Streamlining NLI Dataset Creation: A Large Language Model Approach

Anonymous ACL submission

Abstract

Our study enhances the methodology for creating Adversarial Natural Language Inference datasets by replacing the traditional reliance on human annotators with large language models (LLMs). We employ two LLMs to generate NLI examples, which are then used to evaluate a leading NLI model and identify misclassified instances. These instances undergo further validation by an ensemble of three additional LLMs, collectively assessing their correctness. This process focuses on correcting model weaknesses and retrains the model using challenging examples. Our method improves performance across diverse datasets, increasing accuracy by 1.43% on SNLI, 2.75% on ANLI, and 4.29% on MultiNLI. By automatically targeting model weaknesses, our approach simplifies dataset creation and enhances robustness without human intervention, highlighting the efficacy of using LLMs for adversarial dataset generation.

1 Introduction

Natural Language Inference (NLI) is a fundamental task in Natural Language Processing (NLP), which involves determining the relationship between two sentences — whether one entails, contradicts, or is neutral to the other. Traditionally, the creation of NLI datasets has heavily relied on human annotators to generate and label examples (Oana-Maria, 2018; Nie et al., 2019), a process that is both time-consuming and susceptible to inconsistencies. To overcome these challenges, we propose a novel approach that utilizes Large Language Models (LLMs) to generate and filter NLI examples, thus superseding the need for human annotators.

Building on the methodology used to create adversarial NLI (ANLI) datasets (Nie et al., 2019), we leverage LLMs to produce high-quality data efficiently. Our approach uses two LLMs to generate NLI examples, followed by a high-performance NLI model to filter out examples that are correctly classified. Three additional LLMs then serve as judges for further validation, and the NLI model is retrained on the validated, challenging examples.

Our contributions include a fully automated process that reduces human intervention while preserving integrity and enhancing the robustness of NLI datasets. This sets our approach apart from traditional adversarial dataset creation methods. The primary innovation lies in addressing model weaknesses more effectively than previous approaches. By identifying and focusing on the specific instances that NLI models struggle to classify correctly, our process enhances model robustness. The subsequent sections will elaborate on the steps of our methodology and the corresponding enhancements in model performance.

2 Background and Related Work

Historically, NLI dataset creation relied heavily on human annotators, making the process labor-intensive and prone to errors. The Stanford NLI (SNLI) dataset (Oana-Maria, 2018), a key resource in NLI research, exemplifies these traditional approaches. Active learning (AL) aims to reduce the need for human annotations by optimizing the expected value of each one. However, contrary to our method, AL does not rely on existing labels, which prevents it from using misclassification as an indicator for the value of a sample. Recent advancements in language models have led to the integration of LLMs into NLI dataset creation, allowing for the automatic generation and evaluation of examples. Our methodology uses models 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 data. In this study, we use LLMs to improve SNLI for fine-tuning a RoBERTa-Base model (Liu et al., 2019).

Recent work involving LLMs for data generation includes the use of LLMs for counterfac-

tual generation to improve the model's robustness across many downstream tasks (Li et al., 2023), and bridge paraphrasing and NLI to create cleaner datasets (Klemen and Robnik-Šikonja, 2021). Additionally, (Morris et al., 2020) introduced TextAttack, a framework for adversarial attacks and data augmentation to enhance NLP models.

As for NLI datasets, ANLI (Nie et al., 2019) was developed using a human-and-model-in-the-loop approach to identify and address model weaknesses, adding hard examples iteratively to improve robustness. Similarly, the SNLI dataset (Bowman et al., 2015), with 570k manually labeled sentence pairs, has become a cornerstone for evaluating NLI models. The Multi-Genre NLI (MultiNLI) dataset (Williams et al., 2018) extends this work by introducing 433k sentence pairs from diverse genres to test generalization across various text types, advancing model robustness and transferability.

3 Methodology

This section will cover the stages of our suggested process for improving an NLI model. The complete flow is shown in Figure 1.

3.1 Automated Hypothesis Generation

Our methodology leverages LLMs to automate the generation of hypotheses, thus eliminating the dependency on human annotators. To create diversity, we begin by inputting premises and their corresponding labels into multiple LLMs. These models are tasked with generating a hypothesis that aligns with the given premise, such that the given label holds for the relation between them. The full algorithm is described in Algorithm 1 in Appendix A.

3.2 Adversarial Data Filtering

Once the hypothesis is generated, it is sent with the premise for classification with a pre-trained NLI model, which we try to improve. If the model classifies the correct label for the input pair, both the hypothesis and the premise are discarded. Otherwise, the pair and their label will continue to the next stage of validation. This is because we seek to gather examples on which the current leading NLI models struggle, thus fixing their weaknesses and creating challenging examples.

3.3 Automated Validation Process

If a hypothesis is incorrectly predicted by the NLI model, it enters a validation phase where three additional LLMs independently assess its validity. These models act as judges in an ensemble to ensure a robust validation, free from the biases of a single model's assessment. The agreement among these models, in the form of a majority vote, dictates whether a hypothesis is accepted or subjected to further iteration.

3.4 Iterative Refinement and Retraining

Should the LLMs agree on the validity of the misclassified example, the hypothesis and premise are then reintroduced into the system for retraining. This iterative loop is aimed at refining the accuracy of the base NLI model. This process enhances the training dataset by continually challenging and expanding the model's exposure to complex cases, thereby improving its overall robustness.

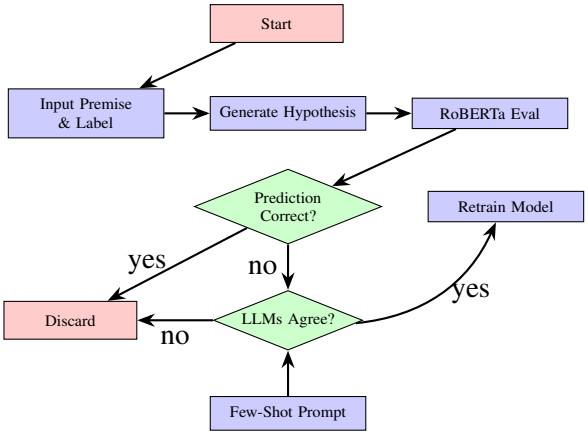

Figure 1: Illustration of the process of improving an NLI model

3.5 Dataset Comparison and Semantic Analysis

To gain insights into the relations between our generated data and the existing datasets, we examined the ten most common non-stopwords in each dataset. Additionally, we assessed the similarity between datasets using TF-IDF and BERTScore F1 metrics (Zhang et al., 2019). TF-IDF, employing cosine similarity, measures lexical overlap to reveal how much vocabulary and syntactic patterns are shared between datasets. BERTScore evaluates semantic similarity using contextual embeddings from transformer language models.

3.5.1 key findings in dataset analysis

In `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` uses more abstract language. The `Generated` dataset, containing misclassified examples, features speculative and gender-focused language.

We also analyzed text lengths and word counts across datasets. The `Generated` dataset exhibited the longest hypotheses, whereas `SNLI train` and `SNLI Test` had similar lengths, suggesting a consistent style. The `ANLI test` and `MultiNLI test` datasets showed longer hypotheses, highlighting their complexity. The length analysis can be seen in Figure 2.

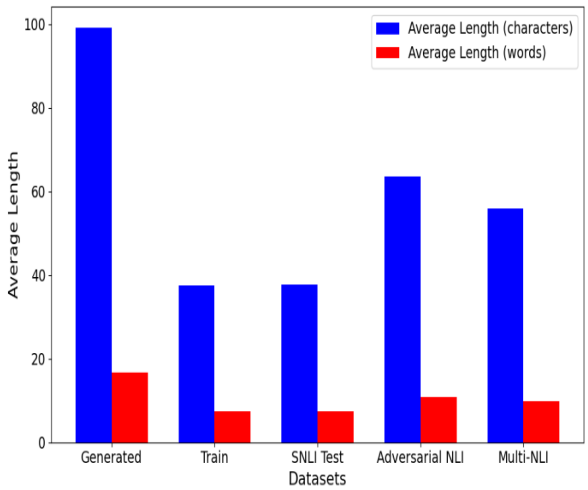

Figure 2: Average text length and word count in hypothesis column across datasets

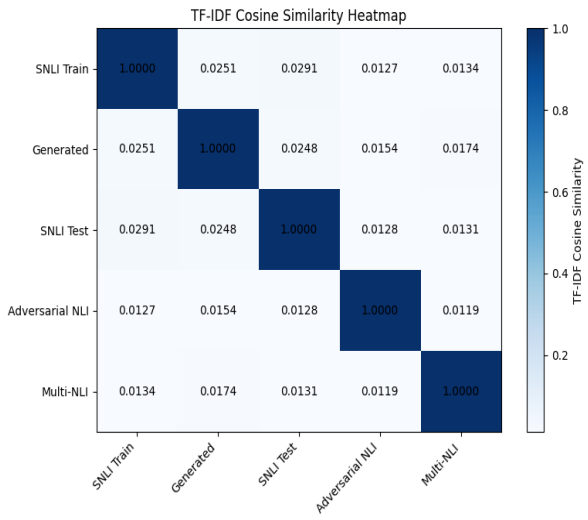

Figure 3: TF-IDF cosine similarity between NLI datasets, including our generated data.

As for similarity between datasets, Figure 3 shows the TF-IDF cosine similarity between every pair of the datasets' test sets. It can be seen

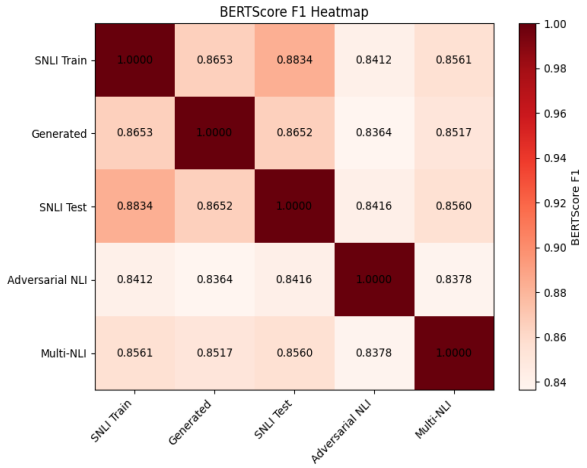

Figure 4: BERTScore F1 similarity between NLI datasets, including our generated data.

that there is limited lexical overlap, with the highest expected similarity between `SNLI Train` and `SNLI Test` and the lowest between `ANLI test` and `MultiNLI test`. Figure 4 shows the BERTScore similarity, in which there are notable semantic alignments, particularly between `SNLI Train` and `SNLI Test`. These insights provide another layer of validation to our method, confirming that our generated data falls within the range of expected lexical and semantic similarities of existing NLI datasets.

3.6 Avoiding Forgetness

One of the challenges of fine-tuning existing pre-trained models is 'forgetness.' Providing a pre-trained model with many new training examples from a possibly different distribution may lead to the model overfitting the new distribution and decreasing performance on the original distribution on which it was pre-trained. To avoid this adverse effect, we added several examples from the original SNLI train set to the new train set we created with the newly generated examples. We have experimented with different ratios of generated to original train samples and selected the one that maximized accuracy. The different ratios and their corresponding accuracy are shown 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 over-fitting and enabling more reliable performance.

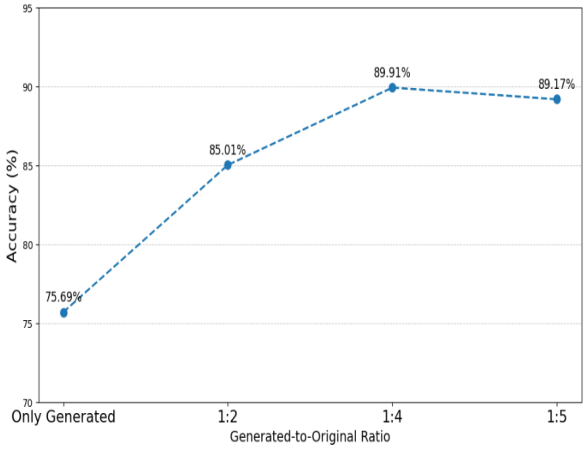

Figure 5: Model performance comparison across datasets

4 Evaluation and Results

For this study, we looked for a popular, open-source NLI model that is trained on a single dataset, to be able to test our method on other datasets that the model has never seen before. We used the RoBERTa-base-SNLI model from HuggingFace (HuggingFace, 2022) (125M parameters). To evaluate our approach, we generated, filtered, and validated thousands of data samples, ending up with 2.5K good-quality samples of NLI data according to our method. In this study, we used Llama-3.1-70B and Mistral-Large 2 (123B) for generation, and Mistral-Large 2, Mixtral-8x7B, and Llama-3.1-70B for validation. Then we fine-tuned the RoBERTa-base-SNLI on it using the along with another 10K samples from the MNLI train set, to keep our suggested ratio of 1:4. To fine-tune the NLI model, we used a single T4 GPU. This evaluation demonstrates notable improvements across three different test datasets. In the first test, done on the SNLI test set, the model trained on our date achieved an accuracy of 89.91%, surpassing the base model's accuracy of 88.48%. This indicates that our enhancements effectively boost performance on the same dataset that the base model was originally trained on. In the second test, using the ANLI test set, our model data again helped outperform the base model, with an accuracy of 77.79% compared to 75.04%. This suggests our data improved the ability to handle challenging adversarial examples. Finally, on the MultiNLI dataset, the model trained on our data achieved an accuracy of 58.96%, significantly higher than the base model's 54.67%, highlighting its enhanced generalization capabilities across diverse data distributions. In addition, we fine-tuned the same model on the same amount of data, taken from the MNLI train set, for comparison of the quality of our data against existing NLI data. These results are summarized in Table 1.

Dataset	Base Model	Fine-Tuned Base Model	Our Approach
SNLI	88.48%	89.42%	**89.91%**
Adversarial NLI	75.04%	77.07%	**77.79%**
MultiNLI	54.67%	57.61%	**58.96%**

Table 1: Comparison of the accuracy differences across datasets for the base model, fine-tuned base model on other NLI data, and fine-tuned base model using our approach for generating data. The best results are in bold.

5 Discussion and Future Research

This study has demonstrated the effectiveness of employing LLMs to automate the generation and validation of NLI datasets. Our findings reveal that this methodology completely reduces the reliance on human annotators, simplifying the dataset creation process while enhancing the robustness and generalizability of NLI models. Using an ensemble of LLMs for hypothesis validation mitigates potential human biases and errors, and leads to noticeable improvements in model performance across diverse datasets, including SNLI, ANLI, and MultiNLI.

Future research should focus on expanding the diversity of data generated by LLMs to include more varied linguistic structures and content areas. It is also worthwhile to test this approach at scale, using multiple iterations and generating more samples. Furthermore, applying the insights from this study to other downstream tasks could help assess the generalizability of this approach and foster advances in models across many domains.

In conclusion, incorporating LLMs into the NLI dataset generation and validation process shows the potential of using LLMs to fix biases in smaller task-dedicated models. It offers a scalable, efficient alternative to traditional methods, leading to substantial improvements in dataset quality and model training processes. As LLMs continue to evolve, with the recent trend of cost-efficient smaller models that are still performing well, the methodology presented in this study should be expanded to improve models across many NLP tasks.

6 Limitations

The dependency 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 while generating hundreds of thousands of examples. We also have not yet tested our approach cyclically, using the model trained with our data as a base 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.

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*.

Camburu Oana-Maria. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems 31*.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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 Validation

In this section, we present the final prompt used for LLM validation of the NLI dataset, as shown in Table 2.

Component	Content
System Prompt	You are a language expert. Your job is to filter rows of an NLI dataset, which contains some data that may not be good enough. Given a premise and a hypothesis, you should determine whether the label suits the relationship between the sentences 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 2: Prompting procedure used to validate the NLI dataset examples.

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 across multiple LLMs to filter problematic examples.

A.2 Model Prompting Procedure for Generation

We use a few-shot learning approach for hypothesis generation, providing the model with SNLI Train examples. Table 3 shows the final prompt used.

Component	Content
Few-Shot Example	This is a premise: {premise} This is the hypothesis: {hypothesis} The label between them is {label}. (8 examples are shown to the model in this format, drawn randomly from the SNLI Train dataset)
System Prompt	You are a language expert that helps create an NLI dataset. Given a premise sentence 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. Make sure to keep the hypothesis short and no longer than a sentence.

Table 3: Prompting procedure used to generate hypotheses for the NLI dataset.

The few-shot learning process used in the hypothesis generation phase is described by the following pseudo-code:

Algorithm 1 Hypothesis Generation using Few-Shot Learning

1: Shuffle the SNLI Train dataset D
2: Randomly select n observations from D: $\{ (p_1, h_1, l_1), (p_2, h_2, l_2), \ldots, (p_n, h_n, l_n) \}$
3: **for** each (p_i, h_i, l_i), where $i \in \{1, \ldots, n\}$ **do**
4: Format the example as:
`This is a premise:` p_i`, this is the hypothesis:` h_i`, and the label between them is` l_i`.`
5: **end for**
6: Provide these n formatted examples as few-shot inputs to the model
7: 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.`
8: The model generates a one-sentence hypothesis h for the given premise p and label l
9: **return** Generated hypothesis h

This generation process helps the model create a hypothesis that aligns with a given premise and label by first showing it several few-shot examples from the SNLI Train dataset. After these examples, the model is tasked with generating hypotheses following the same pattern, ensuring relevance and consistency.

A.3 Optimized Hyperparameters for RoBERTa-base-SNLI

In this appendix, we provide the optimized hyperparameters for the RoBERTa-base-SNLI model. After tuning, the best-performing parameters were: a learning rate of 5.31×10^{-6}, a per-device training batch size of 16, a per-device evaluation batch size of 8, one training epoch, and a weight decay of 0.0093. These values were carefully selected to balance training efficiency and model generalization. The learning rate was adjusted to avoid overfitting, while the batch sizes optimized the use of available computational resources. The weight decay was included to regularize the model and improve its performance on unseen data.