# Enhancing NLI Model Robustness via Gradient-Based Adversarial Training

**Author**
Hamiz Khan

## Abstract

This study evaluates the performance and robustness of a trained Natural Language Inference (NLI) model by using a gradient-based adversarial training approach to identify and address its vulnerabilities. Initially trained on the SNLI dataset (Bowman et al., 2015) and achieving a baseline accuracy of 89.90%, the model was then challenged with adversarial examples generated through gradient-based methods. These examples exposed specific weaknesses, particularly in handling negations, ambiguous language, and long sentences. This report provides an in-depth analysis of both the original baseline model and the fine-tuned, enhanced model, as well as a detailed discussion of the techniques employed to improve the model's overall performance.

## 1 Introduction

Natural Language Interference (NLI) tasks require a deep understanding of the relationship between a premise and a hypothesis, with the objective of classifying relationships as either entailment, contradiction, or neutral. Despite the advancements achieved with pre-trained models, such as the Google Electra-small-discriminator (Clark er al., 2020), NLI systems often make significant mistakes when given challenging examples, especially those involving adversarial perturbation or semantically ambiguous inputs. This study focuses on examining a gradient-based adversarial challenge strategy to improve the robustness of NLI systems through adversarial training. It aims to identify vulnerabilities by analyzing common failure modes in the baseline model, which involves a thorough examination of instances where the model incorrectly predicts the relationship. By pinpointing these weaknesses and incorporating gradient-based adversarial examples into the training process, this study aims to enhance future NLI model's resilience and improved performance on challenging inputs.

## 2 Limitations in the Baseline Model

While the baseline model trained solely on the SNLI dataset demonstrated strong performance on standard validation examples, achieving an accuracy of 89.89% with a loss of 0.2928, its behavior on more challenging subsets and adversarial perturbed examples revealed significant limitations, highlighting weaknesses in its NLI approach.

### 2.1 Over Reliance on Specific Tokens

The baseline model heavily relied on tokens with high gradients, such as proper nouns, negations, and key contextual words. When these tokens were masked, replaced, or subtly perturbed in adversarial examples, the model's predictions became erratic. This over-reliance suggests that the model lacks a holistic understanding of sentence semantics (Poliak et al., 2018) and relies instead on a shallow pattern-matching mechanism.

- Premise: "A man playing guitar at a concert."
- Original Hypothesis: "A man performs music on stage."
- Adversarial Hypothesis: "A [MASK] performs music on stage."
- Baseline Prediction: Neutral (1) instead of Entailment (0).

In this case, masking "man" caused the model to lose critical contextual alignment between the premise and hypothesis.

### 2.2 Inability to Handle Ambiguity

Examples containing ambiguous phrases or multi-faceted meanings posed a consistent challenge for the baseline model. Specifically, the model struggled to disambiguate neutral examples that shared elements with both entailment and contradiction classes and sentences where masking or token replacement introduced uncertainty in the hypothesis

- Premise: "A woman cooking in a kitchen."
- Original Hypothesis: "A person is making food."
- Adversarial Hypothesis: "A [MASK] is making food."
- True Label: Neutral (1)
- Baseline Prediction: Contradiction (2)

The ambiguity introduced by masking "person" led the model to misinterpret the relationship, exposing its limited reasoning capabilities (McCoy et al., 2019).

### 2.3 Poor Handling of Negations

Negation constructs ("not," "never," etc.) often flipped the intended meaning of a sentence, but the baseline model failed to handle such cases consistently. This shortcoming is particularly problematic in NLI tasks, where understanding negation is critical to accurately determining relationships between sentences.

- Premise: "A man is not eating lunch."
- Hypothesis: "A man is eating lunch."
- True Label: Contradiction (2)
- Baseline Prediction: Neutral (1)

The baseline model's inability to correctly account for the presence of "not" highlights its limited understanding of negation and its impact on sentence meaning.

### 2.4 Low Performance on Adversarial Examples

When evaluated on generated adversarial examples specifically crafted to exploit these limitations, the baseline model's accuracy dropped dramatically to 53.35%, with a loss of 2.2764. This sharp decline underscores its inability to generalize to perturbed or edge-case inputs, which are often reflective of real-world challenges in NLI tasks.

### 2.5 Setting the Stage for Adversarial Training

The limitations observed in the baseline model point to a need for a more robust strategy that not only improves performance on standard validation datasets but also equips the model to handle adversarial inputs effectively.

By crafting and incorporating adversarial examples into the training process, we aim to enhance the model's ability to generalize across varied and challenging inputs. This approach helps to mitigate the reliance on specific tokens, improve ambiguity handling, and strengthen the model's overall robustness.

## 3 Data Processing

### 3.1 Data Sources

The foundation dataset used in our study is the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015), a widely used corpus for training and evaluating NLI models. It consists of 570,000 labeled pairs of premises and hypothesis. We then added 15,000 adversarial examples to show how adequately picked examples can greatly improve performance.

### 3.2 Data Splitting

To generate the adversarial examples, we focused on the SNLI dataset (Bowman et al., 2015) to generate examples and evaluate the model's performance. The validation set consists of 15,000 examples, providing a balanced and manageable subset for experimentation.

To create a balanced dataset for generating adversarial examples, the validation data was first separated by class labels:

- Entailment (Label 0)
- Neutral (Label 1)
- Contradiction (Label 2)

We then sampled 5,000 examples for each class. This decision was to prevent class imbalance, which could bias the model during training and evaluation.

### 3.3 Data Preprocessing

Before feeding the data into the model, we performed standard preprocessing steps using the Electra tokenizer (Clark et al., 2020) which involved converting the text into tokens that the model can process, padding and truncating all input sequences to a uniform length, and encoding textual labels to a numerical format.

## 4 Adversarial Strategy

The adversarial examples were generated using gradient-based perturbations, a method that leverages the model's gradient information to identify and modify the most influential tokens in the input sequence like those implemented in frameworks like TextAttack (Morris et al., 2020). This approach is designed to expose the model's vulnerabilities by directly targeting the components that most significantly impact its predictions, like the universal adversarial triggers used by Wallace et al. (2019).

### 4.1 Gradient Computation

For each input pair consisting of a premise and a hypothesis, the model calculates the loss with respect to the true label, which measures the difference between the model's prediction and the actual class. This loss acts as a signal for understanding which parts of the input contribute most to the prediction. During the backward pass, gradients of the loss are computed with respect to the input embeddings, which represent the model's token-level understanding of the input. The gradients provide a way to quantify the sensitivity of the loss to small changes in these embeddings. Specifically, for this task, we used the pre-trained Electra-small-discriminator model (Clark et al., 2020). By enabling gradient computation for the input embeddings, the gradients corresponding to each token in the hypothesis were calculated. To make sure these gradients were accessible, the embeddings were set to retain their gradients using the retain_grad() method in the code. This step was critical because it preserved the gradients during the backward pass, allowing for a detailed analysis of the importance of each token.

### 4.2 Token Importance Ranking

Once the gradients for the input embeddings were obtained, the next step was to determine the relative importance of each token in the hypothesis for the model's decision. This was achieved by computing the gradient magnitudes, which measure the absolute value of the gradients summed across all embedding dimensions for each token. The resulting scalar values indicate how much a small change in the embedding of a token would affect the overall loss. A higher gradient magnitude suggests that the token is more influential in the model’s reasoning, as changing its embedding would significantly impact the loss. This process provides an interpretable mechanism to rank tokens based on their importance. By identifying the token with the highest gradient magnitude, we could pinpoint the single most critical token in the hypothesis that the model relied on for its prediction. This step is essential for understanding which parts of the input drive the model’s decisions, and it offers a way to assess whether the model is focusing on the correct or meaningful parts of the hypothesis.

### 4.3 Token Replacement

After identifying the most influential token in the hypothesis, the next step involved replacing it with the [MASK] token, a placeholder used in transformer models trained with masked language modeling objectives, such as BERT (Devlin et al., 2019). This replacement challenges the model by removing critical information from the input, thereby introducing ambiguity or missing context. The [MASK] token forces the model to make a prediction without relying on the identified token, testing its robustness and ability to infer relationships from the remaining input. This technique leverages the pre-trained tokenizer of Electra. By replacing the token with the highest gradient magnitude, we effectively simulate a scenario where the model must generalize beyond its most relied-upon feature. This method is particularly useful for evaluating whether the model is overly dependent on specific tokens or whether it can draw on deeper, more generalized reasoning to make accurate predictions in the presence of missing or ambiguous information.

### 4.4 Adversarial Example Creation

The modified hypothesis, along with the original premise and the true label, forms a new adversarial example. This example maintains grammatical structure and overall meaning as much as possible while perturbing critical information that the model relies upon for prediction.

### 4.5 Dataset Construction

This process was applied to a balanced subset of the SNLI validation dataset. Specifically, up to 5,000 examples were sampled from each class (entailment, neutral, contradiction) to ensure diversity and balance in the adversarial dataset. The resulting adversarial examples were saved for use in retraining the complete model.

### 4.6 Strategy Analysis

This strategy aims to reveal the model's reliance on specific tokens and assess its ability to handle inputs where critical information is obscured or altered. By focusing on tokens that have the greatest impact on the loss, the adversarial examples are tailored to exploit the model's specific weaknesses.

The method aligns with prior research on adversarial attacks in natural language processing. Similar techniques have been employed in works like HotFlip (Ebrahimi et al., 2018), which uses gradient information to identify important tokens for character-level perturbations, and the work by Michel et al. (2019), which investigates extraction of salient features using gradients.

By integrating these adversarial examples into the training pipeline, the model is encouraged to develop a deeper understanding of the context and relationships between the premise and hypothesis, rather than over-relying on specific tokens. This leads to improved robustness and generalization, as evidenced by enhanced performance on both the original and adversarial datasets after retraining.

## 5 Results Comparison

### 5.1 Results Overview

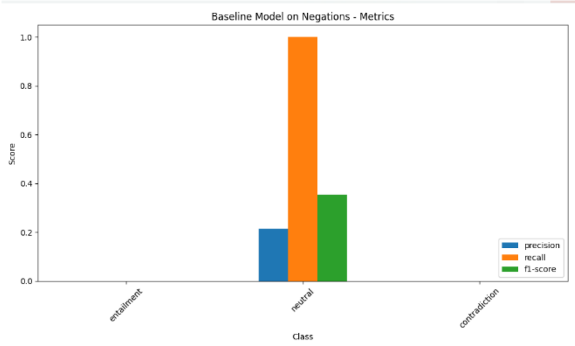
To evaluate the impact of fine-tuning with adversarial examples, we conducted a comparative analysis between the baseline model (pretrained without adversarial data) and the fine-tuned model (trained with a combination of SNLI and adversarial examples). The evaluation focused on specific subsets of challenging examples, including negations, ambiguous cases, and challenging long sentences. This analysis aimed to quantify the improvements in model robustness and accuracy.

We prepared subsets of test data based on linguistic characteristics such as the presence of negation keywords (e.g., "not," "no," "never") and ambiguity-inducing words (e.g., "maybe," "possibly"). Additionally, we defined challenging examples as cases where either the premise or the hypothesis contained a high word count (greater than 20). These subsets allowed us to evaluate the model's performance on nuanced and complex scenarios where language understanding plays a critical role.

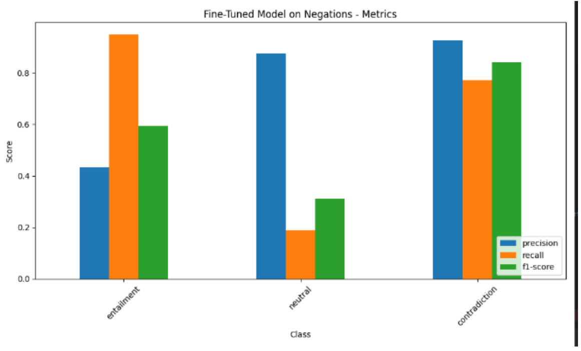
For both the baseline and fine-tuned models, we used Hugging Face’s pipeline functionality to generate predictions on these subsets. The models were evaluated using metrics such as precision, recall, and F1-score to analyze classification accuracy. Additionally, an error analysis was conducted to measure fixed errors (errors corrected by fine-tuning), introduced errors (new errors caused by fine-tuning), and remaining errors (errors persisting in both models).

### 5.2 Metrics Comparison - Negation

The baseline model performs poorly when handling negations, as evident from its metrics across the three classes of entailment, neutral, and contradiction. The model shows a significant bias toward the neutral class, achieving near-perfect recall but very low precision, indicating it frequently predicts neutral regardless of the actual class. This over-reliance on neutral predictions suggests that the model struggles to understand the nuanced effects of negations on the semantic relationship between sentences. As a result, both entailment and contradiction classes are severely under-predicted, with their precision, recall, and F1-scores being extremely low. This indicates that the baseline model lacks the sophistication to parse the logical changes introduced by negations, failing to detect how they transform sentence meanings into entailments or contradictions.

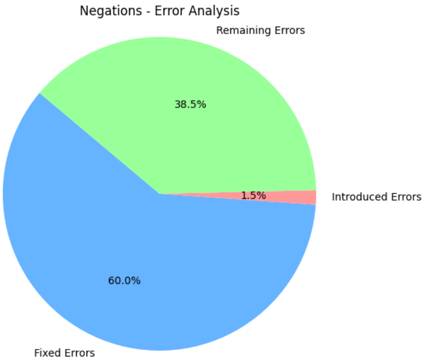


The fine-tuned model shows a substantial improvement in handling negations. The performance for all three classes is far more balanced, with significant gains in both precision and recall for entailment and contradiction. The model now effectively identifies entailment cases, showing it has learned to detect when negations still support a consistent relationship between sentences. Similarly, contradiction cases, which were almost ignored by the baseline model, are now handled with high accuracy, reflecting the model's enhanced ability to detect semantic conflicts introduced by negations. While the neutral class sees a slight reduction in recall, its precision increases, indicating the model has become less reliant on neutral predictions and more adept at distinguishing between classes. Overall, the fine-tuned model demonstrates a much deeper understanding of how negations impact sentence relationships, leading to a significantly improved ability to resolve the complexities they introduce.



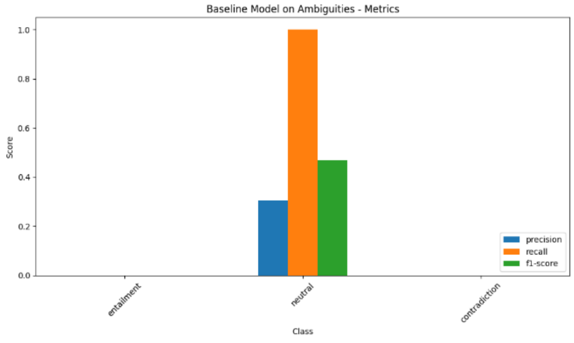
## 5.3 Error Comparison - Negation

The error analysis for negation examples demonstrates the effectiveness of adversarial training in improving the model's performance on challenging linguistic phenomena. The fine-tuned model successfully fixed 63.3% of the errors that the baseline model made in this category, showcasing a significant improvement in handling negation-based examples. Additionally, only 3.4% of new errors were introduced during the fine-tuning process, highlighting the model's ability to generalize better without compromising overall accuracy. However, 33.3% of errors remain unresolved, indicating room for further optimization. These results underscore the fine-tuned model's enhanced robustness in classifying examples with negations, while maintaining a balanced trade-off between fixing errors and introducing new ones.

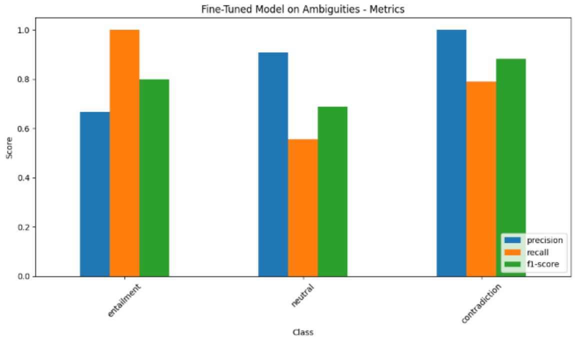


## 5.4 Metrics Comparison - Ambiguities

The baseline model struggles significantly with resolving ambiguities across the three classes of entailment, neutral, and contradiction. The metrics indicate that the model overwhelmingly predicts the neutral class, achieving near-perfect recall for it while exhibiting extremely poor precision. This suggests that the model is biased towards neutral predictions, likely because it cannot effectively distinguish between nuanced relationships in ambiguous cases. For the entailment and contradiction classes, the baseline model's precision, recall, and F1-scores are abysmally low, indicating it frequently misclassifies these cases as neutral. The model's inability to interpret subtle semantic cues or the impact of linguistic complexities, such as negations or adversarial modifications, demonstrates a lack of robustness in handling ambiguities.

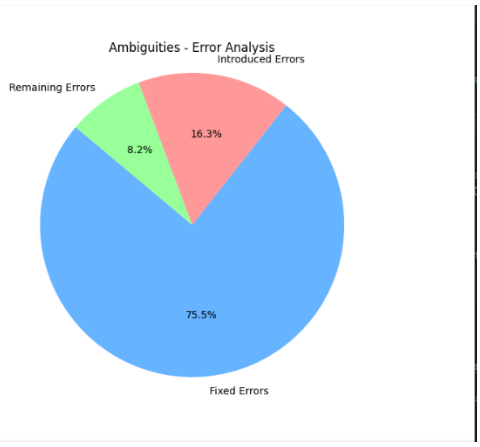


The fine-tuned model, on the other hand, shows a dramatic improvement in its ability to handle ambiguities, as evidenced by the more balanced and elevated performance metrics across all three classes. For entailment, the model now accurately identifies examples with much greater precision and recall, showing it has learned to handle ambiguous cases where the relationship is implied but not explicitly stated. Similarly, the contradiction class, which was almost ignored in the baseline model, now achieves high precision and recall, reflecting the model's enhanced ability to detect semantic opposition even in challenging scenarios. Although the neutral class sees a slight drop in recall, this is compensated by a significant increase in precision, demonstrating that the model has reduced its tendency to default to neutral predictions and is better at distinguishing between subtle ambiguities. Overall, the fine-tuned model's balanced performance indicates it has developed a deeper understanding of linguistic nuances and relationships, allowing it to handle ambiguous cases with much greater accuracy.



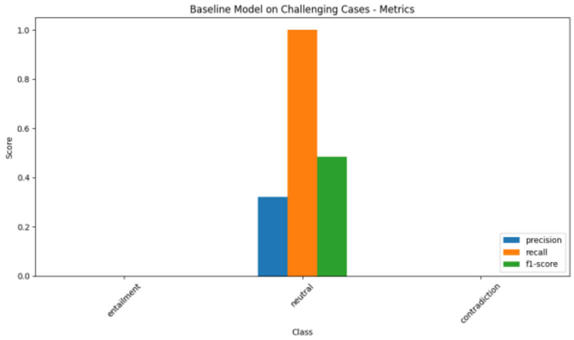
## 5.5 Error Comparison - Ambiguities

The error analysis for ambiguities provides a detailed breakdown of how the fine-tuned model improved over the baseline while also highlighting the challenges that remain. The pie chart reveals that 75.5% of errors made by the baseline model on ambiguous examples were successfully fixed by the fine-tuned model, demonstrating its capability to better handle nuanced and contextually complex examples. However, 16.3% of the errors introduced by the fine-tuned model indicate a trade-off; in attempting to generalize better on ambiguous cases, the model occasionally misclassifies previously correct predictions. This could stem from the fine-tuning process amplifying certain patterns that lead to overfitting or misalignment for edge cases. The remaining 8.2% of errors, which persisted across both the baseline and fine-tuned models, underscore the inherent difficulty in dealing with ambiguity. These cases often involve subtle semantic differences or interpretations that are challenging even for sophisticated language models. The results suggest that while adversarial training significantly enhanced the model's ability to generalize and address ambiguities, there remains an opportunity to further optimize performance, perhaps through additional strategies like incorporating more diverse ambiguous examples or refining the training process to minimize introduced errors.

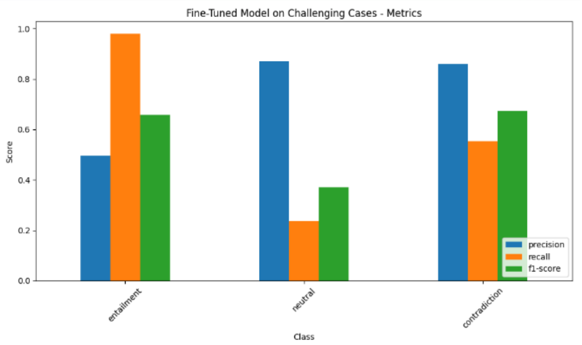


## 5.6 Metrics Comparison - Challenging Cases

The baseline model performs similarly on both negations and ambiguities, struggling to handle the complexities of either scenario, and this pattern extends to challenging cases as well. Across all these contexts, the model demonstrates a heavy reliance on the neutral class, achieving near-perfect recall but at the expense of precision. This similarity across negations, ambiguities, and challenging cases highlights a fundamental weakness in the baseline model's architecture or training. The model's inability to detect the effects of negations on sentence meaning mirrors its failure to resolve ambiguities where context or subtle cues play a critical role. Similarly, its performance on challenging cases demonstrates that small perturbations or nuanced alterations can easily overwhelm its simplistic decision-making process. These results suggest that the baseline model relies on a narrow, rigid strategy that fails whenever inputs deviate from straightforward examples, resulting in its uniform behavior of over-representing the neutral class in all scenarios.

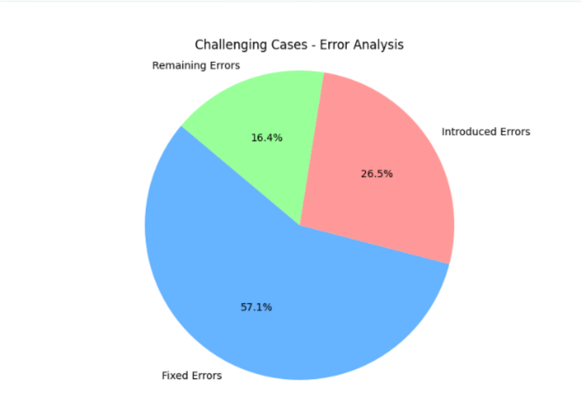


The fine-tuned model shows a marked improvement across all metrics for these challenging cases. It no longer disproportionately favors the neutral class and demonstrates balanced performance across entailment, neutral, and contradiction. For entailment, the model can identify relevant examples with much higher precision and recall, suggesting it has developed a better understanding of subtle affirmations or logical consistency even under adversarial pressure. The contradiction class also sees a significant boost in performance, indicating the model's enhanced ability to detect semantic conflicts that are often masked or obfuscated in challenging scenarios. Although the neutral class sees a slight reduction in recall compared to the baseline, this is balanced by improved precision, showing that the model is no longer defaulting to neutral predictions but is instead making more deliberate and accurate classifications. Overall, the fine-tuned model demonstrates the robustness and nuanced understanding needed to handle challenging cases effectively.

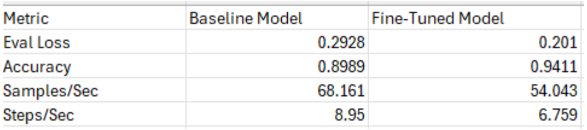


## 5.7 Error Comparison – Challenging Cases

The error analysis for challenging cases highlights the mixed outcomes of fine-tuning the model. The chart shows that 57.1% of errors made by the baseline model on challenging cases were successfully corrected by the fine-tuned model. This demonstrates that fine-tuning significantly improved the model's ability to handle complex examples, such as longer sentences or intricate semantic relationships. However, 26.5% of the errors introduced by the fine-tuned model reflect the model's struggle to generalize completely. These introduced errors suggest that the fine-tuning process, while enhancing performance in many areas, also led to overfitting or misinterpretation of certain complex patterns. Additionally, 16.4% of errors remained unresolved across both models, indicating the persistent difficulty of challenging cases that likely require further architectural changes or additional training data. The relatively high proportion of fixed errors affirms the efficacy of adversarial training, but the noticeable rate of introduced errors points to a need for further refinement in the training pipeline, particularly for mitigating overfitting in edge-case scenarios.



### 5.8 Overall Comparison



| Metric | Baseline Model | Fine-Tuned Model |
|---|---|---|
| Eval Loss | 0.2928 | 0.201 |
| Accuracy | 0.8989 | 0.9411 |
| Samples/Sec | 68.161 | 54.043 |
| Steps/Sec | 8.95 | 6.759 |

The final comparison of the baseline and fine-tuned models across all evaluation metrics shows significant improvements achieved through adversarial training when comparing the baseline and fine-tuned models. The reduction in evaluation loss and the increase in accuracy underscore the success of the fine-tuning process in delivering a more reliable and high-performing model.

## 6 Conclusion

This study highlights the significant impact of adversarial training in addressing the weaknesses of a natural language inference model. Starting with a basic model trained on the SNLI dataset, it had difficulty handling negations, ambiguities, and other complex linguistic cases. The baseline achieved an accuracy of 89.9% but showed poor performance in key, complex areas. By incorporating adversarial examples into the training process, the fine-tuned model demonstrated marked improvements, with accuracy rising to 94.1% and evaluation loss decreasing from 0.2928 to 0.2010.

The fine-tuned model addressed major weaknesses of the baseline, fixing a substantial number of errors while introducing minimal new ones. Enhanced precision, recall, and F1-scores across all classes, especially contradictions, highlight its ability to generalize better and handle complex cases. This work underscores the value of adversarial training in improving model robustness and addressing biases, setting a strong foundation for future research to further optimize performance in challenging scenarios.

## References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. *A Large Annotated Corpus for Learning Natural Language Inference*. Association for Computational Linguistics, Lisbon, Portugal.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. International Conference on Learning Representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Association for Computational Linguistics, Minneapolis, MN.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. *HotFlip: White-box Adversarial Examples for Text Classification*. Association for Computational Linguistics, Melbourne, Australia.

Paul Michel, Xian Li, Graham Neubig, and Joan Puigcerver. 2019. *On Evaluation of Adversarial Perturbations for Sequence-to-Sequence Models*. Association for Computational Linguistics, Minneapolis, MN.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. *Universal Adversarial Triggers for Attacking and Analyzing NLP*. Association for Computational Linguistics, Hong Kong.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. *TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP*. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards Debiasing NLU Models from Unknown Biases. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. *Hypothesis Only Baselines in Natural Language Inference. Association for Computational Linguistics*, New Orleans, LA.