Enhancing NLI Models With an Adversarial LLM Approach

Anonymous ACL submission

Abstract

In this paper, we demonstrate that the performance of natural language inference (NLI) models can be enhanced using a novel adversarial approach, in which large language models (LLMs) are used to systematically address NLI models' weaknesses. We first employ the LLMs to adversarially generate challenging NLI examples, looking for instances that are misclassified by the NLI model, effectively creating a dataset. These examples are validated by an ensemble of LLMs to ensure their correctness and are subsequently used to retrain the NLI model, iteratively refining its performance. In our evaluation, the proposed approach demonstrated substantial accuracy improvements on multiple datasets, including 1.65% on the SNLI dataset, 3.37% on the ANLI dataset, and 4.91% on the MultiNLI dataset. Our evaluation highlights the utility of LLMs in adversarial model improvement, providing a pathway toward robust and human-independent enhancements for NLI systems. Additionally, our LLM-based approach can also be used to automate the creation of NLI datasets.

1 Introduction

A fundamental task in natural language processing (NLP), natural language inference (NLI) is performed to determine the relationship between two sentences, ascertaining whether one sentence entails, contradicts, or is neutral to the other. While NLI models have achieved impressive performance, their robustness remains a challenge (Glockner et al., 2018; Carmona et al., 2018). Addressing these weaknesses is crucial for improving the reliability of NLI systems.

Inspired by the methodology used to create the adversarial NLI (ANLI) dataset (Nie et al., 2019), we propose a novel approach for automatically identifying and addressing the weaknesses of NLI models. Our approach leverages large language models (LLMs) to adversarially generate challenging NLI examples that aim to gather instances that are misclassified by the target NLI model. These examples are validated by an ensemble of LLMs to ensure their correctness before being used to retrain the NLI model. This iterative process focuses on strengthening the model's ability to handle difficult cases, ultimately improving its performance.

To evaluate our approach, we trained a leading NLI model using our approach and another data augmentation method, on the same amount of data, using 10 different sets of hyper-parameters. We then evaluated this model on three popular NLI test-sets and observed consistent improvements.

The contributions of our work are as follows: (1) our proposed approach systematically addresses NLI model weaknesses, improving their robustness and accuracy, as demonstrated by performance improvements on the SNLI (Bowman et al., 2015), ANLI, and MultiNLI (Williams et al., 2018a) datasets; (2) we introduce a fully automated dataset creation process that eliminates the traditional reliance on human annotators; and (3) our approach can scale to generate complete NLI datasets, enabling large-scale training of NLI models.

By combining automation, adversarial examples, and LLMs, our approach represents a significant step forward in enhancing NLI model performance and reliability. Moreover, by applying our method extensively to generate NLI examples, we can assemble a dataset that can be used to train NLI models.

2 Background and Related Work

Improving the robustness and performance of NLI models remains a significant challenge in natural language understanding (Glockner et al., 2018; Carmona et al., 2018). While traditional approaches heavily relied on manually created datasets, such as the Stanford NLI (SNLI) corpus (Bowman et al., 2015), this labor-intensive process highlighted the need for more efficient alternatives.

Recent advances in LLMs have enabled their use in the creation of NLI datasets, offering a more automated and scalable alternative to current practice. Our methodology leverages state-of-the-art LLMs such as Llama-3.1-70B (Touvron, 2023), Mistral-Large 2 (Jiang et al., 2023), and Mixtral-8x7B (Jiang et al., 2024) to generate and validate NLI examples. These models give our approach the ability to generate high-quality NLI examples and fine-tune NLI models like RoBERTa-Base (Liu et al., 2019), enhancing their robustness and performance.

Several recent studies have explored the use of LLMs for data generation. For example, counterfactual generation (Li et al., 2023) has been used to improve the robustness of the model in various downstream tasks, while paraphrasing (Klemen and Robnik-Šikonja, 2021) has facilitated the expansion of existing datasets. TextAttack (Morris et al., 2020) is a framework for adversarial attacks and data augmentation, which has proven to be effective in enhancing models.

In the domain of NLI datasets, ANLI (Nie et al., 2019) used a human-and-model-in-the-loop approach to iteratively identify and address model weaknesses by manually creating challenging examples. Similarly, SNLI, with its 570K manually labeled sentence pairs, has become a standard benchmark for evaluating NLI models. Building on SNLI, the MultiGenre NLI (MultiNLI) dataset (Williams et al., 2018b) consists of 433K sentence pairs from various text genres, enhancing the training and evaluation of the models' generalization capabilities and robustness in varied contexts.

3 Methodology

In this section, we describe the four stages in our suggested approach for improving NLI models. The complete flow is presented in Figure 1.

Automated Hypothesis Generation To create diversity in the hypotheses, we begin by inputting premises and their corresponding labels into multiple LLMs. These models are given examples of both correct and incorrect classifications made by the target NLI model and are then tasked with generating a hypothesis that aligns with the given premise, such that the given label reflects the relation between them. The pseudocode of the hypotheses generation is provided in Appendix A.1.

Adversarial Data Filtering Once the hypothesis is generated, it is sent, along with the premise, for classification by the target NLI model, which we try to improve. If the model assigns the correct label for the input pair, both the hypothesis and the premise are discarded. If the model misclassifies the input pair, the pair and its correct label continue to the validation stage. This is done because we want to gather examples that leading NLI models struggle with, in order to address their weaknesses.

Automated Validation The validity of a hypothesis misclassified by the NLI model is evaluated by an ensemble of three LLMs. These models act as independent judges, using majority voting to ensure robust, unbiased validation.

Iterative Refinement and Retraining If, in the previous stage, the LLMs agree on the validity of the misclassified example, the hypothesis and premise are then used for retraining. This iterative loop is aimed at refining the accuracy of the target NLI model. This process also enhances the training dataset by continually challenging the model and increasing its exposure to complex cases, thereby improving its overall robustness.

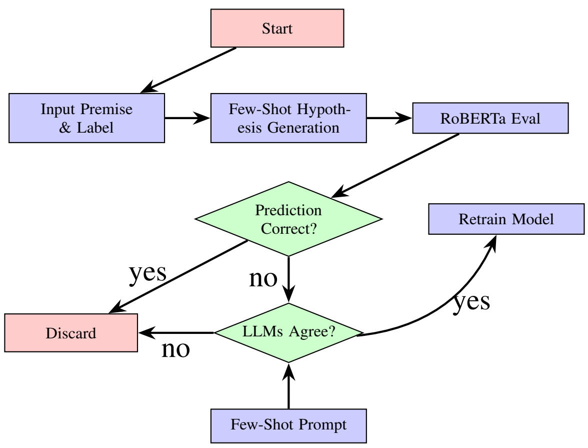

Figure 1: Illustration of our approach for improving an NLI model.

3.1 Dataset Comparison and Semantic Analysis

To gain insights into the relation between the data generated in out experiment and existing datasets, we examined the 10 most common non-stopwords in each dataset. We also assessed the similarity between the datasets using the TF-IDF and BERTScore F1 metrics (Zhang et al., 2019). The

TF-IDF metric, employing cosine similarity, measures lexical overlap to reveal how much vocabulary and how many syntactic patterns are shared between datasets. The BERTScore metric evaluates semantic similarity using contextual embeddings from transformer language models.

3.1.1 Key Findings From the Dataset Analysis

In the `SNLI train` dataset, some of the most frequent words are 'man,' 'woman,' and 'people,' indicating themes of gender and social interactions. In contrast, the `ANLI test` dataset focuses on media and chronology with words like 'film' and 'first,' while the `MultiNLI test` dataset uses more abstract language. The `Generated` dataset, containing misclassified examples, consist mainly of speculative and gender-focused language.

We also analyzed the hypotheses' length and word counts in the datasets. The hypotheses in the `Generated` dataset were the longest, whereas `SNLI train` and `SNLI test` had similar lengths, suggesting a consistent style. The `ANLI test` and `MultiNLI test` datasets had longer hypotheses, highlighting their complexity. A comparison of the text length and word counts in the hypotheses of the examined datasets is provided in Figure 2.

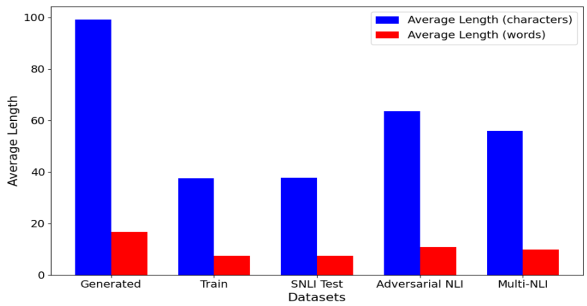

Figure 2: Average text length and word count in the hypothesis column for the examined datasets.

As for the similarity between datasets, Figure 3 presents the TF-IDF cosine similarity between every pair of the datasets' test sets. As can be seen, there is limited lexical overlap, with the greatest expected similarity between the `SNLI train` and `SNLI test` datasets and the least similarity between the `ANLI test` and `MultiNLI test` datasets. Figure 4 presents the BERTScore similarity; as can be seen, there are notable semantic alignments, particularly between the `SNLI train` and `SNLI test` datasets. These insights provide further validation of our approach, confirming that the data generated falls within the range of expected lexical and semantic similarities of existing NLI datasets.

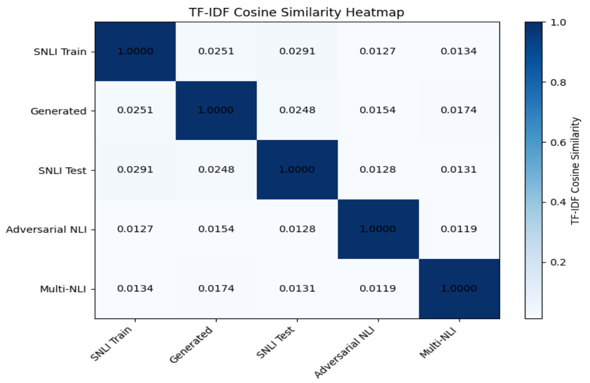

Figure 3: TF-IDF cosine similarity among NLI datasets, including our generated dataset.

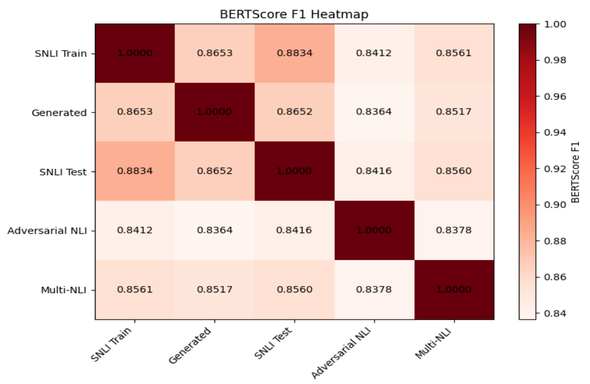

Figure 4: BERTScore F1 similarity among NLI datasets, including our generated dataset.

3.2 Avoiding Forgetness

One of the challenges of fine-tuning existing pretrained models is 'forgetness.' Providing a pretrained model with many new training examples from a different distribution may cause the model to overfit the new distribution and degrade its performance on the original distribution on which it was pretrained. To prevent this adverse effect, we added several examples from the original SNLI training set to the new training set we created with the newly generated examples. We experimented with different ratios of generated to original training samples and selected the ratio that maximized accuracy. The different ratios and their corresponding accuracy value are presented in Figure 5. The incorporation of both original and generated train samples also enhances their generalizability. This diversity helps models recognize a broader spectrum of patterns and scenarios, reducing the risk of overfitting and enabling more reliable performance.

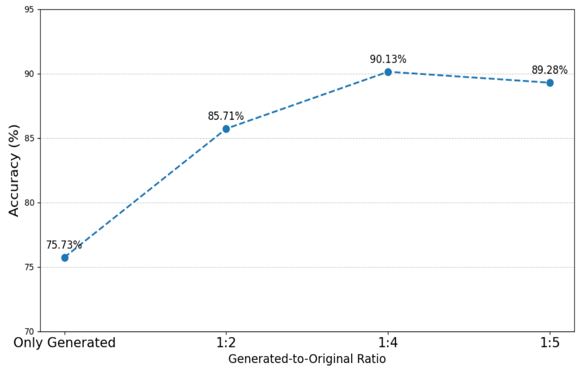

Figure 5: Model performance comparison across datasets.

4 Evaluation and Results

In this study, we used the RoBERTa-base-SNLI model from Hugging Face (HuggingFace, 2022) (125M parameters), a popular, open-source NLI model trained on a single dataset. To evaluate our approach, we generated, filtered, and validated thousands of data samples, ending up with 2.5K high-quality samples of NLI data according to our approach. We used Llama-3.1-70B and Mistral-Large 2 (123B) for generation and Gemini-2.0-Flash-Lite (Google, 2025), Mixtral-8x7B, and Qwen-2.5-72B-Instruct (Qwen, 2024) for validation. Then, we fine-tuned the RoBERTa-base-SNLI on it, along with another 10K samples from the original SNLI train set, to maintain our suggested ratio of 1:4. To fine-tune the NLI model, we used a single T4 GPU. We conducted experiments using 10 different sets of hyperparameters to confirm the robustness of our approach. This evaluation demonstrates notable improvements across three different and diverse test sets. In the first experiment, conducted on the SNLI test set, the model trained on our data achieved accuracy of **90.13%**, surpassing the RoBERTa-base-SNLI's accuracy of 88.48%. This demonstrates that our approach effectively boosts performance on the dataset that the base model was originally trained on. In the second experiment, using the ANLI test set, our model again outperformed RoBERTa-base-SNLI, achieving an accuracy of **78.41%** compared to 75.04%. This result shows that our approach improved the model's ability to handle challenging adversarial examples. Finally, on the MultiNLI dataset, the model trained on our data achieved an accuracy of **59.58%**, which is significantly higher than RoBERTa-base-SNLI's accuracy of 54.67%. This emphasizes the enhanced generalization capabilities of our approach across diverse data distributions. For comparison, we fine-tuned the same model on the same amount of data taken from the MNLI train set. We also performed paraphrasing to transform the same amount of samples from MNLI. This approach achieved moderate improvements, with accuracies of 84.73% on SNLI, 72.39% on ANLI, and 50.01% on MultiNLI, but remained below the performance of our proposed method. These results are summarized in Table 1.

Dataset	**RoBERTa base-SNLI**	**Additional Data**	**Para-phrasing**	**Our Approach**
SNLI	88.48%	89.42%	84.73%	**90.13% ± 0.67**
Adversarial NLI	75.04%	77.07%	72.39%	**78.41% ± 0.31**
MultiNLI	54.67%	57.61%	50.01%	**59.58% ± 0.71**

Table 1: Comparison of accuracy on the examined datasets, for RoBERTa-base-SNLI, RoBERTa-base-SNLI fine-tuned with additional data from MNLI, RoBERTa base-SNLI fine-tuned with additional data generated using paraphrasing based on the SNLI train set, and RoBERTa-base-SNLI fine-tuned with additional data generated using our approach.

5 Discussion and Future Research

This study demonstrates the effectiveness of employing LLMs to automatically identify and address NLI models' weaknesses by generating and validating challenging datasets. By targeting model misclassifications, our approach systematically enhances NLI model robustness and accuracy, achieving significant performance improvements on diverse datasets - SNLI, ANLI, and MultiNLI. Our approach represents a major step forward in automating model refinement, reducing reliance on human annotators while preserving data quality and consistency.

Using an ensemble of LLMs for hypothesis validation reduces human biases and errors while enabling a scalable, iterative process for creating complete NLI datasets. This scalability supports both retraining existing models and building comprehensive datasets for future NLI models.

Future research should explore ways to further diversify the data generated by LLMs, incorporating varied linguistic structures and content domains. To explore our approach's potential to further address model weaknesses, its performance when employed on a larger scale and with multiple iterations should be explored. Additionally, applying these techniques to other NLP tasks could examine our approach's utility in other domains.

6 Limitations

Our approach's dependence on the initial quality of LLMs and the substantial computational resources required for training and deploying multiple models simultaneously could be prohibitive for some applications. This research was conducted with low-resource computation, which imposed certain constraints, limiting the scale and speed of processing. Additionally, the use of outsourced APIs for model generation introduced a bottleneck, as API response times delayed the generation of necessary data. These limitations prevented us from generating data at scale and testing our approach by generating hundreds of thousands of examples. We also have not yet examined our approach cyclically, using the model trained with our data as a target model for another iteration of data generation. We plan to address these limitations in future research.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Vicente Iván Sánchez Carmona, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of nli models: Uncovering the influence of three factors on robustness. *arXiv preprint arXiv:1805.04212*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*.

Google. 2025. Gemini 2.0 flash-lite. https://deepmind.google/technologies/gemini/flash-lite/.

HuggingFace. 2022. pepa/roberta-base-snli. Accessed: October 12, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Matej Klemen and Marko Robnik-Šikonja. 2021. Extracting and filtering paraphrases by bridging natural language inference and paraphrasing. *arXiv preprint arXiv:2111.07119*.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. 2023. Prompting large language models for counterfactual generation: An empirical study. *arXiv preprint arXiv:2305.14791*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv*.

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. *arXiv preprint arXiv:2005.05909*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial NLI: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Qwen. 2024. Qwen2.5-72b-instruct. https://qwen2.org/qwen2-5.

Hugo Touvron. 2023. Llama: Open and efficient foundation language models. *arXiv*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018a. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018b. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Appendix

A.1 Model Prompting Procedure for Generation

The few-shot generation process of our approach is described in Algorithm 1. This process uses curated examples to guide the model in generating hypotheses that align with the desired premise-hypothesis relationship. These examples were meticulously selected to include both instances where the target model failed to classify correctly and those where it achieved successful classification. By utilizing these examples, the LLM generates contextually appropriate, more informed, and accurate hypotheses, thereby enhancing efficiency and consistency.

Algorithm 1 Few-Shot Hypothesis Generation

Shuffle the SNLI train dataset D

Select n observations from D: $\{(p_1, h_1, l_1), (p_2, h_2, l_2), \ldots, (p_n, h_n, l_n)\}$ such that there are an equal number of observations for each label

for each (p_i, h_i, l_i), where $i \in \{1, \ldots, n\}$ **do**

  Format the example as:
  `This is a premise:` p_i`, this is the hypothesis:` h_i`, and the label between them is` l_i`.`

end for

Provide these n formatted examples as few-shot inputs to the model

After providing the examples, prompt the model with the following instruction:
`You are a language expert that helps create an NLI dataset. Given a premise sentence` p `and a desired label` l`, generate a one-sentence hypothesis` h `such that the label is relevant to the relation between the premise and the generated hypothesis. Keep the hypothesis short.`

The model generates a one-sentence hypothesis h for the given premise p and label l

return Generated hypothesis h

In Table 2, we present the final prompt structure used, which includes detailed instructions, carefully selected examples, and a structured response format. This design ensures that the generated hypotheses align with the desired premise-hypothesis relationship while maintaining consistency and reducing ambiguity in the output. The few-shot examples are shown in Appendix A.5.

A.2 Model Prompting Procedure for Validation

In Table 3, we present the final prompt used for LLM validation of the NLI dataset. The prompt asks the model if the provided label matches the premise-hypothesis relationship, with the system responding 'Accepted' or 'Not Accepted.' This process is repeated with multiple LLMs to filter challenging and problematic examples. The prompt was designed with detailed instructions, illustrative examples, and a structured response format to ensure consistency and accuracy in the validation process, contributing to the overall quality and robustness of the dataset.

Component	Content
Few-Shot Example	Here are cases where the target model made mistakes: This is a premise: {premise} This is the hypothesis: {hypothesis}. The label between them is {label} ($\mathcal{L}_{incorrect}$). (repeat for four incorrect examples) Now, here are cases where the taget model got it right: This is a premise: {premise} This is the hypothesis: {hypothesis}. The label between them is {label} ($\mathcal{L}_{correct}$). (repeat for four correct examples) (Eight examples are shown to the model in this format, randomly selected from the correct and incorrect predictions to ensure balanced representation of successful and failed classifications.)
System Prompt	You are a language expert that helps create an NLI dataset. Given a premise and a desired label, your job is to provide a one-sentence hypothesis such that the label is relevant to the relation between the given premise and your generated hypothesis.

Table 2: Prompting procedure used to generate hypotheses for the NLI dataset.

Component	Content
System Prompt	You are a language expert. Your job is to filter rows of an NLI dataset, which contain some data that may not be good enough. Given a premise and a hypothesis, you should determine whether the label reflects the relationship between them or not.
User Prompt	This is the premise: {premise}. This is the hypothesis: {hypothesis}. The relationship between them is {label}. Do you accept this relationship? Respond only with 'Accepted' or 'Not Accepted.'

Table 3: Prompting procedure used to validate the NLI dataset examples.

A.3 Optimized Hyperparameters for RoBERTa-base-SNLI Model

After 10 experiments, the best RoBERTa-base-SNLI hyperparameters were: learning rate 5.31×10^{-6}, batch size 16/8, one epoch, and weight decay 0.0093, balancing efficiency and generalization.

A.4 Examples of Generated Hypotheses

In Table 4, we provide some examples of the hypotheses generated. Each row contains the original premise, the generated hypothesis, and the original label, highlighting the model's generalization ability.

Premise	Hypothesis (Generated)	Label
A small girl with a necklace is swimming.	There is unlikely to be a relationship between the material composition of the necklace and the girl's swimming proficiency.	1
Swimmers leap off the starting blocks into their race lanes at an indoor pool.	Athletes jump off the starting blocks into their designated lanes at the beginning of a swimming competition.	0
Two women are sitting at a table working with clay.	The women are engaged in a quiet activity.	1
Young man playing darts in a curtained room.	A young man is throwing darts in a private space.	0
3 people in a small hut or house.	There are more than 10 people in the hut.	2

Table 4: Examples of generated hypotheses with their corresponding original labels.

A.5 Examples of Correct and Incorrect Predictions

Table 5 presents a set of examples illustrating both correct and incorrect classifications made by the model when predicting the label for a given premise and its corresponding generated hypothesis. The first four rows highlight instances where the model failed to assign the correct label, showcasing cases where the classification was erroneous. In contrast, the last four rows contain examples where the model successfully identified the correct label, demonstrating its capability to accurately recognize the premise-hypothesis relationship. These examples were specifically incorporated into the few-shot generation process to provide the language model with informative guidance during hypothesis generation. By including both misclassified and correctly classified examples, the few-shot approach ensures that the model learns from past errors while reinforcing successful patterns, ultimately improving the quality, consistency, and robustness of the generated hypotheses.

Premise	Hypothesis	Label
A woman with a green headscarf, blue shirt and a very big grin.	The woman appears to be in distress after a violent incident.	2
A land rover is being driven across a river.	A car is parked on the side of the road.	2
People are cleaning up a street.	A group of individuals are picking up trash and debris from the street.	0
Three firefighters come out of a subway station.	Three people in casual clothes walk out of an airport.	1
This church choir sings to the masses as they sing joyous songs from the book at a church.	The church has cracks in the ceiling.	1
A woman with a green headscarf, blue shirt and a very big grin.	The woman is young.	1
A man playing an electric guitar on stage.	A man playing banjo on the floor.	2
A young family enjoys feeling ocean waves lap at their feet.	A young man and woman take their child to the beach for the first time.	1

Table 5: Examples of correct and incorrect model predictions. The first four rows illustrate cases where the model misclassified the label, while the last four rows show cases where the model correctly predicted the label.

A.6 Ablation Study

In this section, we conduct a detailed analysis of each component of our method to better understand its contribution and overall impact on the final dataset. A critical aspect of our approach is the multi-stage validation process, which systematically filters out lower-quality or less adversarial samples. We begin by generating 30K samples, which then undergo an initial filtering phase where the target model classifies them. In this stage, 15K samples are discarded because they are correctly classified by the target model, meaning that only the remaining 50% of the generated data contains sufficiently challenging examples for further validation.

Following this initial filtering, we employ a secondary validation step using three large language models (LLMs) to further refine the dataset. These LLMs independently assess the remaining 15K adversarial samples, filtering out 12.5K of them. At this stage, only one out of every six examples is approved by the majority voting mechanism of the three LLMs, ensuring that only the most adversarial and informative samples are retained. After this rigorous multi-stage filtering process, we are left with a final dataset consisting of 2,500 high-quality adversarial samples, accounting for just 8.33% of the original 30K generated examples.

Beyond analyzing data filtration, we also investigate the impact of each individual component in our approach. Table 6 presents an ablation study where we assess the accuracy of the SNLI test when specific components of our method are included or removed. This analysis helps to isolate the effectiveness of each component, providing a deeper understanding of how various aspects of our approach contribute to improving model robustness and performance.

Method	**SNLI Test Accuracy**
Target Model + Few-Shot Generated examples	89.25%
Target Model + Few-Shot Generated examples + Only Adversarial Samples	89.64%
Target Model + Few-Shot Generated examples + Only Adversarial Samples + LLMs validation (our approach)	90.13%

Table 6: Ablation study, with the performance of each component on the SNLI test.