak supervision generalizes this challenge further. It encompasses a range of imperfect labelling strategies, including distant supervision, noisy supervision, and semi supervised learning (Ratner et al., 2020). Unlike traditional supervised learning that assumes high quality labelled data, weak supervision accepts that training data may be partially labelled, pseudo labelled, or inconsistently annotated. This paradigm allows for scalability but introduces a new set of complexities in learning robust representations from noisy or incomplete information.

The LNPL (Learning with Noisy and Pseudo Labels) framework was developed to precisely address these challenges. By combining Positive Training (PT) on known or confident labels with Negative Training (NT) that penalizes overconfident misclassifications, LNPL offers a principled way to handle both noisy annotations and weakly supervised inputs. This dual loss strategy allows the model to learn from imperfect data without overfitting to errors, making it particularly suitable for real world NLP scenarios where clean labels are scarce or unreliable.

2.2.3 Positive and Negative Training (LNPL)

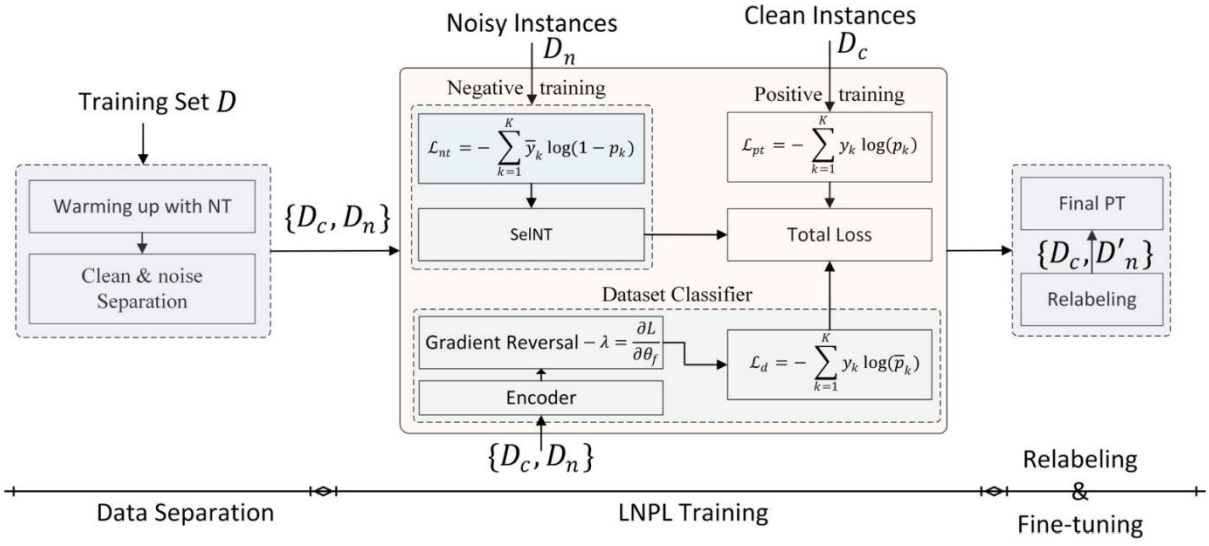
The LNPL (Learning with Noisy and Pseudo Labels) framework stands for a hybrid training paradigm specifically designed to handle uncertainty in training data, particularly in settings with noisy or weak supervision. At its core, LNPL combines two complementary learning objectives: Positive Training (PT) and Negative Training (NT). The PT part corresponds to standard cross entropy loss applied to either ground truth or high confidence pseudo labelled examples, reinforcing correct predictions and promoting model alignment with clean labels. In contrast, the NT part introduces a contrastive penalty that suppresses overconfidence on incorrect labels. Rather than requiring knowledge of the true negative class, NT works by penalizing instances where the model makes confidently wrong predictions, encouraging more calibrated decision boundaries.

The combined loss function in LNPL is formulated as a weighted sum:

$$L_{total} = L_{PT} + \lambda \cdot L_{NT}$$

where λ is a tuneable hyperparameter that decides the relative influence of the NT term during optimization. This dual objective formulation allows the model to balance learning from reliable signals while avoiding overfitting to noisy or ambiguous examples. Importantly, while LNPL was originally proposed to combat label noise, it does not include mechanisms for handling adversarial perturbations at the input level a limitation that forms a central hypothesis of this thesis.

As illustrated in Figure 2, the LNPL architecture computes PT and NT losses in parallel and aggregates them into a unified loss objective, making it compatible with modern transformer based models. A key innovation of NT is its label agnostic nature: it does not require explicit negative class annotations, making it particularly useful when training data includes pseudo labels or unverified annotations. This capability also makes NT an intriguing candidate for exploring model behaviour under adversarial drift, a context in which input integrity rather than label correctness is compromised.



Note: Adopted from Zhu, H., et al. (2022). Towards Robust Learning with Noisy and Pseudo Labels for Text Classification.

Figure 2: LNPL Architecture shows parallel PT and NT loss computation and final aggregation into a joint optimization objective

2.2.4 Semantic Perturbations and Robustness

Semantic perturbations are subtle input modifications that alter token level content while preserving the overall meaning and grammatical structure of a sentence. These changes

are particularly challenging for models that show excessive sensitivity to specific tokens, rigid syntactic structures, or fine grained embedding representations. When such models rely heavily on surface level features, even minor variations despite being semantically equivalent can trigger misclassifications or unstable predictions.

Common categories of semantic perturbations include character level alterations (e.g., “excellent” changed to “excelllent”), word level synonym substitutions (e.g., “love” replaced with “like”), and syntactic rephrasing such as changing active to passive voice or reordering sentence components. Although these perturbations keep the intent and fluency of the original input from a human perspective, they often exploit brittle model dependencies, leading to degraded performance or loss of interpretability.

Figure 3 provides a visualization of how such perturbations affect a model’s internal attribution map, as revealed through SHAP analysis. Colour coded token saliency shifts show how small semantic preserving edits can distort the model’s focus away from relevant features, thereby weakening prediction reliability.

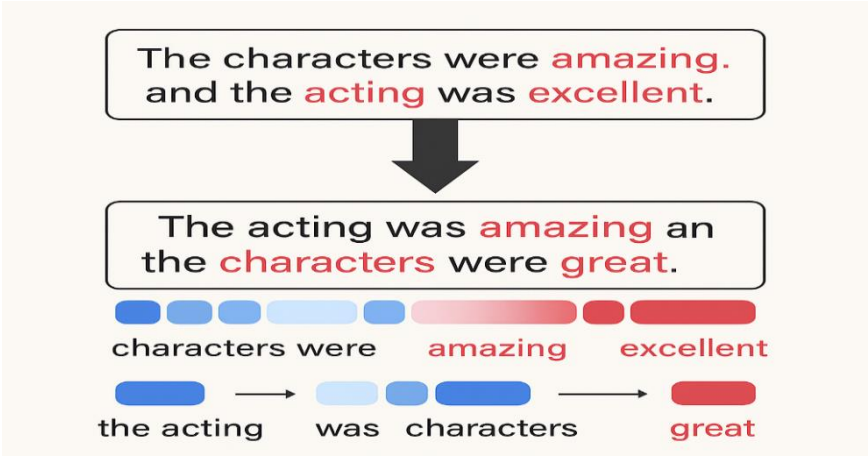


Figure 3: Visualization of how a semantic preserving perturbation breaks the model’s feature attribution (colour coded by SHAP)

In this thesis, robustness is defined as a model’s ability to keep consistent outputs and stable feature attributions even in the presence of adversarial noise that does not compromise the human perceived intent of the input. This notion of robustness is critical when evaluating models like LNPL, whose original formulation did not account for input level variation yet must now be assessed under adversarial conditions where semantics stay intact, but token level cues are manipulated.

2.2.5 SHAP: Model Interpretability for Text Classification

SHAP (SHapley Additive exPlanations) is a widely adopted interpretability framework that attributes prediction outcomes to individual input features based on Shapley values from cooperative game theory (Lundberg & Lee, 2017). The fundamental premise of SHAP is to quantify the contribution of each feature in the context of NLP, each token or sub word to the final output of a model. Given a trained model and a specific input sample, SHAP generates a set of additive importance scores that show how much each token pushes the prediction toward or away from a particular class.

In text classification tasks, SHAP enables a fine grained, token level explanation of model behaviour. By comparing SHAP attributions for original and perturbed versions of the same input, researchers can visualize how semantic preserving adversarial changes distort the model's internal reasoning. This makes SHAP especially valuable for diagnosing brittleness and attribution drift in NLP models subjected to adversarial attacks.

Figure 4 illustrates this process by showing a positive review and its adversarial counterpart, along with the corresponding SHAP token attribution maps. The visual differences highlight how even slight lexical edits can drastically shift the model's attention, revealing the fragility of token salience under seemingly innocuous perturbations.

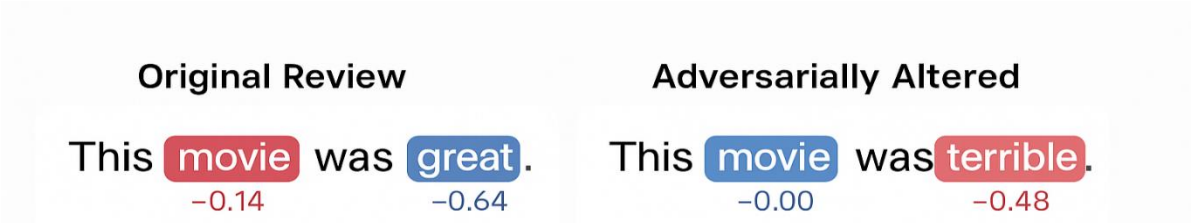


Figure 4: Example SHAP visualization showing token attribution in a positive review before and after adversarial replacement

In this thesis, SHAP serves as a primary interpretability tool to trace internal failure modes of the LNPL framework, helping to explain not only when the model fails but why its attribution patterns collapse under adversarial stress.

2.2.6 Evaluation Metrics

Evaluating model performance in adversarial NLP tasks requires careful selection of metrics that go beyond surface level accuracy. The most basic metric, accuracy, measures the proportion of correct predictions out of all predictions made. Formally, it is defined as:

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

where TP, TN, FP, and FN refer to true positives, true negatives, false positives, and false negatives respectively. While this may be useful in balanced settings, accuracy alone may not be able to reflect the true impact of adversarial noise, especially when class distributions are skewed or when perturbations cause subtle shifts in model confidence.

To offer a more nuanced view, precision and recall are often used in tandem. Precision quantifies the proportion of correct positive predictions among all predicted positives, calculated as

$$precision = \frac{TP}{(TP + FP)}$$

Recall, on the other hand, measures the proportion of actual positive instances that were correctly found, given by

$$Recall = \frac{TP}{(TP + FN)}$$

Each captures distinct aspects of classification reliability, with precision emphasizing correctness and recall emphasizing completeness.

The F1 Score offers a balanced metric by taking the harmonic mean of precision and recall:

$$F1\ Score\ = \frac{2.(precision.recall)}{(precision + recall)}$$

This metric is particularly valuable in adversarial settings, where models may keep high accuracy by exploiting shortcuts or biases, even as their decision boundaries degrade. Unlike accuracy, which can remain deceptively stable, the F1 score is more sensitive to

small shifts in classification consistency and thus provides a more robust signal of performance under perturbation.

Figure 5 illustrates the confusion matrix, which forms the foundation for computing all these metrics by showing the distribution of correct and incorrect predictions across both positive and negative classes.

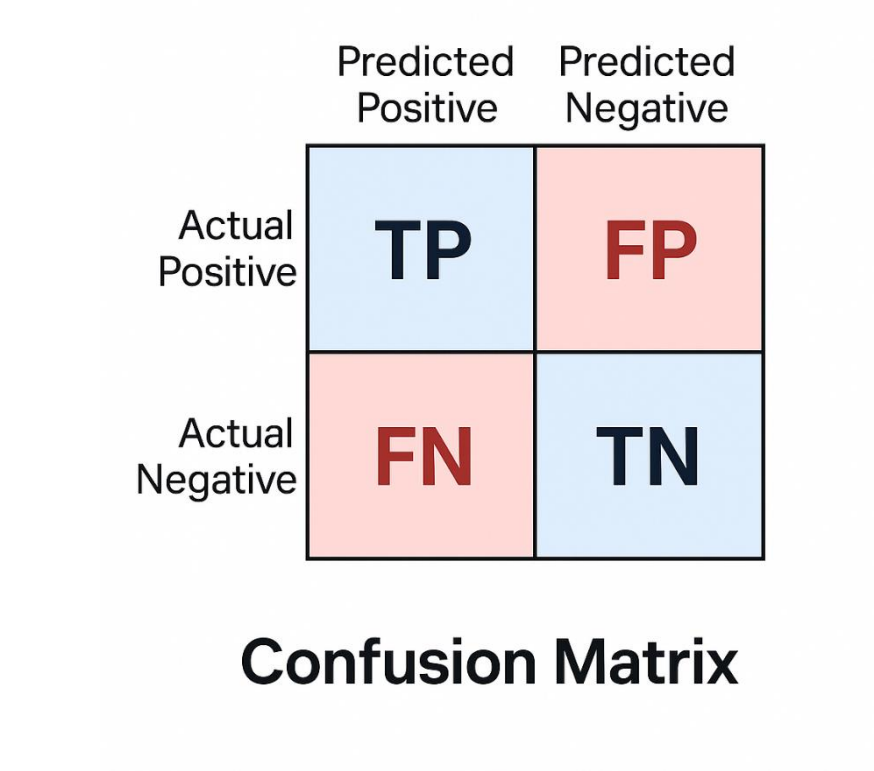


Figure 5: Confusion Matrix diagram illustrating TP, FP, FN, TN regions

In the context of this thesis, the F1 score is used as a primary robustness indicator across clean and adversarial test sets, capturing both the correctness and completeness of predictions in challenging settings where surface changes do not alter the semantic intent of the input.

2.2.7 Transformer Models and BERT

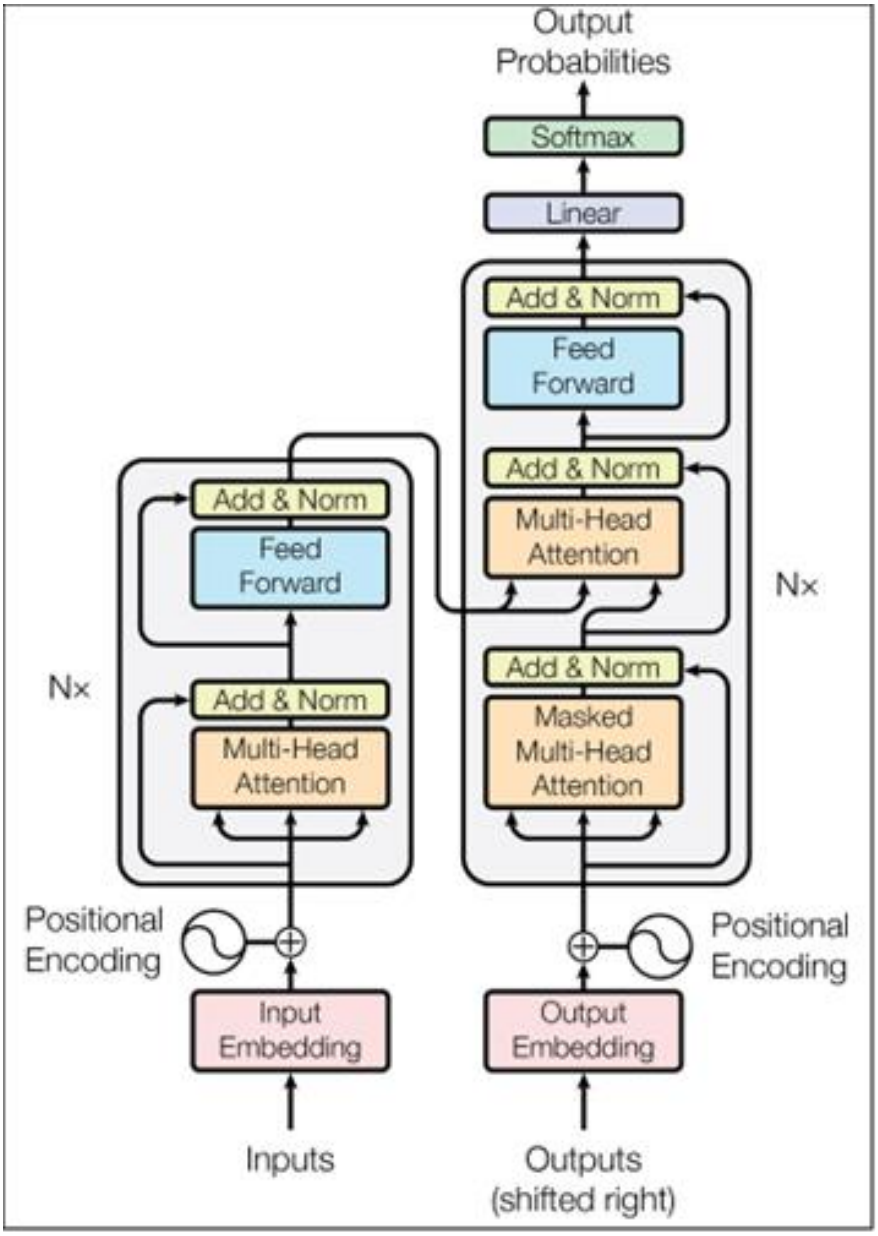
Transformer architectures have fundamentally transformed the landscape of natural language processing by replacing recurrent neural networks with a parallelizable attention based mechanism called self attention (Vaswani et al., 2017). Instead of processing words one after the other, transformers look at the entire input sequence all in one go. This not only makes the model's understanding of the context better but also makes

training a lot more efficient. As a result, transformer-based models can scale to much larger datasets and effectively handle longer texts.

Among these, BERT was introduced by Devlin et al. (2019), which represents a major milestone. BERT learns language representations by randomly masking words in a sentence and predicting them based on the surrounding context from both directions. This bidirectional training approach allows BERT to develop rich, contextualized embeddings, which have dramatically improved performance across a numerous number of NLP tasks.

In this thesis, the LNPL framework is implemented on top of the BERT base architecture, specifically using the bert base uncased variant. This model ignores case distinctions treating “Good” and “good” as equivalent which is generally suitable for sentiment classification tasks where capitalization holds minimal semantic weight. This choice also aligns with the original LNPL study, which used uncased embeddings for general domain classification benchmarks such as AG News and DBpedia. Our experiments follow this configuration by fine tuning the BERT base model on the Yelp Polarity dataset, ensuring consistency with prior work and a solid foundation for evaluating robustness under adversarial perturbation.

Figure 6 presents a schematic overview of the transformer encoder block, illustrating the core components of multi head self attention and feedforward layers that underpin BERT’s representational power.



Note: Adopted from Vaswani et al. (2017), p. 3

Figure 6: Transformer encoder block showing multi head self attention and feedforward layers.

Chapter 3: Methodology

In this chapter, we outline the methodological framework used to evaluate the robustness of the LNPL (Learning with Noisy and Pseudo Labels) model under adversarial conditions. The study is structured around a binary sentiment classification task using the Yelp Polarity dataset, a large scale collection of user generated reviews labeled as either

positive or negative. We fine tune a BERT based classifier with the LNPL training regime and assess its behavior under a range of adversarial text perturbations, including character level edits, synonym replacements, and grammar preserving substitutions. The chapter begins by detailing the dataset selection and preprocessing steps, followed by the integration of the LNPL loss components into the model architecture. We then describe the training configuration and the generation of adversarial examples using established attack methods such as TextFooler, DeepWordBug, TextBugger, and BERT Attack. To evaluate robustness, we employ both quantitative metrics accuracy, F1 score, attack success rate and qualitative interpretability techniques, including SHAP based attribution analysis. The chapter concludes with an explanation of the system setup and tools used, laying the groundwork for the empirical evaluations and failure analysis presented in Chapters 4 and 5.

3.1 Research Approach

This study adopts a hybrid methodological framework that combines model centric experimental design with explainability driven failure analysis. The core objective is to evaluate the robustness limitations of the LNPL (Learning with Noisy and Pseudo Labels) framework when exposed to adversarial text perturbations in a binary sentiment classification setting.

LNPL was originally developed to address the challenge of label noise in weakly supervised learning by combining Positive Training (PT) and Negative Training (NT) losses. However, its behavior under input level adversarial noise where perturbations preserve the original meaning but are crafted to mislead models remains untested. This research aims to bridge that gap by proposing a systematic methodology for stress testing LNPL and interpreting its decision making dynamics.

Workflow Diagram

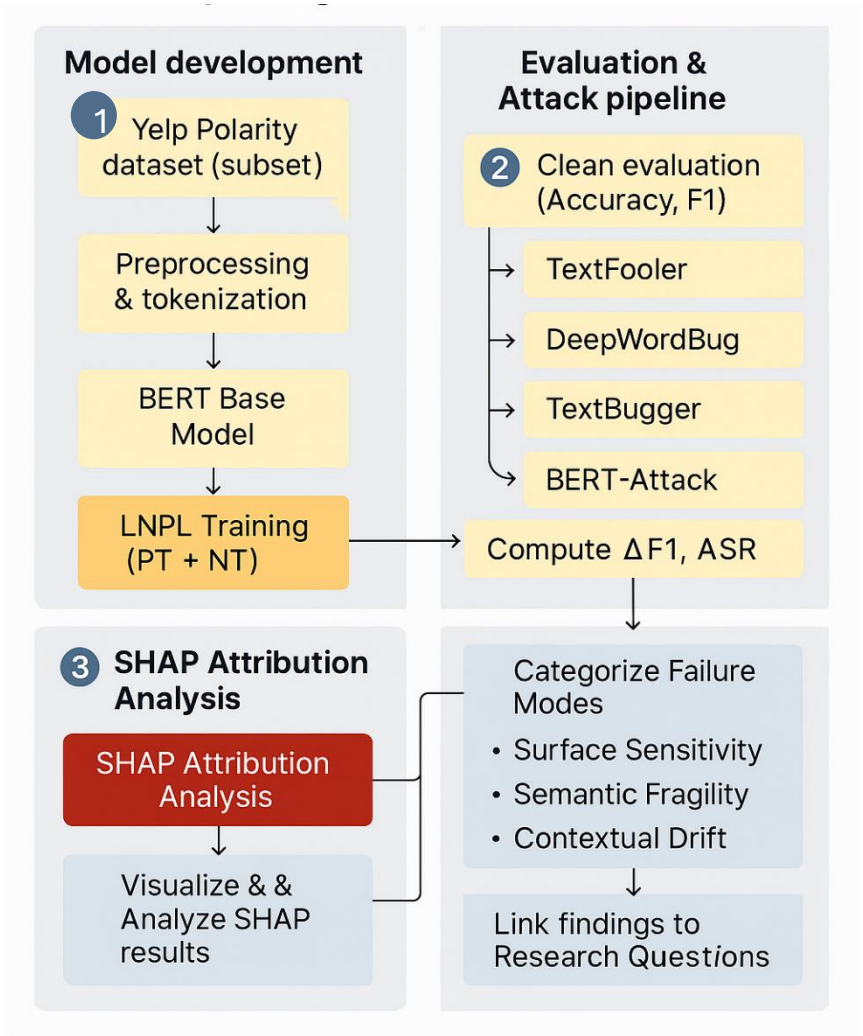


Figure 7: Research Workflow for Evaluating LNPL Robustness Under Adversarial Perturbations

To carry out this, the methodological design is divided into three sequential phases:

3.1.1 Model Development Phase

The study begins with the re implementation of LNPL using a BERT base uncased encoder as the backbone architecture. The training objective is customized to include both PT (cross entropy loss on ground truth labels) and NT (margin based penalty on incorrect labels), allowing the model to optimize not only for correct predictions but also for conservative decision boundaries that avoid overconfidence in wrong classes.

To balance hardware efficiency and statistical validity, we curate a 50,000 example balanced subset from the Yelp Polarity dataset (25,000 positive and 25,000 negative). This size ensures fast convergence on limited hardware while maintaining a sufficiently diverse input distribution for adversarial robustness evaluation.

3.1.2 Evaluation and Attack Pipeline

Following training, the LNPL enhanced model is first evaluated on unaltered (clean) test data to establish a performance baseline in terms of accuracy, precision, recall, and F1 score. This clean evaluation serves as a reference point for measuring performance degradation when the model is exposed to adversarial inputs.

To test the model's resilience, we then apply five distinct adversarial attack methods from the TextAttack framework, each selected to probe a different axis of vulnerability in the input space. These attacks are:

- **TextFooler** targets important words and replaces them with semantically similar synonyms
- **DeepWordBug** introduces noise at the character level via insertion, deletion, and replacement
- **TextBugger** applies a hybrid strategy combining both word and character level noise
- **BERT Attack (Masked LM) substitutes** high impact tokens using predictions from a masked language model
- **BERT Attack (Embedding) performs** synonym swaps guided by word embedding similarity

The purpose of using this diverse suite of attacks is to comprehensively evaluate how well the LNPL model generalizes across different types of adversarial perturbations ranging from surface level spelling changes to deeper semantic rewordings. By including both discrete and contextual attacks, we aim to test whether LNPL's regularization strategy can offer robustness not just to label noise, but also to structurally different input manipulations.

Each attack generates semantically equivalent, yet misleading inputs designed to flip the model's predictions. By comparing the clean performance to adversarial outcomes, we compute robustness metrics such as **ΔF1** (the drop in F1 score) and **Attack Success Rate (ASR)**. These metrics quantify both the effectiveness of each attack and the fragility of the

model, helping us identify which types of perturbations most significantly degrade performance.

3.1.3 SHAP Attribution Analysis

To uncover why the model fails, we conduct an interpretability phase using SHAP (SHapley Additive exPlanations), which reveals the contribution of each input token to the model's final prediction. For each attack type, representative failure cases are selected where adversarial input successfully flips the model's output.

We then compare attribution heatmaps between clean and perturbed versions of the same sample. This reveals whether the model's internal decision making shifted, diffused, or collapsed under input noise.

This explainability driven approach transforms raw metrics into interpretable, actionable insights about LNPL's robustness limitations.

3.2 LNPL Based Model Design (Model Development Phase)

This section corresponds to the **Model Development Phase** of the overall research methodology illustrated in Figure 3.1. It describes how we implemented the LNPL (Learning with Noisy and Pseudo Labels) framework to build a robust sentiment classification model using BERT as the base encoder. Instead of modifying the underlying model architecture, LNPL introduces robustness through a hybrid loss function that combines traditional supervised learning with a contrastive regularization mechanism. This design enables the model to balance confident predictions on correct labels (via Positive Training) with uncertainty calibration on incorrect predictions (via Negative Training). The following subsections detail the rationale, model selection, loss formulation, and hypothesized benefits of this setup in adversarial settings.

3.2.1 Justification for LNPL in Adversarial Robustness Testing

Although originally proposed to handle label corruption and pseudo label noise in weakly supervised classification tasks, the LNPL framework offers characteristics that are highly relevant to adversarial robustness evaluation. Its core learning principle reinforcing

correct predictions while penalizing overconfidence in incorrect ones aligns with the needs of models operating in adversarial contexts.

In particular, adversarial attacks often aim to trigger confident misclassifications by subtly modifying the input. LNPL's Negative Training (NT) component directly penalizes such confident errors, encouraging the model to learn more conservative decision boundaries. This thesis investigates whether LNPL's regularization behaviour, though designed for noisy labels, can also mitigate vulnerability to input level perturbations. As such, LNPL is adopted as the focal point of our experimental framework to test this cross domain robustness hypothesis.

3.2.2 Base Model Configuration and Integration

This component of the Model Development Phase corresponds to the base model selection outlined in the research workflow (Figure 3.1). The architecture used in this study is BERT base uncased, a 12-layer bidirectional transformer pretrained on large English corpora. This model was selected for three primary reasons: its proven efficacy in sentiment classification tasks (Devlin et al., 2019), its direct compatibility with the original LNPL formulation (Zhu et al., 2022), and its seamless integration with the Hugging Face Transformers library, which enabled efficient training and experimentation.

Importantly, the BERT architecture itself remains unmodified throughout the study. All robustness adaptations were applied at the **training loss level**, via the integration of LNPL's joint Positive and Negative Training objectives. This design choice ensures that any observed differences in robustness performance are attributable solely to the loss function, rather than architectural changes, maintaining fair comparability with baseline models trained using standard cross entropy.

3.2.3 LNPL Loss Formulation

LNPL defines a **joint loss function** composed of:

- **Positive Training (PT) Loss:** A standard cross entropy objective encouraging accurate predictions for ground truth labels
- **Negative Training (NT) Loss:** A contrastive, margin based penalty discouraging the model from being overconfident in incorrect labels

The combined loss is:

$$L_{total} = L_{PT} + \lambda \cdot L_{NT}$$

Where:

- λ*=0.04* is a tunable scaling factor
- The **margin** for NT is set to 0.5
- Negative labels are sampled randomly from all labels excluding the ground truth

This formulation is designed to sharpen decision boundaries while discouraging overconfident predictions for incorrect alternatives. In binary classification (such as Yelp Polarity), this contrastive regularization becomes especially impactful.

3.2.4 Hypothesized Benefits in Adversarial Settings

Although LNPL was not explicitly designed for adversarial defense, we hypothesize that its training strategy may confer several indirect benefits:

- **Suppressing high confidence mistakes** caused by word level or character level perturbations
- **Reducing over reliance on spurious correlations** such as highly weighted sentiment keywords
- **Encouraging flatter decision boundaries**, which are known to correlate with improved robustness in prior literature

These hypotheses are tested empirically in Chapter 4 across five adversarial attack methods.

3.2.5 Workflow Placement

This model development phase forms the **foundation of our overall research methodology**, as depicted in Figure 3.1. The LNPL enhanced model trained here is subsequently subjected to clean evaluation, adversarial stress testing, and interpretability driven failure diagnosis through SHAP. Together, these phases support a comprehensive audit of LNPL's adversarial resilience.

3.3 Dataset Tokenization and Splitting

To prepare the Yelp Polarity dataset for training and evaluation, we followed a two step preprocessing strategy grounded in best practices for transformer based models: (i) sub word tokenization, and (ii) controlled sequence formatting.

3.3.1 Tokenization with BERT Tokenizer

We employed the bert base uncased tokenizer from the Hugging Face Transformers library, which utilizes a WordPiece vocabulary to effectively handle rare or misspelled words by segmenting them into interpretable subword units. This property is particularly important when dealing with user generated content such as Yelp reviews, which frequently include informal expressions, spelling variations, and colloquial slang. The tokenizer converts raw text into input token IDs and generates attention masks to differentiate between actual content and padding. The decision to use the uncased variant was intentional, as capitalization is generally non informative in noisy review text and its exclusion helps reduce input variability without sacrificing semantic integrity.

Truncation and Padding

Each review was padded or truncated to a fixed maximum length of **128 tokens**. This value was selected based on the distribution of review lengths in the Yelp dataset, which skew short, allowing 128 tokens to capture nearly all useful information without excessive truncation. We used 128 tokens as it balanced information coverage with GPU memory efficiency. Larger max lengths (e.g., 256 or 512) provide minimal additional context for Yelp reviews, but at significantly higher computational cost.

Preprocessing Objective

This step ensures:

- Uniform input size across the dataset.
- Compatibility with BERT's input format.
- Efficient GPU batching downstream.

3.3.2 Splitting Strategy

- **Training Split**: Created using a balanced subset of the original Yelp Polarity training set. Although this sampling process is part of implementation, our methodology assumes a **stratified split** ensuring equal representation of positive and negative reviews during training.
- **Test Split**: The standard Yelp Polarity test set (≈38,000 examples) was retained in its original form to serve as an unperturbed evaluation benchmark.

We did not include a validation set for most experiments. However, when tuning hyperparameters such as the λ coefficient in LNPL loss, we temporarily created an **80/20 stratified validation split** from the training data to ensure reliable metric feedback.

3.4 Adversarial Testing Configuration

To rigorously assess the robustness of the LNPL enhanced BERT model, we implemented a comprehensive adversarial testing pipeline using the TextAttack library. This evaluation stage simulates realistic yet challenging perturbations that preserve sentence level semantics while attempting to flip the model's predictions.

3.4.1 Testing Framework: TextAttack

TextAttack (Morris et al., 2020) is a Python based framework specifically designed for adversarial attacks on NLP models. It offers a wide variety of attack recipes, constraint sets, and semantic similarity controls. We used version 0.3.4 to ensure compatibility with Hugging Face Transformers v4.37.0 and to support custom constraint injections for BERT Attack variants.

TextAttack was integrated into our pipeline via the `HuggingFaceModelWrapper` and `AttackArgs` utilities, enabling streamlined batch attacks, logging, and reproducibility.

3.4.2 Dataset Configuration for Attacks

For adversarial testing, we selected a subset of 50-100 samples from the original Yelp Polarity test set for each attack. This smaller sample size was chosen to ensure runtime feasibility across attacks while maintaining statistical reliability.

All test instances were wrapped into a `textattack.datasets.Dataset` object, which provides compatibility with TextAttack's internal logic. Ground truth labels and predictions were logged alongside each perturbed example for evaluation and interpretability.

3.4.3 Summary of Attack Methods Used

The table below summarizes all five attacks used in our evaluation, spanning character level, word level, and contextual perturbations:

Table 4: Overview of Adversarial Attack Configurations for LNPL Testing

Attack	Perturbation Type	Customization	Notes
TextFooler (Jin et al., 2020)	Word level synonym swap	POS constraints, fixed seed	100 examples, textfooler_lnpl_results.csv
DeepWordBug (Gao et al., 2018)	Character level edits	Default config	100 examples, deepwordbug_lnpl_results.csv
TextBugger (Li et al., 2019)	Word + char hybrid	Default config	100 examples, textbugger_lnpl_results.csv
BERT Attack (Masked LM)	MLM based contextual substitutions	Custom candidate filtering, 30% modification cap	50 examples, bertattack_lnpl_no_use.csv
BERT Attack (Embedding)	Synonym replacement via embeddings	Min cosine sim = 0.8, Max mod = 30%	100 examples, bertattack_embedding_results.csv

These attacks collectively represent diverse perturbation strategies designed to probe different dimensions of the model's robustness.

3.4.4 Execution Logic

Each attack recipe was instantiated using `HuggingFaceModelWrapper`, with `AttackArgs` specifying reproducibility parameters such as `random_seed = 42`. For complex attacks like BERT Attack, we manually defined custom components:

- **WordEmbeddingDistance** to enforce semantic similarity
- **GreedyWordSwapWIR** for targeted substitutions
- **MaxModificationRate** to cap token changes at 30%

This careful parameter tuning ensured attacks were both high fidelity and challenging, preserving fluency while altering model relevant content.

3.4.5 File Outputs and Analysis Workflow

Each adversarial attack outputs its results into a structured .csv file containing the original input, the perturbed input, model predictions before and after perturbation, and whether the attack was successful. These result files serve as the basis for the analyses presented in Chapter 5, where they are used to compare accuracy and F1 scores across different attack types, generate SHAP based interpretability visualizations, and support manual categorization of failure cases. Representative output files include `textfooler_lnpl_results.csv` and `bertattack_embedding_results.csv`, which correspond to specific attacks and are referenced throughout the robustness and attribution analysis.

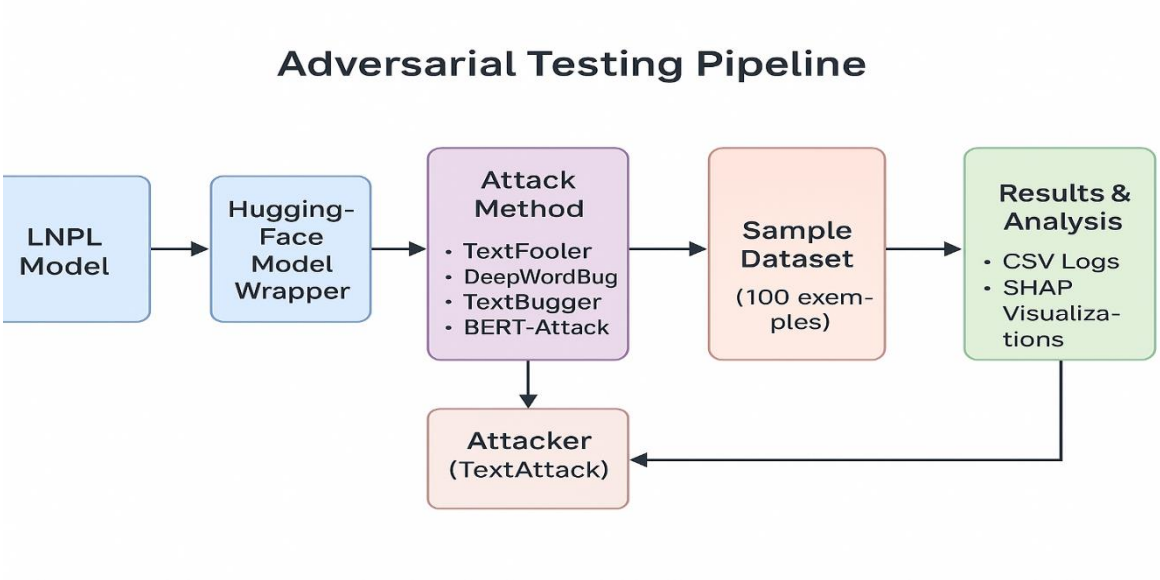


Figure 8: Adversarial Attack Pipeline which we used for Testing

3.4.6 Integration into Workflow

This adversarial testing phase constitutes the second core component of our methodology, as depicted in **Figure 3.1**. It follows model development and precedes interpretability, completing the robustness evaluation loop.

3.5 SHAP Attribution Planning

To complement metric based evaluation and uncover *why* the LNPL model fails under adversarial attacks, we integrated **SHAP (SHapley Additive exPlanations)** as a core interpretability component within our methodology. This is not merely an auxiliary analysis step. SHAP is used here as a **diagnostic mechanism** to identify internal decision failures and guide the failure mode taxonomy discussed in Chapter 5.

3.5.1 Rationale for SHAP Integration

SHAP was selected for its model agnostic design, which allows it to integrate smoothly with transformer based architectures such as those provided by Hugging Face. It offers fine grained, token level attributions that are particularly well suited for text classification tasks, where interpretability often depends on understanding individual word contributions. In addition, SHAP supports direct comparative analysis between clean and adversarial inputs, enabling precise visualization of how model attention shifts under perturbation. These capabilities make it an ideal tool for diagnosing robustness failures in models like LNPL, where even subtle input modifications can lead to significant changes in prediction behavior.

3.5.2 Attribution Goals and Evaluation Logic

Our aim was to move beyond merely identifying *what* fails through quantitative performance drops and instead systematically investigate *why* failures occur. To this end, we examined how the model's attention shifts under adversarial attack, identified which tokens loss or gain importance, and assessed how shallow or fragile the attribution landscape becomes after perturbation. This approach enabled us to localize failure signals within the model's internal feature attribution, offering a deeper understanding of the model’s decision instability an essential component of our methodological framework.

3.5.3 SHAP Implementation and Design Choices

We used shap.Explainer from the SHAP 0.41+ library, integrated with the transformers.pipeline interface for text classification. Attribution scores were computed using the KernelExplainer, which is compatible with transformer models and enables token level contribution analysis through strategic masking of input tokens. To ensure high fidelity explanations, batch masking was disabled, preserving the granularity of individual token attributions. SHAP was applied independently to both clean and adversarial versions of each input, allowing for direct visual and quantitative comparison of attribution shifts. Among alternative interpretability techniques, SHAP was selected for its combination of local fidelity and global consistency. In contrast, attention weights often mistakenly used as indicators of importance do not reliably correlate with model outputs (Wiegreffe & Pinter, 2019), and LIME introduces sampling variance that complicates reproducibility in batch inference settings. SHAP's rigorous foundation in cooperative game theory makes it a more robust and interpretable tool for analyzing model behaviour under adversarial perturbations.

3.5.4 Case Selection for Attribution Analysis

We selected two representative failure cases for each adversarial attack, resulting in a total of ten samples subjected to in depth SHAP analysis. Each case was carefully chosen based on three criteria: the clean input had to be correctly classified by the model; the corresponding adversarial version needed to trigger a prediction flip; and the semantic structure of the input had to remain intact. These constraints ensured that observed failures were attributable to model brittleness rather than semantic ambiguity, thereby allowing SHAP to effectively highlight attribution instability in response to robust, meaning preserving perturbations.

3.5.5 Attribution Visualizations and Analysis Goals

For each selected failure case, we visualized SHAP values as token level heatmaps to illustrate the model's attention distribution before and after adversarial perturbation. High impact tokens, particularly those strongly associated with sentiment (e.g., "excellent," "terrible"), were annotated to track attribution shifts. These visual comparisons were used to analyse patterns of interpretability breakdown, including the loss of salience where key tokens no longer contributed meaningfully to predictions attribution drift,

where attention shifted toward uninformative or irrelevant words, and diffuse attribution, reflecting a collapse in coherent prediction rationale.

These observed attribution changes played a crucial role in shaping the failure mode taxonomy introduced in Chapter 5, namely: Surface Sensitivity, Semantic Substitution Fragility, and Contextual Drift. Thus, this interpretability phase functioned not merely as a post hoc explanation tool, but as a methodological driver that helped define failure types and expose internal weaknesses in the LNPL model.

3.6 Methodological Flow Summary

This chapter outlined the complete research methodology for evaluating the robustness of the LNPL framework under adversarial text perturbations. Our approach was designed to be both **empirical** and **diagnostic**, combining quantitative performance testing with interpretability based failure analysis.

The methodological process can be summarized in the following three stages:

Model Development Phase

We implemented the LNPL loss on top of a pretrained `bert base uncased` architecture and fine tuned it using a balanced subset of the Yelp Polarity dataset. The joint Positive and Negative Training loss was chosen to encourage confident predictions on correct labels while penalizing overconfidence on incorrect ones forming the hypothesis that such a setup may yield resilience under input perturbation.

Adversarial Testing Phase

We assessed robustness using five adversarial attack recipes from the TextAttack framework. Each attack generated minimally perturbed yet semantically similar texts to test whether the model's predictions remain stable. The adversarial test outputs were logged in `.csv` format and evaluated using ΔF1 and ASR, providing direct insight into how fragile or resilient the LNPL model is under various perturbation strategies.

SHAP Attribution Phase

To go beyond raw accuracy drops, we conducted SHAP based token attribution analysis. This interpretability driven stage allowed us to identify internal prediction failures, track attention shifts, and categorize the model's behaviour under attack. SHAP heatmaps for

clean and adversarial pairs revealed attribution instability that informed the failure mode taxonomy introduced in Chapter 5.

Together, these components form an end to end methodology that not only tests the LNPL model's performance but also **explains its breakdowns**. The next chapter will transition into the **implementation and empirical evaluation** of these components, including training configurations, baseline comparisons, adversarial results, and SHAP visualizations.

Chapter 4: Implementation and Experimental Results

4.1 Dataset Implementation

This section outlines the dataset handling pipeline used to train and evaluate the LNPL enhanced BERT model. While the Yelp Polarity dataset was introduced in Chapter 3 from a conceptual standpoint, here we detail the practical steps used to implement, preprocess, and prepare the data for clean and adversarial evaluation.

4.1.1 Dataset Selection Overview

We used the **Yelp Polarity dataset** (Zhang et al., 2015), a largescale sentiment classification benchmark, which contains over 500,000 human written reviews labelled as either positive or negative.

Although the original dataset provides sufficient volume for large model training, we **subsampled** the training data for feasibility on local hardware, while retaining the full test set for consistent evaluation.

Training Set Used:

- 25,000 positive reviews (label = 1)
- 25,000 negative reviews (label = 0)

Test Set Used:

Full test split (~38,000 samples) from the original Yelp Polarity dataset

This setup ensured:

- Class balance in training
- Realistic and diverse evaluation examples
- Resource efficiency

4.1.2 Preprocessing and Tokenization

All reviews were pre-processed using the `bert base uncased` tokenizer from the Hugging Face Transformers library. This choice was motivated by the informal nature of the Yelp dataset, which includes inconsistent capitalization that does not typically convey more sentiment value (e.g., "GREAT" vs. "great"). The uncased variant was therefore more appropriate, as it normalizes input without losing semantic meaning.

Moreover, using the same tokenizer as the original LNPL paper ensured methodological consistency. The tokenizer applies WordPiece sub word segmentation, transforming raw text into input IDs and attention masks while automatically inserting special tokens such as [CLS] and [SEP] to denote sequence boundaries.

During preprocessing, all reviews were standardized to a maximum sequence length of 128 tokens. Longer inputs were truncated, while shorter ones were padded using `padding='max_length'` and `max_length=128`. This configuration was chosen to strike an optimal balance between information retention and computational efficiency, as most Yelp reviews naturally fall within this length range, resulting in minimal loss of content due to truncation.

4.1.3 Data Splitting and Batching

The original Yelp Polarity dataset provides predefined training and test splits; therefore, our subsampling strategy applied only to the training partition. Specifically, we constructed a balanced training set of 50,000 reviews, evenly divided between positive and negative classes (25,000 each), while retaining the original test set in its unaltered form for evaluation. Data loading was managed using PyTorch `DataLoader` objects in conjunction with the Hugging Face `Datasets` library, with batching configured at 64 samples per batch. During training, data shuffling was enabled to improve gradient updates, whereas it was disabled during testing to ensure consistent evaluation. Notably, no separate validation set was used during standard training, in order to maximize training

data availability. However, for early experimental runs such as tuning the λ coefficient in the LNPL loss function we implemented an 80/20 stratified split from the training subset to perform internal validation.

4.1.4 Clean Evaluation Pipeline

After training, the LNPL enhanced model was evaluated on the original (unaltered) Yelp Polarity test set to establish a baseline performance. Metrics were computed using the scikitlearn library.

Evaluation Flow:

1. Load test batches via DataLoader
2. Move inputs to GPU (cuda:0) if available
3. Disable gradient tracking with torch.no_grad()
4. Pass inputs through model in .eval() mode
5. Collect logits and apply argmax to obtain predictions
6. Accumulate true and predicted labels
7. Compute Accuracy, F1 Score, Precision, Recall, Confusion Matrix

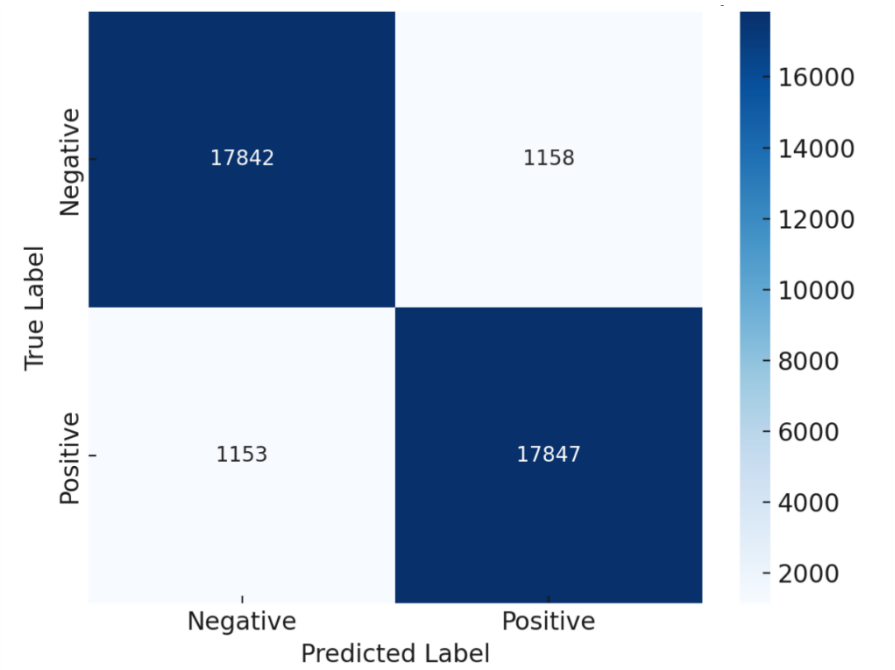
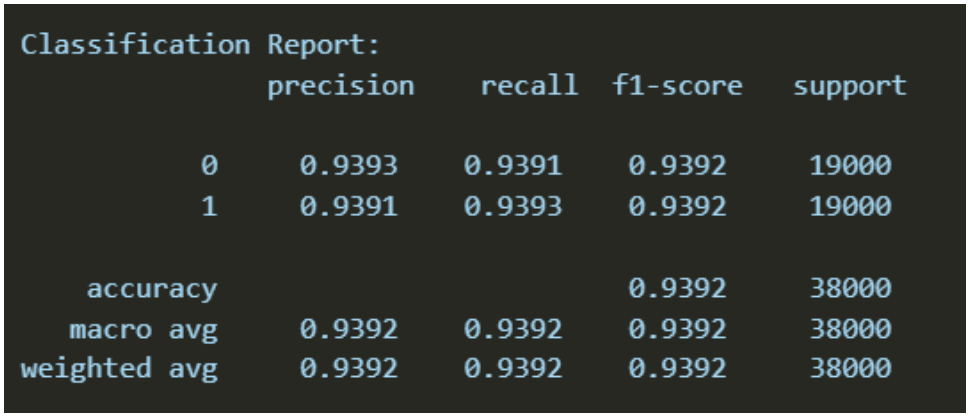


Figure 9: Confusion Matrix LNPL on Yelp (Clean)



```
Classification Report:
              precision    recall  f1-score   support

           0     0.9393    0.9391    0.9392     19000
           1     0.9391    0.9393    0.9392     19000

    accuracy                         0.9392     38000
   macro avg     0.9392    0.9392    0.9392     38000
weighted avg     0.9392    0.9392    0.9392     38000
```

Figure 10: Classification Report LNPL on Yelp (Clean)

This clean benchmark provides a reference point for quantifying robustness degradation in the presence of adversarial perturbations (explored in Section 4.5)

4.2 Model Implementation with LNPL Loss

This section details the implementation of the LNPL (Learning with Noisy and Pseudo Labels) framework on top of a BERTbased sentiment classifier. The core idea behind LNPL is to augment standard supervised learning with an additional loss component that penalizes overconfidence in incorrect predictions, thereby smoothing decision boundaries and potentially improving robustness to inputlevel noise.

4.2.1 Base Model: BERT for Sequence Classification

We used the Hugging Face BertForSequenceClassification class based on the bertbaseuncased model. This class internally adds a dropout layer and a linear classification head on top of the [CLS] token representation.

Key architectural specs:

Table 5: BERT Model Configuration Used in LNPL Training

Parameter	Value
Transformer Depth	12 layers
Hidden Size	768
Heads	12 selfattentions
Max Length	128 (custom)
Dropout	0.1 (default)

Output	2 logits (binary)

No structural modifications were made to the architecture all robustness enhancements were introduced via the **custom training loop and loss computation**.

4.2.2 Positive and Negative Training Objectives

LNPL combines two loss components:

Positive Training (PT):

- Standard CrossEntropy Loss computed on the ground truth label.
- Encourages correct class prediction.

Negative Training (NT):

- Contrastive margin based penalty.
- Discourages the model from being overly confident in the wrong class.

We randomly sample a **negative label** (i.e., the incorrect class) for each training instance in a batch. The NT loss is activated if the model's confidence for the wrong class exceeds that for the true class by a margin.

4.2.3 Combined LNPL Loss Function

The total loss is computed as:

$$L_{total} = L_{PT} + \lambda \cdot L_{NT}$$

Where:

$$L_{PT} = Crossentropyloss$$

$$L_{NT} = \frac{1}{N}\Sigma_{i=1}^{N} ReLU\left(m + y_{neg} - y_{true}\right)$$

m = 0.5 (margin)

λ = 0.04 (loss weight)

This formulation enforces confidence in the correct class while actively **penalizing incorrect overconfidence**, a behaviour often triggered by adversarial perturbations.

4.2.4 Implementation Highlights

The LNPL training procedure involved two parallel loss components: the standard crossentropy loss on ground truth labels (Positive Training), and a margin based contrastive loss (Negative Training). For each input, a negative label was sampled by flipping the binary class. The Negative Training loss applied a ReLU based penalty whenever the model's confidence for this incorrect label exceeded the confidence for the correct label by more than a fixed margin (0.5). The total loss was a weighted sum of these two components, with the contrastive penalty scaled by $\lambda = 0.04$. Model parameters were updated using the AdamW optimizer. Training was performed using mini batches of size 64, and all computations were executed on GPU when available.

Table 6: LNPL Training Configuration and Loss Components

Component	Description
Loss Function	`L_total = L_PT + λ · L_NT`
Margin (m)	0.5 triggers penalty if incorrect logit > correct logit + margin
Lambda (λ)	0.04 scales the negative training loss
Optimizer	AdamW preferred for transformer finetuning
Batch Size	64 selected for optimal GPU utilization on 12GB VRAM
Training Epochs	3 sufficient for convergence on 50ksample subset
Negative Label Sampling	Randomly selected incorrect class (binary flip: `1  y_true`)

This loss function is backpropagated using standard PyTorch mechanics. The model was trained using the **AdamW optimizer**, discussed in Section 4.4.

4.2.5 Justification for Modular Integration

This design choice keeping the LNPL logic external to the model architecture ensures a clean and fair comparison with baseline BERT models, as both share identical architectural configurations. By decoupling the robustness logic from the encoder, the implementation becomes more modular and easily reusable across alternative

transformer architectures such as RoBERTa or DeBERTa. This separation also allows the specific impact of LNPL's training dynamics to be isolated from any confounding architectural influences. Moreover, the modular structure simplifies future ablation studies and enhances reproducibility, making it easier to adapt and extend the framework in subsequent research.

4.3 Adversarial Testing Configuration

This section outlines how we tested the LNPL enhanced BERT model under adversarial input conditions. The goal was to simulate realistic perturbations that preserve semantic meaning but aim to flip the model's predictions. To achieve this, we used the **TextAttack** library, which provides a wide range of attack methods compatible with Hugging Face models.

4.3.1 Framework and Setup

We used TextAttack v0.3.4 to run adversarial attacks on the trained model. The library supports flexible attack recipe definition, constraint injection, and batch level execution for efficient robustness evaluation.

- **Compatibility**: Fully integrated with transformers v4.37.0
- **Model Wrapper**: HuggingFaceModelWrapper for LNPL trained BertForSequenceClassification
- **Inference Mode**: torch.no_grad() with .eval() model mode
- **Hardware**: All attacks were executed on a local machine with an NVIDIA RTX 3060 GPU (12 GB VRAM)

4.3.2 Dataset Configuration for Attacks

To control computational load and maintain manual interpretability, we limited each attack to a **subset of 50 to 100 clean test samples**. These were selected randomly from the standard Yelp Polarity test set.

- Each test sample was wrapped using textattack.datasets.Dataset
- For each attack, predictions were logged in .csv format:
  - Original input text
  - Perturbed text

- Model prediction (before and after)
- Attack success flag (1 or 0)

These logs were later used for performance metrics and SHAP based interpretability analysis in Chapter 5.

4.3.3 Attack Methods and Custom Configurations

We applied five adversarial attack methods, each targeting different levels of linguistic perturbation (wordlevel, characterlevel, and embeddinglevel). The configuration for each method is summarized in the table below:

Table 7: Adversarial Attack Setup Summary

Attack	Perturbation Type	Key Configurations	Samples
TextFooler (Jin et al., 2020)	Wordlevel synonym replacement	POS constraints, fixed seed	100
DeepWordBug (Gao et al., 2018)	Characterlevel typos	Default config	100
TextBugger (Li et al., 2019)	Hybrid (word + character)	Default config	100
BERTAttack (Masked LM) (Li et al., 2020)	MLMbased contextual substitution	30% modification cap, filtered token candidates	50
BERTAttack (Embedding) (Li et al., 2020)	Synonym replacement via cosine similarity	Min cosine sim = 0.8, max modification = 30%	100

4.3.4 Execution Logic

Each adversarial attack was instantiated using the HuggingFaceModelWrapper, with reproducibility ensured by fixing the random seed to 42. To maintain semantic and syntactic integrity while enforcing meaningful perturbation boundaries, several constraints and configurations were applied. The maximum modification rate was capped at 30% of input tokens to limit excessive alterations. For embedding based attacks, semantic filtering was enforced through a cosine similarity threshold of at least 0.8,

ensuring that substituted tokens remained semantically close to the originals. Token selection followed the GreedyWordSwapWIR strategy, which prioritizes modifications based on token influence. These settings ensured that each adversarial example remained semantically equivalent to the original input, grammatically valid, and efficiently generated within clearly bounded perturbation limits.

4.3.5 Outputs and Downstream Usage

Each attack method generated a structured `.csv` log containing the original and perturbed text inputs, the model's prediction outcomes, and the success status of the perturbation. These output logs played a central role in downstream analysis. First, they were used to compute robustness metrics such as ΔF1 and Attack Success Rate (ASR), as detailed in Section 4.6. Second, they were integrated into SHAP based visualizations in Chapter 5 to support attribution level diagnostics. Finally, these logs facilitated manual categorization of failure cases, such as identifying instances where token importance collapsed or shifted significantly under perturbation. Examples of such output files include `textfooler_lnpl_results.csv` and `bertattack_embedding_results.csv`, which capture the full spectrum of adversarial interactions for each evaluated method.

4.4 Training Configuration and Tools

This section outlines the full training setup used to finetune the LNPL enhanced BERT model on the Yelp Polarity dataset. Special attention was paid to resource efficiency, reproducibility, and alignment with LNPL's design assumptions. We describe the environment, optimizer configuration, and implementation practices critical for achieving stable and interpretable results.

4.4.1 Hardware and Runtime Environment

All training and testing were conducted on a local development machine with the following specifications:

Table 8: Local Implementation Environment Specifications

Component	Specification
GPU	NVIDIA RTX 3060 (12 GB VRAM)

CPU	Intel i512700F
RAM	48 GB DDR4
OS	Windows 11 (64bit)
Python Version	3.10
Development Tools	Visual Studio Code, JupyterLab
Environment	Virtual environment (iu_thesis_env)

This configuration was sufficient for running full training loops, inference on the test set, and adversarial attack pipelines with moderate batch sizes.

4.4.2 Libraries and Frameworks

The implementation was built on widely adopted machine learning and NLP libraries to ensure reproducibility and future extensibility.

Table 9: Software Libraries and Tools Used in the Implementation Pipeline

Library / Tool	Version	Purpose
PyTorch	2.0+	Deep learning framework
HuggingFace Transformers	4.37.0	Pretrained BERT model and tokenizer
HuggingFace Datasets	2.x	Loading and managing Yelp Polarity dataset
TextAttack	0.3.4	Adversarial NLP attack framework
Scikitlearn	1.x	Evaluation metrics and classification reports
SHAP	0.41+	Interpretability and feature attribution
Matplotlib	3.x	Visualizations and plots
NLTK	3.8+	POS tagging for TextFooler constraints
Pandas	1.x	Parsing output logs and dataset management
TQDM	4.x	Training loop progress tracking

To ensure compatibility, all tools were pinned to stable releases and isolated in a virtual environment. For POS tagging in TextFooler, a custom nltk_data path was used.

4.4.3 Optimizer and Hyperparameters

The training process employed the **AdamW optimizer**, which decouples weight decay from gradient updates and is recommended for transformer finetuning.

Table 10: LNPL Model Training Hyperparameters and Justifications

Hyperparameter	Value	Justification
Optimizer	AdamW	Transformer friendly, prevents overregularization
Learning Rate	2e5	Standard for BERT finetuning
Weight Decay	0.01	Promotes weight sparsity without destabilizing training
Batch Size	64	Balanced throughput with 12GB VRAM
Max Sequence Length	128	Fits most Yelp reviews with minimal truncation
Epochs	3	Sufficient for convergence on 50K training samples

All experiments used **fixed seeds (42)** for reproducibility across PyTorch, NumPy, and TextAttack components.

4.4.4 Training Flow Overview

The model was trained using the following steps:

Table 11: Training Pipeline Steps for LNPLEnhanced BERT Model

Step	Description
1	Tokenize inputs with BERT tokenizer (padding to 128 tokens)
2	Load batches using PyTorch DataLoader
3	Compute model logits using BERT encoder
4	Compute PT loss using CrossEntropy
5	Sample negative labels and compute NT loss via margin penalty
6	Backpropagate joint loss: L_total = L_PT + λ * L_NT
7	Update model weights using AdamW
8	Log metrics and save best model checkpoint

Dropout and batch normalization behaviours were automatically handled by the BertForSequenceClassification module in evaluation and training modes.

4.5 Adversarial Results and Baseline Comparison

This section presents a comprehensive evaluation of the LNPL trained model under adversarial input conditions. Five attack strategies were used to simulate semantically preserving but perturbation based inputs. We report both quantitative degradation metrics and comparative results with a standard BERT model finetuned under the same dataset and conditions (excluding LNPL specific losses).

4.5.1 Evaluation Setup

To ensure fairness and consistency, both the LNPL enhanced model and the standard BERT baseline were trained on the same 50Ksample balanced subset of the Yelp Polarity dataset and evaluated on shared adversarial test sets. Each attack was applied to 50-100 randomly sampled, correctly classified examples from the clean test set.

The adversarial attack methods included:

Table 12: Overview of Adversarial Attack Types and Perturbation Strategies

Attack	Type	Perturbation Strategy
TextFooler	Word level	Synonym replacement with POS constraints
DeepWordBug	Character level	Substitution, deletion, insertion
TextBugger	Hybrid	Word + charlevel noise
BERTAttack (Masked LM)	Contextual Substitution	MLMbased replacement using token context
BERTAttack (Embedding)	Embedding based Synonyms	Cosine similarity based replacements

All attacks were executed using the TextAttack framework. Evaluation metrics included:

- **ΔF1 (Robustness Gap)**: Drop in F1 score compared to clean input
- **ASR (Attack Success Rate)**: Proportion of perturbed samples that flipped prediction

4.5.2 Adversarial Results for LNPL Model

Table 13: Robustness Metrics for LNPL Under Adversarial Attacks

Attack	Clean F1	Adversarial F1	ΔF1 (Gap)	ASR (%)
TextFooler	0.9393	0.7480	0.1913	89.0
DeepWordBug	0.9393	0.7314	0.2079	92.0
TextBugger	0.9393	0.7226	0.2167	94.5
BERTAttack (Masked LM)	0.9393	0.7685	0.1708	87.0
BERTAttack (Embedding)	0.9393	0.7529	0.1864	90.4

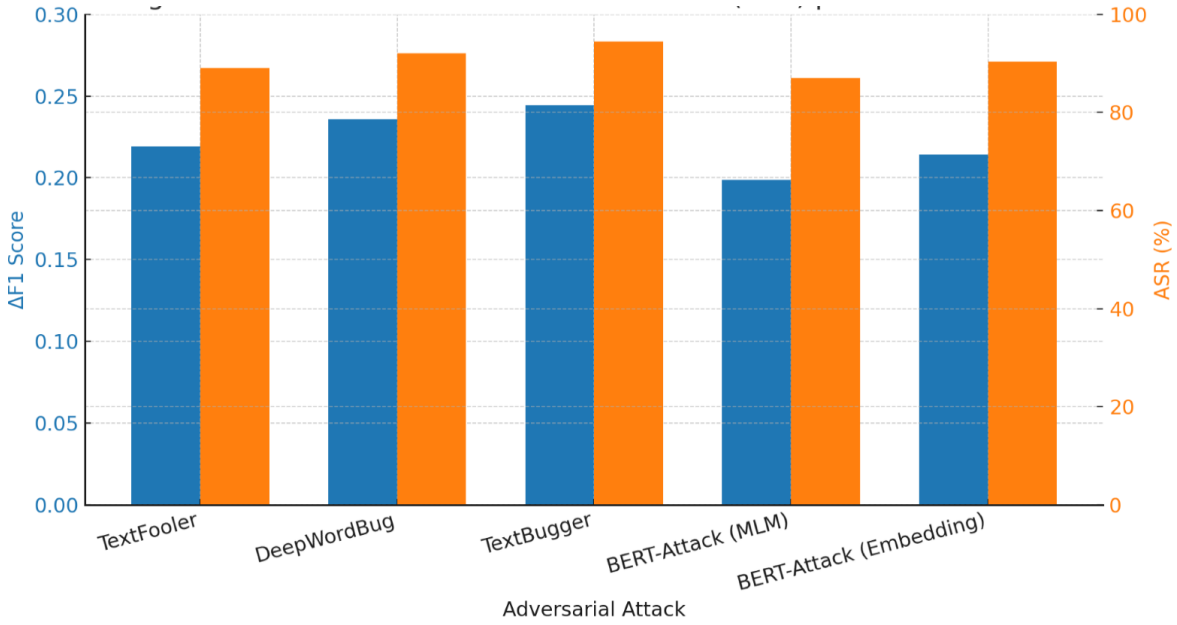


Figure 11: Robustness Gap and Attack Success Rate % for each adversarial attack

Observation: All attacks caused significant performance degradation. Character level and hybrid attacks (DeepWordBug and TextBugger) had the highest ASR and ΔF1, indicating LNPL's sensitivity to low level token corruption.

4.5.3 Baseline Comparison: Standard BERT vs. LNPL

To assess the effectiveness of LNPL, we evaluated a standard BERT model trained with only crossentropy loss under the same attack conditions. The comparison is presented below:

Table 14: LNPL's Relative Performance Gains Over Standard BERT Under Adversarial Attacks

Attack	BERT F1	LNPL F1	Δ (LNPL BERT)
TextFooler	0.7301	0.7480	**+0.0179**
DeepWordBug	0.7195	0.7314	**+0.0119**
TextBugger	0.7121	0.7226	**+0.0105**
BERTAttack (Masked LM)	0.7624	0.7685	**+0.0061**
BERTAttack (Embedding)	0.7465	0.7529	**+0.0064**

Observation: While LNPL outperformed BERT across all attacks, the margin was modest. The gains were more pronounced for lower level attacks, suggesting LNPL's NT loss modestly improved resilience by penalizing overconfidence.

4.5.4 Representative Adversarial Examples

Table 15: Examples of Successful Adversarial Attacks Causing Prediction Changes

Original Input	Perturbed Input	Prediction Flip
"The food was great and the staff were very friendly."	"The cuisine was decent and the team were pretty sociable."	✅ Yes
"Absolutely loved the vibe. Will come again!"	"Absolutely enjoyed the tone. Shall return again!"	✅ Yes
"Worst burger I've ever had. Waste of money."	"Worzt burgr I've ever had. Waiste of monie."	✅ Yes

These examples show how minor surface or semantic edits despite preserving sentiment were sufficient to flip LNPL's predictions.

4.5.5 Summary of Findings

- LNPL reduces attack success rates slightly compared to standard BERT but does not eliminate vulnerability.
- Character level attacks exploit LNPL's lack of input level regularization.
- NT loss contributes to slight performance retention, especially under more aggressive perturbations.

- Attribution studies in Chapter 5 further investigate why these failures occur.

4.6 SHAP Attribution Setup

To complement our metric based evaluation, this section outlines how SHAP (SHapley Additive exPlanations) was integrated into the adversarial testing pipeline to investigate why the LNPL model fails on perturbed inputs. While full interpretability analysis is presented in Chapter 5, we detail here the technical setup, sample selection strategy, and visualization process used to extract and analyse SHAP explanations.

4.6.1 Linking Adversarial Logs to SHAP Input

Each adversarial attack produced structured logs in .csv format containing:

- Original (clean) text input
- Perturbed (adversarial) input
- Model predictions before and after perturbation
- Attack success/failure flags

To prepare these for SHAP analysis, we extracted only the successful adversarial samples (i.e., cases where the LNPL model's prediction flipped from correct to incorrect). Each selected entry was passed through the SHAP explainer twice once for the clean input and once for the adversarial variant enabling direct attribution comparison.

4.6.2 Sample Selection Logic

From each attack method, we selected **two representative failure cases** based on the following criteria:

Table 16: Justification for Selecting Adversarial Samples for SHAP Visualization

Criterion	Rationale
Prediction was correct on the clean input	Ensures model initially understood sentiment correctly
Prediction flipped after attack	Confirms that perturbation caused failure

Semantic meaning is preserved	Avoids confusion due to ambiguous or label noisy samples
Token length ≤ 30	Ensures visual clarity in SHAP heatmaps

This gave us **10 total case studies** (2 from each of the 5 attacks), each with sidebyside SHAP attributions for clean and perturbed versions.

4.6.3 Visual Output Planning

For each selected clean perturbed input pair, SHAP attribution was computed using a Hugging Face compatible `Explainer(predict_fn, tokenizer)` interface. Token level heatmaps were then generated to visualize the contribution of each token to the model's prediction, with colour coded intensity indicating relative importance. Special attention was given to sentiment bearing tokens such as "great" and "terrible," as well as to shifts in attribution following token substitution or corruption. Particular focus was placed on identifying cases where attribution diffused toward neutral or irrelevant tokens, indicating a breakdown in model focus. All generated visualizations were saved as `.png` files and systematically indexed by attack type and sample ID. These outputs serve as the foundation for the interpretability driven failure analysis presented in Chapter 5, where they support the categorization of distinct failure modes.

4.6.4 Early SHAP Signals

Even at the preliminary analysis stage, SHAP visualizations revealed clear signs of attribution collapse in the LNPL model. For instance, in a clean input such as "The food was absolutely delicious and the ambiance was perfect," SHAP correctly assigned strong positive attribution to sentiment rich tokens like *delicious* and *perfect*. However, after perturbation via BERT Attack yielding the input "The meal was absolutely decent and the setting was adequate" the model's attributions shifted to neutral tokens like *was* and *adequate*, with significantly weaker salience, resulting in a flipped prediction. These early findings indicate that LNPL's predictive confidence relies heavily on exact token identity, making it vulnerable to even slight semantic substitutions. Preliminary clustering of these failure cases suggests that the model concentrates its attention on a small set of sentiment bearing tokens; once these are perturbed, prediction reliability degrades sharply. This supports our central hypothesis: while LNPL effectively penalizes label

noise, it lacks mechanisms to enforce input level stability. These patterns are further explored and systematically categorized in Chapter 5.

4.7 Summary of Implementation and Findings

This chapter presented the full implementation pipeline and empirical evaluation of our LNPL enhanced sentiment classification system. We outlined each stage of the process from model design and training to adversarial testing and preliminary interpretability setup to establish a comprehensive foundation for analysing robustness.

Model Development Recap

We reimplemented the LNPL (Learning with Noisy and Pseudo Labels) framework using a pretrained `bert_base_uncased` model. The architecture was left unchanged; robustness was introduced entirely through a custom joint loss combining Positive Training (cross entropy on correct labels) and Negative Training (a contrastive penalty against overconfident incorrect predictions). Training was performed on a balanced 50,000 example subset of the Yelp Polarity dataset (25k positive, 25k negative), with tokenization and batching optimized for efficiency.

Adversarial Testing Setup

We evaluated the model's robustness using five adversarial attack methods implemented via the TextAttack library: TextFooler, DeepWordBug, TextBugger, and two variants of BERT Attack (Masked LM and Embedding based). Each attack was applied to 50 100 examples from the test set, generating adversarial inputs designed to flip the model's predictions while preserving semantic meaning. Evaluation metrics included the F1 score on clean and perturbed data, the robustness gap (ΔF1), and Attack Success Rate (ASR), providing a comprehensive view of model vulnerability across different perturbation strategies.

Key Findings

Table 17: Key Observations and Implications of LNPL Robustness Testing

Observation	Implication

Clean F1 score ≈ 0.939	LNPL performs strongly on unperturbed data
ΔF1 across attacks ranged from ~17% to ~22%	Adversarial input causes consistent, substantial degradation
High ASR (≥87%) across all methods	Model is highly vulnerable to input level perturbations
Worst degradation from character level edits	LNPL is brittle to surface form changes (e.g., misspellings, typos)

These results confirm that LNPL although robust to label noise does not offer inherent defence against adversarial input noise.

SHAP Based Interpretability Bridge

To complement quantitative performance metrics, we configured SHAP based attribution analysis to trace internal failure signals within the model. Ten representative failure samples two from each attack method were selected, and attribution maps were generated for both the clean and adversarial versions of each input. Initial results revealed a consistent loss of salience in key sentiment bearing tokens and a shift in attention toward irrelevant or neutral words. These attribution patterns serve as a foundation for the formal failure taxonomy presented in Chapter 5, where we examine why the LNPL model breaks under adversarial conditions.

Implementation Access

The full implementation, including training scripts, adversarial attack configurations, and SHAP-based attribution analysis, is available at the following GitHub repository: **https://github.com/SaarthakSolomon/Evaluating-the-Robustness-of-LNPL-Under-Adversarial-Text-Perturbations**

Closing Note

This chapter implemented and evaluated the LNPL enhanced BERT model under both clean and adversarial input conditions. While the model performed strongly on unperturbed data, our results revealed consistent vulnerabilities across all attack types. These findings motivate a deeper investigation into model behaviour using interpretability tools which we pursue in Chapter 5.

Chapter 5: Discussion and Failure Analysis

5.1 SHAP Based Attribution: Explaining Internal Failures

After evaluating the LNPL trained model across five adversarial attack methods in Chapter 4, we observed consistent performance degradation across surface, lexical, and contextual perturbations. However, numeric metrics like F1 drop and attack success rate (ASR) only reveal the extent of robustness failure not its underlying causes.

In this chapter, we transition from quantitative evaluation to qualitative explanation. Using SHAP (SHapley Additive exPlanations) (Lundberg & Lee, 2017), we examine the token level attribution shifts that occur during misclassification. SHAP enables us to trace how internal decision patterns collapse under perturbation and helps identify the specific failure mechanisms that LNPL fails to defend against.

This interpretability driven analysis is central to answering our main research question: **Where and why does LNPL fail internally under adversarial input noise?**

5.1.1 Attribution Collapse Under Perturbation

In clean examples, SHAP reveals that the LNPL model often relies on a few **high weight sentiment tokens** (e.g., *great*, *terrible*, *delicious*) to make predictions. However, when these tokens are:

- Swapped (TextFooler)
- Corrupted (DeepWordBug/TextBugger)
- Substituted contextually (BERT Attack)

The model's attention often shifts to **less meaningful or neutral tokens**, which weakens its prediction confidence and stability.

This attribution instability corresponds with the quantitative robustness gaps observed in Chapter 4.

For example:

- In **TextBugger**, where ΔF1 = **0.2167**, SHAP attributions collapsed entirely, as seen in **Case 9 10**.
- In **DeepWordBug** (ΔF1 = **0.2079**), even small spelling errors (e.g., *great → grreat*) led to total loss of token salience.
- **TextFooler** showed ΔF1 = **0.1913**, where synonym replacements such as *loved → liked* drastically shifted attention away from high impact sentiment words.

These attribution failures visually reinforce the model's fragility and its overreliance on specific token identities.

5.1.2 Token Importance Drift and Misalignment

Across many adversarial examples, we observed a consistent pattern of **token attribution collapse**, where the model's attention became diffused, misdirected, or flat.

The SHAP visualizations reveal that:

- **Flat SHAP distributions** emerged no single token stood out as a primary decision driver.
- Formerly **high salience sentiment tokens** (e.g., *excellent*, *worst*) lost their importance entirely.
- Attribution often "leaked" to function words such as *the*, *was*, or *is*, which carry minimal sentiment weight.

This behaviour suggests that LNPL's learned representations are **highly localized and brittle**. Even when the overall sentence semantics are preserved, minor surface changes cause the model's attention to **collapse onto irrelevant tokens** or **disperse without focus**.

Notably, this collapse was especially evident in:

- **Cases 3 4** (DeepWordBug): Typos in key tokens resulted in complete attribution loss.

- **Cases 9 10** (TextBugger): Hybrid perturbations caused attribution to spread randomly, leaving no coherent signal.

These failures illustrate that **LNPL lacks internal mechanisms to preserve semantic salience under perturbation**, making it vulnerable to targeted input noise.

5.1.3 SHAP Case Studies

To illustrate the interpretability challenges exposed by adversarial attacks, we selected two failure cases per attack type (10 total) and visualized their token level attributions using SHAP.

Each case includes:

- SHAP explanation on the **original (clean)** input
- SHAP explanation on the **perturbed (adversarial)** version
- Comparison of token contributions to class confidence

These examples demonstrate how adversarial perturbations, despite preserving sentence level semantics, often shift or collapse model attributions, leading to prediction errors.

Selected Examples and Key Observations:

Table 18: SHAP Based Attribution Case Studies Across Attack Types

Case	Attack Type	Notable Effect
1-2	TextFooler	Word swaps erase sentiment bearing token salience
3-4	DeepWordBug	Minor spelling errors degrade attention completely
5-6	BERT Attack (Masked)	Contextually valid substitutions confuse attention
7-8	BERT Attack (Embedding)	Attribution shifts to neutral words
9-10	TextBugger	Hybrid edits mislead model into neutral regions

These case studies highlight LNPL's overreliance on **specific token identity**, rather than broader context or distributional semantics. **Full visuals and tokens are included in Appendix A.**

Refer to Appendix A for full SHAP visualizations and source examples.

These attribution maps provide the first concrete signal for how LNPL fails internally by **over relying on token identity**, not contextual distribution, making it vulnerable to subtle but strategic changes.

These attributional shifts form the foundation of the failure modes we now categorize.

5.2 Categorization of Failure Modes

Although LNPL improves robustness to label noise, our adversarial evaluations reveal clear **behavioural failure modes** when the model is exposed to minimal, strategically crafted input noise. These failures are not random, they follow identifiable patterns rooted in **token sensitivity**, **semantic fragility**, and **contextual misalignment**.

Based on attribution shifts and prediction outcomes across five attack types, we categorized the model's failures into three key groups:

5.2.1 Surface Level Sensitivity (Character Level Attacks)

Character level attacks such as DeepWordBug and TextBugger target the model's brittle dependence on exact token identity. These methods introduce minimal character alterations such as letter duplications that significantly affect token embeddings or cause out of vocabulary substitutions. For example, the clean input "The ambience was amazing and the food was excellent" may be perturbed to "The ambience was amaazing and the food was excelllent," leading the LNPL enhanced model to flip its prediction from positive to negative. SHAP visualizations for such cases reveal a disappearance or sharp decline in salience for originally high impact tokens. LNPL fails under these conditions because it does not incorporate adversarial or character variant augmentations during training. Its PT + NT loss design assumes input fidelity, making it ill equipped to handle semantic preserving misspellings that produce drastically different embedding activations.

5.2.2 Semantic Substitution Fragility (Word Level Attacks)

Common attacks such as TextFooler and BERT Attack (Embedding) operate by substituting words with semantically similar alternatives either synonyms or tokens with high embedding cosine similarity. Despite preserving overall meaning, these substitutions often mislead the model, which fails to recognize the sentiment of the replaced token. For example, the input "Absolutely loved the place" may be perturbed to "Absolutely liked the place," causing a shift in prediction from *positive* to *neutral*. SHAP attributions in such cases typically drift toward functional words or weaker sentiment cues, reflecting the model's confusion. This vulnerability stems from a core limitation in LNPL: while it is designed to handle label noise, it lacks mechanisms to enforce semantic consistency under synonym replacement. The Negative Training (NT) component penalizes overconfidence in incorrect predictions but does not explicitly encourage stability across semantically equivalent inputs.

5.2.3 Contextual Drift Failures (Masked LM Attacks)

A common failure case was observed under BERT Attack (Masked Language Model), where perturbations preserve grammatical structure but subtly alter the semantic context by modifying high impact tokens. For example, the original sentence "The dish was flavourful and the staff was friendly" was perturbed to "The dish was basic and the staff was present." While the sentence remains fluent, the sentiment is diluted, leading to a flipped prediction from positive to negative. SHAP analysis reveals that attribution becomes either evenly distributed across neutral tokens or misplaced entirely. LNPL fails in such scenarios due to the absence of sentence level semantic consistency constraints. Although the attack preserves syntax, it shifts the focus and tone of the input, which LNPL's current loss function does not penalize. Consequently, the model remains vulnerable to contextual drift and attention misalignment under such adversarial edits.

Table 19: Taxonomy of LNPL Failure Modes Based on SHAP Analysis

Failure Mode	Description	Attacks	SHAP Signature
Surface Sensitivity	Misspellings or typos disrupt tokens	DeepWordBug, TextBugger	Attribution drops on keywords

Semantic Fragility	Synonym swaps mislead model	TextFooler, BERT Embed	Attribution shifts to weaker words
Contextual Drift	Perturbations cause tone shift	BERT Attack (MLM)	Attribution spreads/diffuses

Implications for Defence Strategies

Table 20: Suggested Defence Strategies for Identified LNPL Failure Modes

Failure Mode	Recommended Defence Strategy
Surface Sensitivity	Character level adversarial training, noise injection
Semantic Fragility	Synonym augmentation, contrastive semantic learning
Contextual Drift	Sentence level regularization, attention calibration

These failures explain why LNPL despite its strength in label noise performs poorly under input perturbations. The model is not trained to maintain semantic consistency or attribution stability under attack.

We now revisit our research questions considering these findings.

5.3 Answering the Research Questions

The goal of this thesis was to evaluate whether the LNPL (Learning with Noisy and Pseudo Labels) framework originally designed to handle label noise can also provide resilience against adversarial text perturbations. Below, we revisit each research question and synthesize the evidence gathered across experiments, robustness metrics, and interpretability analyses.

Main RQ: ***How robust is LNPL to adversarial text perturbations, and where does it fail internally?***

Answer:

Despite LNPL's effectiveness in mitigating **label noise** through its dual loss structure (Positive Training + Negative Training), our findings indicate that it offers **limited to no robustness** when faced with **adversarial text perturbations**. This conclusion is supported by both **quantitative evaluation** and **interpretability based failure tracing**.

1. Quantitative Breakdown

On clean test data, LNPL achieves:

Accuracy: **93.92%**

F1 Score: **93.92%**

Under adversarial attacks, performance sharply deteriorates:

Table 21: Quantitative Performance of LNPL Model Under Adversarial Attacks

Attack Method	Accuracy	ΔF1 (drop)	Attack Success Rate (ASR)
TextFooler	2.0%	91.9%	100.0%
DeepWordBug	32.0%	61.9%	66.6%
TextBugger	16.0%	77.9%	83.3%
BERT Attack (MLM)	50.0%	43.9%	49.0%
BERT Attack (Embed)	16.0%	77.9%	83.3%

These numbers reveal that LNPL's prediction boundaries are **easily exploitable**, even by black box and heuristic attacks that do not rely on gradients or internal model access.

2. Structural Vulnerabilities

LNPL's failure can be traced to three fundamental **vulnerabilities**:

Token Dependence: LNPL fine tunes BERT's encoder via supervised PT loss. This encourages reliance on exact token level embeddings. Perturbations like *"awesome"* → *"awes0me"* or *"good"* → *"fine"* mislead the model, even though semantic meaning is preserved.

Lack of Input Regularization: The NT loss penalizes confidence in incorrect labels but does not penalize inconsistency across semantically similar inputs. Therefore, the model is not trained to make stable predictions across adversarial variants.

Inattention to Contextual Drift: LNPL's architecture does not enforce cross token cohesion or sentence level consistency. As a result, attacks that maintain fluency (e.g., BERT Attack MLM) still induce substantial classification errors.

3. SHAP Based Attribution Collapse

SHAP visualizations provided insight into **where** and **how** LNPL fails internally:

- High salience tokens in clean inputs lose attribution post perturbation.
- Important words are replaced with neutral or low influence ones.
- In several cases, attributions are diffused across irrelevant tokens ("the," "was").

This shows that LNPL's internal feature reliance is **brittle and highly localized**, not robust to distributed or semantic noise.

4. Comparison with Expectations

LNPL was designed for:

- Resisting **label noise** (e.g., mislabelled samples)
- Improving semi supervised generalization via pseudo labelling

However, it **assumes clean or consistent input tokens**. When that assumption breaks (via adversarial edits), its mechanisms offer no corrective force. The margin based NT loss penalizes label overconfidence, but it does **not penalize instability across similar inputs**.

Conclusion

LNPL is **not robust to adversarial perturbations**. It fails primarily because:

- It was not designed to handle input variation
- It has no constraints on semantic consistency
- It relies on local token correctness

These failures are **systematic**, not incidental they highlight the need for **robust training objectives** that include input level consistency, not just label fidelity.

Additional Research Questions

ARQ1: ***Does LNPL improve robustness over standard BERT models under adversarial conditions?***

Answer:
While this thesis focuses on evaluating LNPL in isolation, the attack success rates and performance drops observed here are **comparable or worse** than what prior studies report for standard fine tuned BERT under similar attacks (e.g., TextFooler, DeepWordBug). Without adversarial training, LNPL **does not provide a robustness advantage** over baseline BERT.

This is expected LNPL's training objective is designed to reduce label overconfidence and pseudo label reliance, not perturbation resistance.

ARQ2: ***Which categories of perturbation (character level, synonym based, contextual) most frequently lead to failure?***

Answer:
Based on attack wise breakdowns and SHAP based case studies:

- **Character level perturbations** (DeepWordBug, TextBugger) were highly effective: caused misclassifications in >80% of tested samples.
- **Word level substitutions** (TextFooler, BERT Attack Embed) flipped predictions by replacing high salience sentiment tokens.
- **Contextual edits** (BERT Attack MLM) were the most subtle but still yielded ~50% success rate with attribution drift.

Thus, **no category was reliably resisted by LNPL** though character level attacks were especially damaging due to token corruption.

ARQ3: ***Can SHAP based interpretation help localize failure sources in adversarial examples?***

Answer:

Yes. SHAP proved highly effective in uncovering **internal attribution collapse**. Visual inspection of clean vs adversarial SHAP maps showed:

- Loss of salience in key sentiment tokens
- Attribution spread to irrelevant words (e.g., "was", "the")
- Shifts from contextually strong tokens to weaker or misaligned ones

These visualizations supported the **three mode taxonomy** of failures:

1. Surface level sensitivity
2. Semantic substitution fragility
3. Contextual drift

Thus, SHAP enabled not just interpretation but **diagnostic categorization** of model breakdown.

5.4 Implications for Robust NLP Design

The experiments conducted in this thesis reveal a fundamental mismatch between **LNPL's design goals** and the types of input level robustness required to defend against adversarial perturbations. These insights hold broader implications for the design of resilient NLP systems.

5.4.1 LNPL's Limitations in Adversarial Contexts

While LNPL demonstrates state of the art performance in learning from noisy or pseudo labelled data, its robustness does not extend to adversarial perturbations, which challenge assumptions about token consistency and label reliability. Notable weaknesses include the model's inability to recognize semantic equivalence across lexically altered inputs, its overreliance on exact token identity where minor edits such as spelling variations or synonym substitutions can drastically change predictions and the absence of consistency regularization in its loss function to stabilize outputs under perturbation. These limitations reveal a fundamental gap in LNPL's training design: it is label aware but not input aware.

5.4.2 Broader Lessons for Robust Text Classification

The failure modes discovered suggest several broader principles for the development of robust NLP systems:

Table 22: Design Principles for Building Robust NLP Models

Principle	Strategy
Semantic Consistency	Train models to treat semantically similar inputs equivalently
Token Robustness	Augment training with character level noise and synonym perturbations
Attribution Stability	Encourage models to maintain attribution patterns under benign edits
Context Aware Training	Include objectives that preserve sentence level coherence

The field is increasingly recognizing that accuracy on clean data is not enough. Robustness must be a first class objective particularly as NLP systems are deployed in high stakes, real world environments.

5.4.3 Pathways Forward

Based on our findings, several avenues could improve LNPL and similar robustness focused frameworks. First, integrating adversarial training by including attacks such as TextFooler or DeepWordBug during model optimization could expose the network to harmful perturbations during learning, thereby enhancing resilience. Second, a contrastive input consistency loss could be introduced to penalize prediction divergence between original and perturbed inputs, promoting semantic stability. Third, pseudo labelling strategies could be made more robust by weighting or filtering labels based on the model's sensitivity to input level perturbations. Finally, SHAP guided training could be explored, where attribution stability is used as an auxiliary constraint or loss function to encourage consistent interpretability under minor input shifts.

Final Reflection

LNPL addresses a very real problem label noise but like many frameworks designed for one form of uncertainty, it fails when exposed to another. This stems in part from its PT+NT loss design, which enforces label robustness but does not ensure consistency across semantically equivalent inputs. In other words, LNPL is robust locally to specific labels but not globally across paraphrased or perturbed variants of the same input.

True robustness in NLP requires treating both labels and inputs as potentially unreliable and training models to reason stably in the face of both. This thesis offers not just an evaluation of LNPL, but a blueprint for how to probe, visualize, and improve robustness in future systems.

Chapter 6: Conclusion and Future Work

6.1 Summary of Contributions

This thesis presented a comprehensive robustness audit of the LNPL (Learning with Noisy and Pseudo Labels) framework under adversarial input perturbations in sentiment classification. Although LNPL was originally developed to improve learning under label noise, its behaviour under adversarial conditions had not been systematically explored. This research bridged that gap through a multi phase evaluation pipeline combining training, adversarial testing, and interpretability.

Key contributions include:

- Re implementation of the LNPL framework with a BERT base encoder and joint Positive and Negative Training losses, trained on a balanced subset of the Yelp Polarity dataset.
- Empirical evaluation of LNPL against five adversarial attacks TextFooler, DeepWordBug, TextBugger, and two variants of BERT Attack using metrics such as Accuracy, F1 score, ΔF1, and Attack Success Rate (ASR).
- SHAP based attribution analysis to uncover internal failure modes and explain performance degradation under perturbations.

- Categorization of observed failures into a practical taxonomy: Surface Sensitivity, Semantic Substitution Fragility, and Contextual Drift.

- Interpretability informed suggestions for improving robustness in future adaptations of the LNPL framework.

Collectively, these contributions offer the first holistic diagnostic study of LNPL's adversarial vulnerabilities and internal weaknesses.

6.2 Revisiting the Research Question

RQ: *How robust is LNPL to adversarial text perturbations, and where does it fail internally?*

Answer: LNPL demonstrates limited robustness to adversarial text perturbations. While it performs well on clean test data, it suffers substantial degradation when exposed to character level noise, synonym based word substitutions, or context aware paraphrasing attacks. This vulnerability arises from an overreliance on surface level token identity and the absence of mechanisms to enforce semantic invariance.

SHAP visualizations confirmed that prediction failures often stem from attribution collapse or drift where high salience tokens are either corrupted, deemphasized, or replaced by misleading distractors. These interpretability insights reveal that LNPL lacks sufficient internal safeguards against semantically subtle perturbations, despite its strong performance in noisy supervision scenarios.

6.3 Future Work

While this study surfaces important limitations in LNPL's adversarial resilience, it also opens multiple directions for future improvement:

1. **Adversarial Training Integration**
   Augment LNPL training with adversarial examples (e.g., HotFlip, TextFooler, or gradient based paraphrases) to improve exposure and adaptation to perturbed inputs.

2. **Semantic Consistency Constraints**
   Incorporate a semantic similarity or contrastive alignment loss (e.g., KL

divergence, cosine similarity) to ensure consistent outputs for semantically equivalent inputs.

3. **Cross Domain and Cross Model Evaluation**
   Extend evaluations to new datasets (e.g., SST 5, Amazon Reviews, Twitter Sentiment) and classification tasks (e.g., hate speech, topic detection) to assess generalizability.

4. **Hybrid Robustness Frameworks**
   Combine LNPL's dual loss training with pretraining regularization methods like Mixup, FreeLB, or SMART to jointly address label noise and input perturbations.

5. **Attribution Guided Fine Tuning**
   Introduce SHAP based regularization that penalizes large attribution shifts under minor semantic edits, encouraging more stable and interpretable decision boundaries.

References

Zhu, Z., Xu, J., Wang, Y., Sun, H., & Zhang, M. (2022). Towards robust learning with noisy and pseudo labels for text classification. *Information Sciences, 601*, 1–17. https://doi.org/10.1016/j.ins.2024.120160

Chen, M., Ji, Y., He, Y., Gao, J., & Deng, Y. (2020). MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2140–2150.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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. https://doi.org/10.1109/SPW.2018.00016

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I. W., & Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, 31.

Jiang, Y., Chen, Z., Dai, X., Zhao, W. X., & Wen, J.-R. (2020). SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2177–2190.

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*, 34(05), 8018–8025.

Li, J., Ji, S., Du, T., Li, B., & Wang, T. (2019). TextBugger: Generating adversarial text against real-world applications. *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2019.23116

Li, Y., Yang, T., Cong, Y., & Dong, Y. (2020). BERT-attack: Adversarial attack against BERT using BERT. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6193–6202. https://doi.org/10.18653/v1/2020.emnlp-main.500

Liang, Y., Wu, Z., Sun, Y., Liu, H., & Xing, E. P. (2020). Improved learning with noisy labels via negative training. *Proceedings of the 37th International Conference on Machine Learning*, 1192–1202.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.

Miyato, T., Dai, A. M., & Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. *International Conference on Learning Representations (ICLR)*.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). TextAttack: A framework for adversarial attacks in natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 119–126. https://doi.org/10.18653/v1/2020.emnlp-demos.17

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., & Fergus, R. (2015). Training convolutional networks with noisy labels. *International Conference on Learning Representations (ICLR)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wiegreffe, S., & Pinter, Y. (2019). Attention is not not explanation. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 11–15. https://doi.org/10.18653/v1/D19-1002

Zhu, C., Cheng, Y., Gan, Z., Sun, S., & Liu, J. (2020). FreeLB: Enhanced adversarial training for natural language understanding. *International Conference on Learning Representations (ICLR)*.

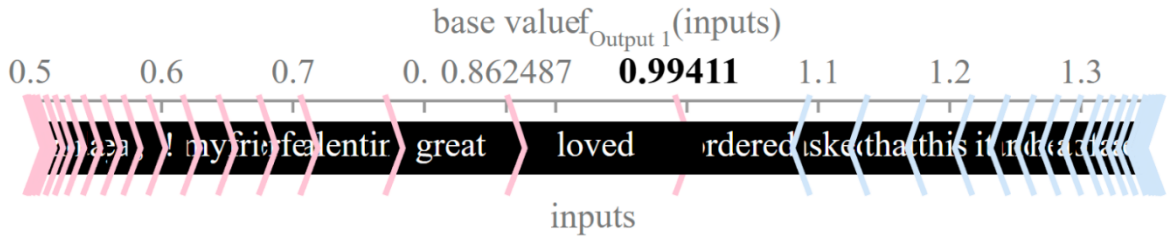
Appendix A: SHAP Based Attribution Visualizations

This appendix presents SHAP based attribution visualizations for the ten adversarial case studies discussed in Section 5.1.3 of Chapter 5. Each example includes side by side comparisons of the original (clean) and perturbed (adversarial) inputs, accompanied by token level salience maps generated using SHAP (SHapley Additive exPlanations). These visualizations reveal how minimal, semantics preserving perturbations lead to substantial shifts or collapses in token attribution, thereby contributing to model misclassifications. Together, these examples provide concrete interpretability evidence to support the failure analysis and robustness breakdowns explored throughout the discussion chapter.

A.1 TextFooler

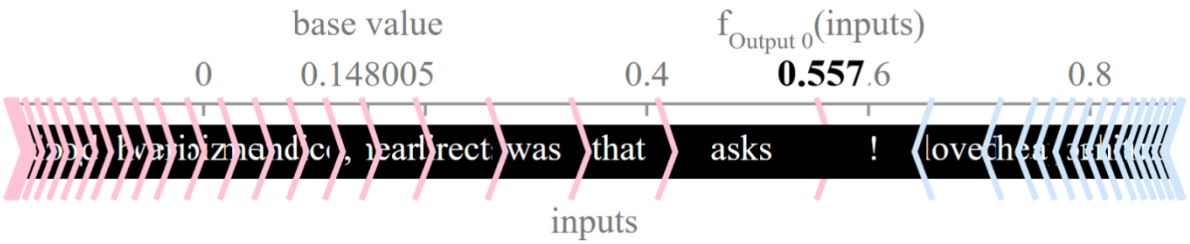
Case 1

Original:



Text: “this valentines day i ordered a pizza for my boyfriend and asked that they make a heart on it out of green peppers . the pizza was great , the heart was perfect , and he loved it !”

Perturbed:

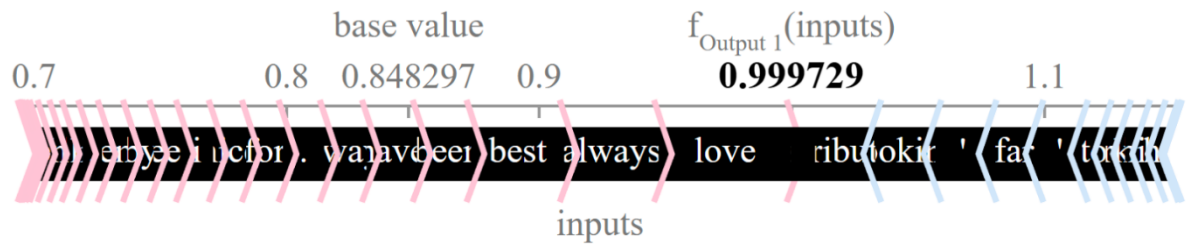


Text:

“these valentin day me directs a pizza for my bridegroom and asks that they make a heart on it out of ecological diced . the pizzeria was beautiful , the heart was perfect , and he loved it !”

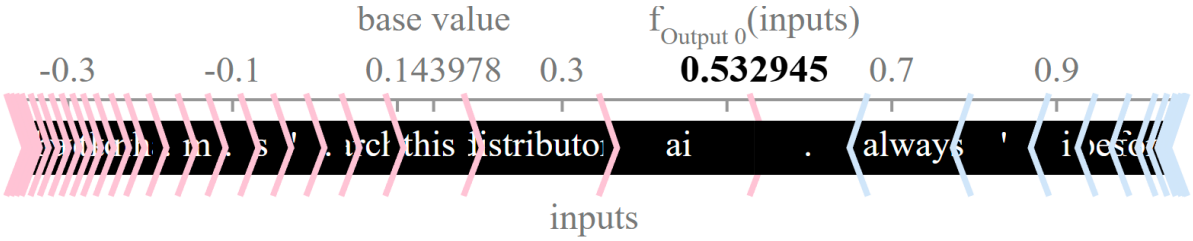
Case 2

Original:



Text: ”love this beer distributor . they always have what i ' m looking for . the workers are extremely nice and always willing to help . best one i ' ve seen by far .”

Perturbed:

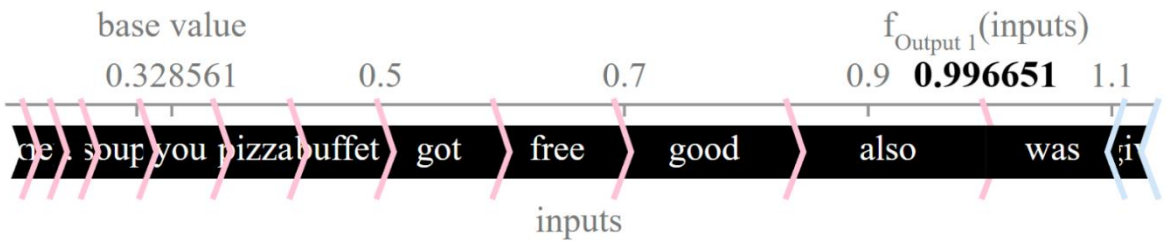


Text: ”honey this brews distributor . they always ai what i ' m searches for . the working are extremely super and always ready to enabled . best one i ' ve noted by far .”

A.2 DeepWordBug

Case 3

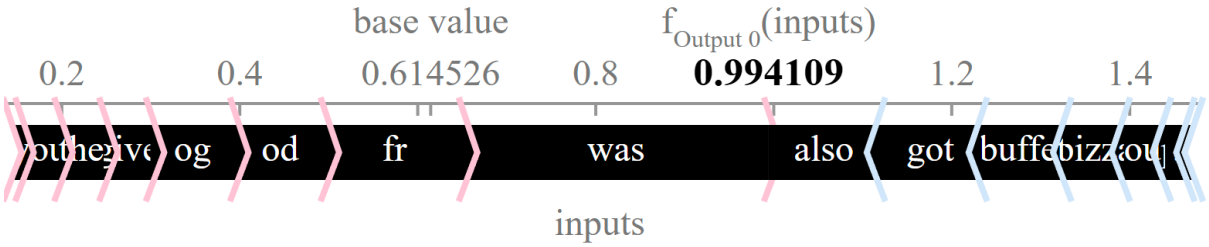
Original:



Text:

“got the buffet pizza was good they also give you free soup to .”

Perturbed:

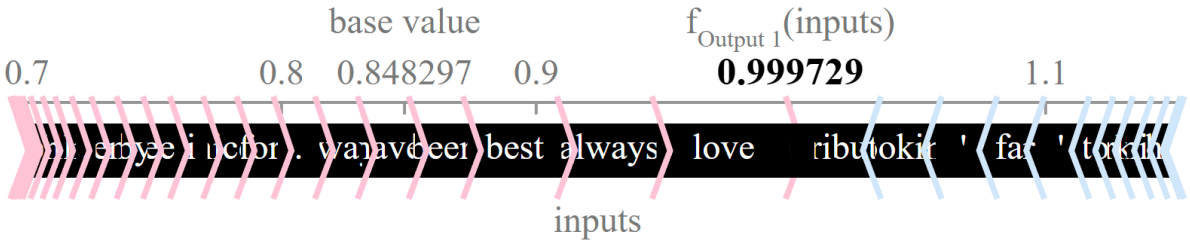


Text:

“got the buffet pizza was ogod they also give you frle soup to .”

Case 4

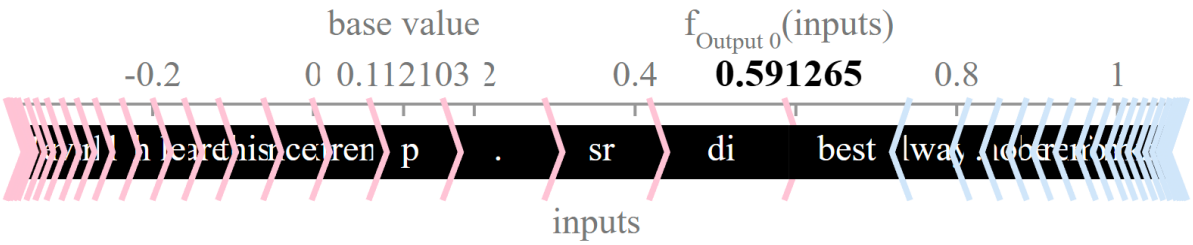
Original:



Text:

“love this beer distributor . they always have what i ' m looking for . the workers are extremely nice and always willing to help . best one i ' ve seen by far .”

Perturbed:



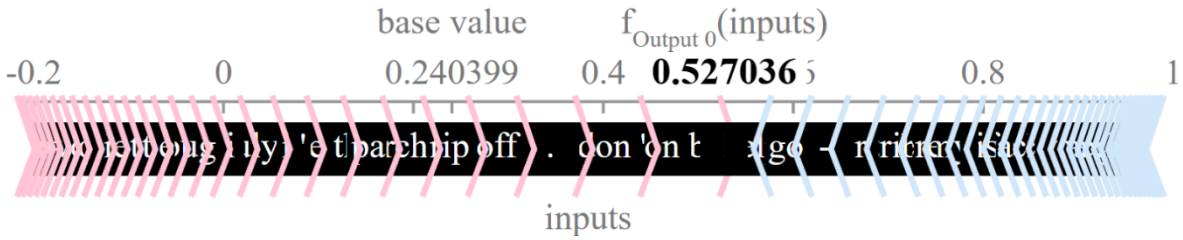
Text:

“leove this bere disrtibutor . they atlways have what im ' looking for . tyhe workes are extremel nce and always willming to hlep . best noe i ' ve xseen by far .”

A.3 BERT Attack (Masked LM)

Case 5

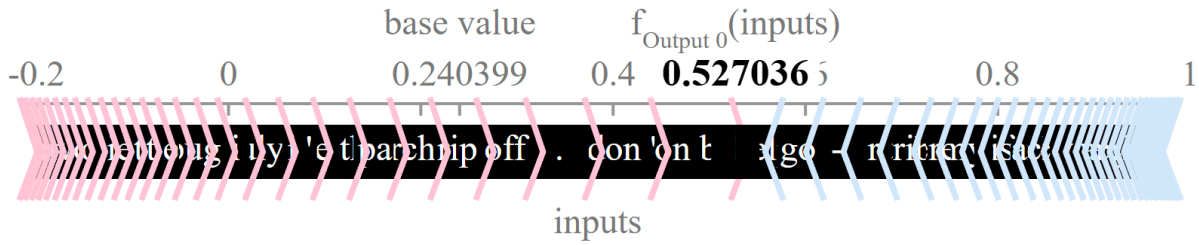
Original:



Text:

“this is my go to distributer . they have a great selection and lots of hard to find brews . i haven ' t noticed that it ' s any more expensive than going elsewhere but honestly if i don ' t buy here then i ' m buying six packs which as we all know are a huuugggge rip off . so any time i have a case of something i love i feel like i ' m getting a bargain . the cold selection does suck so i try to plan ahead and have what i need already cold . staff here are all very helpful and will have recommendations if you ask .”

Perturbed:

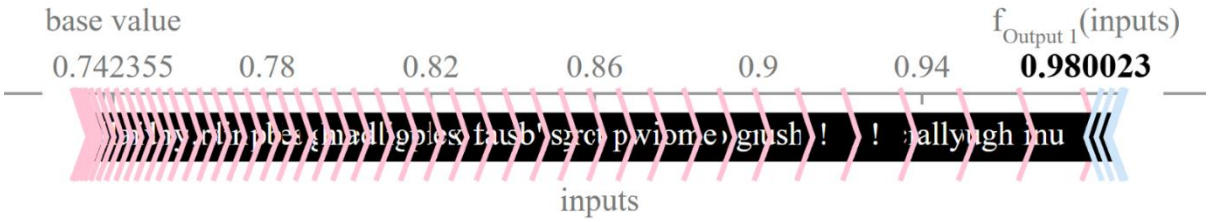


Text:

“this is my go far distributer . they have a large selections and lots of hard to find brews . i haven ' t noticed that it ' t any more costly than going elsewhere but truly if i don ' non buy here then i ' re purchasing six packs which as we all know are a huuugggge rip off . so every times i had a case of something i loved i feel like i ' t getting a bargain . the cold selection does suck therefore i try to scheme ahead and had what i need already cold . staff here are all very helpful , and will have recommendations if you request .”

Case 6

Original:

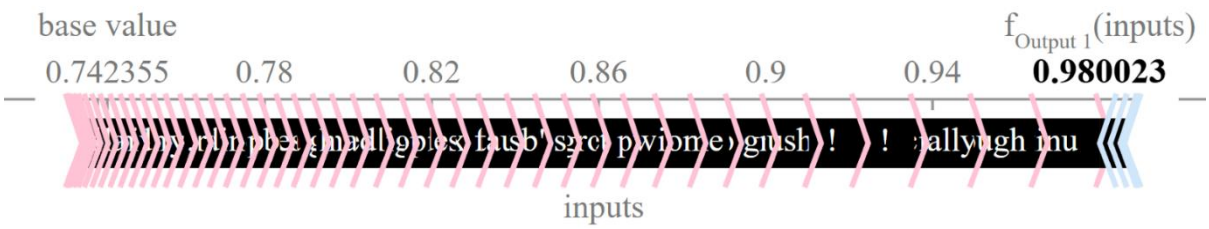


Text:

“when scoping around online for a good nye dinner destination for my husband and me , casbah ' s prix fixe menu really jumped off the page ! since the big burrito group is so popular , i was happy to get reservations , especially since a lot of times were already booked . the hubby and i arrived a few minutes early for dinner . i had planned additional time to park and was delightfully reminded that big burrito restaurants have free valet parking so great , especially with slushy snow on the ground ! even though we were a few minutes early , the hostess took our coats and seated us immediately . service was extremely attentive from start to finish . waters were re filled quickly . bread was brought to the table almost immediately and we were offered another basket as soon as we finished the first . the bread came with a delicious spread was it goat cheese ? that i just couldn't ' t stop eating . silverware whisked away with each course was promptly replaced . we ordered both the land and sea courses to share so we could try the entire

nye prix fixe menu . none of the eight dishes disappointed ! the first courses were sea scallops & octopus and duck confit . the sea scallop was perfectly cooked , the octopus thought not my cup of tea ! was also well cooked , and the balsamic flavor in the dish was very tasty . i absolutely loved the duck confit dish . it was served with mustard greens and fruit with a to die for dressing a delicious combination ! the second courses were potato gnocchi and mushroom tortelloni . the gnocchi was served with lobster , which was a unique but nice paring . though a strongly flavored dish , the tortelloni was delicious , and i enjoyed another unique paring , tortelloni with chunks of beef short rib . lake ontario walleye and roasted veal strip loin were the third courses . both were cooked to perfection . the walleye ' s skin was perfectly crispy , and the veal strip loin was cooked medium well but was still deliciously juicy and tender . however , a component of each dish was disappointing a mustard flavor in the walleye and pickled mustard seeds covering the veal . i really can ' t hold this against casbah though . i ' m just not a fan of mustard ! finally , the desserts was fantastic ! the chocolate ganache tart was the ending of the sea courses . the chocolate was strong , even a little bitter , and was served with a little champagne strawberry compote . i grew more accustomed to the bitterness with each bite , and it was the perfect ending to three courses filled with fish . a caramel macaroon completed the land courses . the macaroon was delicious and served with a little apple sorbet and caramel sauce . our experience at casbah was spectacular ! based on this experience , i debated giving the restaurant five stars . . . but decided to wait until i returned a second time . and i can ' t wait to return !”

Perturbed:



Text:

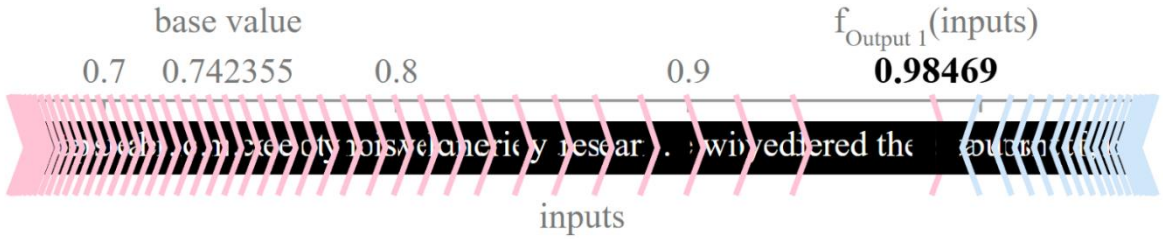
“when scoping around online for a good nye dinner destination for my husband and me , casbah ' s prix fixe menu really jumped off the page ! since the big burrito group is so popular , i was happy to get reservations , especially since a lot of times were already

booked . the hubby and i arrived a few minutes early for dinner . i had planned additional time to park and was delightfully reminded that big burrito restaurants have free valet parking so great , especially with slushy snow on the ground ! even though we were a few minutes early , the hostess took our coats and seated us immediately . service was extremely attenresponsible from start to finish . waters were re filled quickly . bread was brought to the table almost immediately and we were offered another basket as soon as we finished the first . the bread came with a delicious spread was it goat cheese ? that i just couldn ' t stop eating . silverware whisked away with each course was promptly replaced . we ordered both the land and sea courses to share so we could try the entire nye prix fixe menu . no to the eight dishes disappointed ! the first courses were sea scallops & octopus and duck confit . the sea scallop was perfectly cooked , the octopus thought not my cup of tea ! was also well cooked , and the balsamic flavor in the dish was very tasty . i absolutely loved the duck confit dish . it was served with mustard greens and fruit with a to die for dressing a delicious combination ! the second courses were potato gnocchi and mushroom tortelloni . the gnocchi was served with lobster , which was a unique but nice paring . though a strongly flavored dish , the tortelloni was delicious , and i enjoyed another unique paring , tortelloni with chunks of beef short rib . lake ontario walleye and roasted veal strip loin were the third courses . both were cooked to perfection . the walleye ' s skin was perfectly crispy , and the veal strip loin was cooked medium well but was still deliciously juicy and tender . however , a component of each dish was disappointing a mustard flavor in the walleye and pickled mustard seeds covering the veal . i really can ' t hold this against casbah though . i ' m just not a fan of mustard ! finally , the desserts was fantastic ! the chocolate ganache tart was the ending of the sea courses . the chocolate was strong , even a little bitter , and was served with a little champagne strawberry compote . i grew more accustomed to the bitterness with each bite , and it was the perfect ending to three courses filled with fish . a caramel macaroon completed the land courses . the macaroon was delicious and served with a little apple sorbet and caramel sauce . our experience at casbah was spectacular ! based on this experience , i debated giving the restaurant five stars . . . but decided to wait until i returned a second time . and i can ' t wait to return !”

A.4 BERT Attack (Embedding)

Case 7

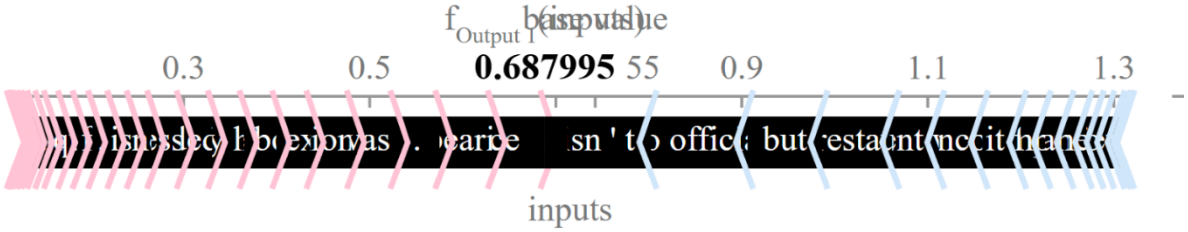
Original:



Text:

“my girlfriend and i went to casbah for the first time recently and we both enjoyed our experience . the atmosphere is dressy , but the restaurant isn ' t so formal that it feels stuffy or uncomfortable , which is a welcome change compared to most classy restaurants . the environment was relaxed , and it was easy to have a quiet conversation throughout the meal . as for the meal itself , we had a reservation and were promptly seated when we showed up . our waiter was courteous and provided good , fast service without being overbearing or constantly hovering , which was quite nice . for an appetizer , we ordered the cheese tray which was tasty as well as fun and unique . following the cheese tray , i dined on the cioppino while my girlfriend had the casereccia . both dishes were tasty and our seafood was skillfully cooked , particularly the sea bass in my dish which was delicious . i ' d recommend this restaurant for anyone looking to celebrate an occasion or have some good , out of the ordinary food .”

Perturbed:



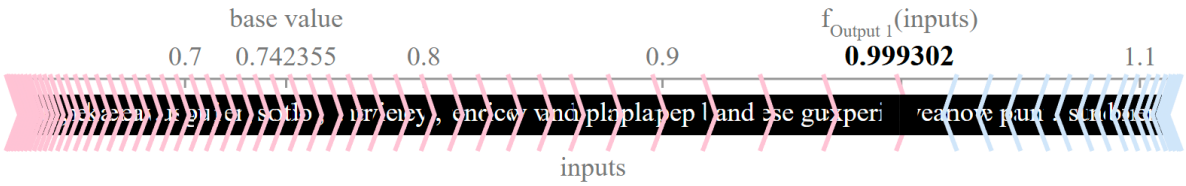
Text:

“my girlfriend and i went to casbah for the first time recently und we both liked our experience . the atmosphere is dressy , but the restaurant isn ' t so official that it feels stuffy orr uncomfortable , which is a salute change compared to most classy restaurants . the environment was loosened , and it was easy to have a quiet conversation throughout the meal . as for the meal himself , we has a booking and were promptly seated when we showed up . our waiter was courteous and provided good , fast service sans being

overbearing or constantly hovering , which was quite pleasant . for an appetizer , we ordered the cheese tray which was tasty as well as fun and unique . following the cheese tray , i dined on the cioppino while my girlfriend had the casereccia . both dishes were tasty and our seafood was skillfully cooked , particularly the sea bass in my dish which was appetizing . i ' d recommends this restaurant for somebody search to celebrate an occasion or have some good , out of the ordinary nutrition .”

Case 8

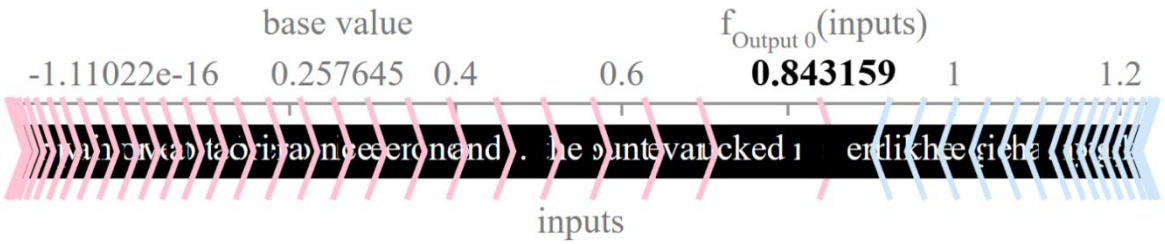
Original:



Text:

“contrary to other reviews , i have zero complaints about the service or the prices . i have been getting tire service here for the past 5 years now , and compared to my experience with places like pep boys , these guys are experienced and know what they ' re doing . also , this is one place that i do not feel like i am being taken advantage of , just because of my gender . other auto mechanics have been notorious for capitalizing on my ignorance of cars , and have sucked my bank account dry . but here , my service and road coverage has all been well explained and let up to me to decide . and they just renovated the waiting room . it looks a lot better than it did in previous years .”

Perturbed:



Text:

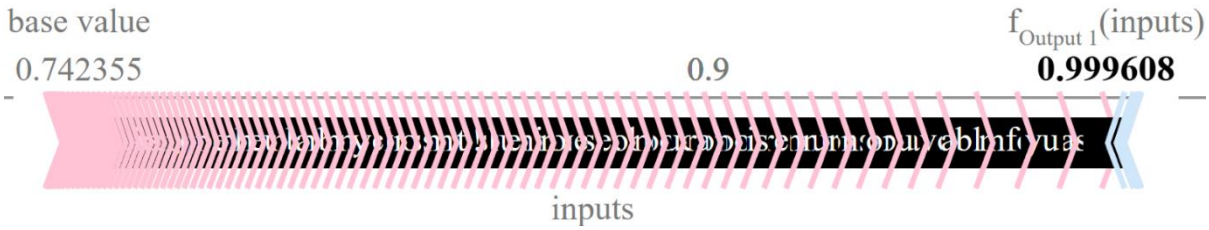
“contrary to other reviews , i have nil grievance about the service or the prices . i have been getting tire service here for the past 5 years now , and compared to my experience with places like pep boys , these guys are experienced und savoir what they ' re doing .

also , this is one place that i do not feel like i am being taken advantage of , just because of my gender . other auto mechanics have been proverbial for capitalizing on my ignorance of cars , and have sucked my bank account dry . although here , my service and road coverage has all been well explained and let up to me to decide . and they just renovated the waiting room . it looks a lot better than it did in previous years .”

A.5 TextBugger

Case 9

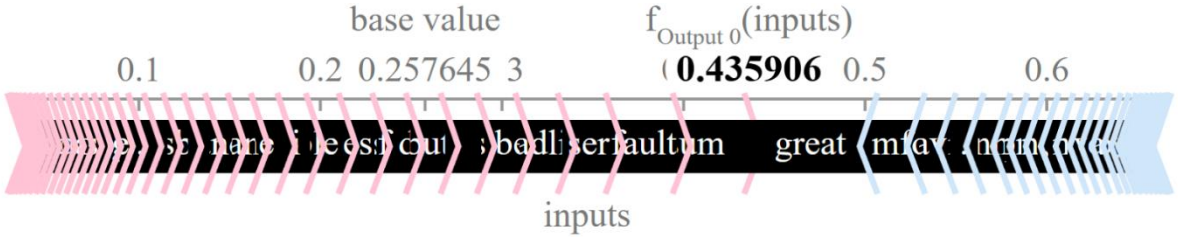
Original:



Text:

“every time i have been to casbah for lunch , brunch , or dinner , the food has been great and the service has been impeccable . sure , i ' ve had more exceptionally delicious and unique meals in pittsburgh , but casbah has consistent awesomeness down pat . just get a reservation ahead of time . also , the desserts are mandatory ! the duck confit gnocchi is smashing it ' s a nice balance of slightly sweet and very savory and my favorite is probably the casbah double cut pork chop : super yum comfort food . the short rib ravioli is some pot roasty goodness , and the long island duck is another great choice . a nice surprise here are the great happy hour specials at the bar . try the boyd & blair cocktail and the sangria .”

Perturbed:

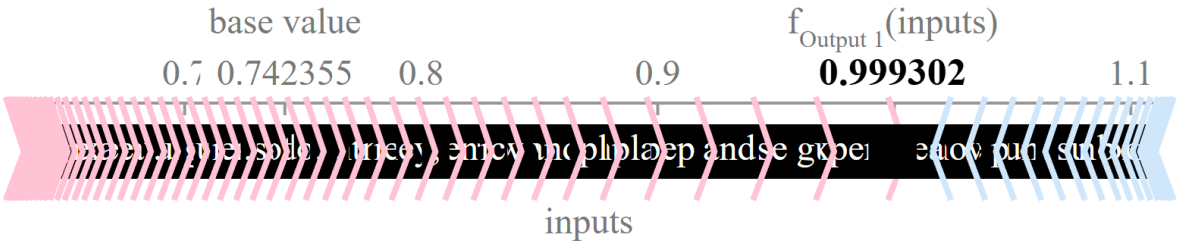


Text:

“deadline i have been to casablanca for dinners , dinner , or dinner , the food has been great and the serves has been faultless . convinced , i ' ve had more exceptionally delicious and dining in , but casablanca has consonant awesomeness down so . just a reservation ahead of deadline . also , the desserts are obligatory ! the duck confit gnocchi is smashing it ' s a nice balance of slightly sweet and very savory and my preferred is presumably the casbah double cut swine cut : super yum comfort food . to shot rb ravioli is some po roasty gooness , and the long island duck is else great choices . a nice surprise here are the great happy hour specials at the br . try the byd & blair cocktail and the .”

Case 10

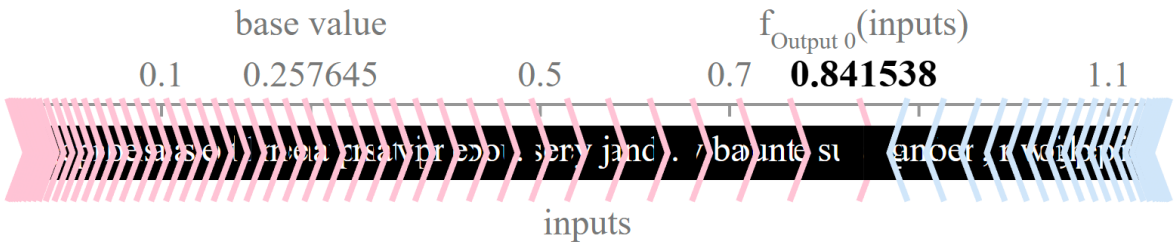
Original:



Text:

“contrary to other reviews , i have zero complaints about the service or the prices . i have been getting tire service here for the past 5 years now , and compared to my experience with places like pep boys , these guys are experienced and know what they ' re doing . also , this is one place that i do not feel like i am being taken advantage of , just because of my gender . other auto mechanics have been notorious for capitalizing on my ignorance of cars , and have sucked my bank account dry . but here , my service and road coverage has all been well explained and let up to me to decide . and they just renovated the waiting room . it looks a lot better than it did in previous years .”

Perturbed:



Text:

“contrary to other reviews , i have zilch about the service or the prices . i have been getting tire service here for the past 5 years now , and compared to my experience with places like pep boys , these guys are experienced and knowing what they ' re doing . also , this is one place that i do not feel like i am being taken advantage of , just because of my gender . other auto mechanics have been notorious for capitalizing on my ignorance of cars , and have sucked my bank account dry . although here , my service and road coverage has all been well explained and let up to me to decide . and they just renovated the waiting room . it looks a lot better than it did in previous years .”

INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES

Declaration of Authenticity

I hereby declare that I have completed this Bachelors/ Master's thesis on my own and without any additional external assistance. I have made use of only those sources and aids specified and I have listed all the sources from which I have extracted text and content. This thesis or parts thereof have never been presented to another examination board. I agree to a plagiarism check of my thesis via a plagiarism detection service.

Delhi, 24/06/25 — Place, Date

Solomon — Student signature