Dataset Artifacts Analysis and Mitigation

Abstract

Even though modern fine-tuned ELECTRA-small model can achieve 90% human-like accuracy on benchmarks tests for Natural Language Inference (NLI), these results may not reflect the real world intelligence or problem-solving skills of this Large Language model (LLM). This high accuracy result may be the outcome of dataset artifacts exploitation: spurious correlations acting as shortcuts to provide correct predictions for these inferences. This report surveys some analysis techniques for uncovering dataset and artifacts. Furthermore, fine mitigation techniques for fixing or addressing them, with an emphasis on Stanford Natural Language Inference (SNLI) dataset.

1 Introduction

The modern ELECTRA-small model (Clark et al., 2020) has been trained and finetuned to provide 90% accuracy against the SNLI dataset. This paper will be analyzing multiple techniques such as Contrast Sets, Adversarial Natural Language Inference (ANLI) Dataset (from Facebook), Contra-factual datasets, both augmented and generated, along with the Uncertain SNLI Dataset. These datasets were used to analyze the baseline ELECTRA-small to understand the performance of the baseline model. For Contrast sets, the the perturbations were required, however this would be a tedious and deliberate process, with the lack or time and resources to conduct these perturbations, for this purpose a pipeline was implemented to generate Contrast set that was augmented using ACE (Packard, 2018) and English Resource Grammar (ERG) (Flickinger et al., 2014). This augmented dataset was run against the GPT-OSS model (OpenAI, 2025) with llama-cpp-python framework (Gerganov, 2023b) to select the top-k candidates. ELECTRA-small was then fine-tuned using these datasets and filtered by leveraging dataset cartography method to create a revised dataset that can be used to improve the robustness and accuracy. (see implemetation section 4 and analysis section 5.

2 Related Work

2.1 Contrast Sets

Contrast sets add minimal perturbation 1 to the SNLI training dataset to keep but alter the inference, these perturbations can cause the fine-tune model evaluation to fail at boundary conditions.

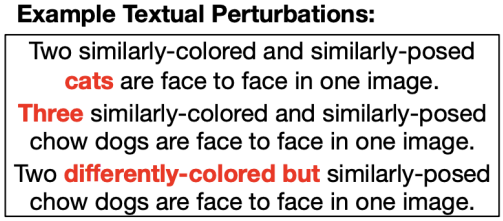


Figure 1: Examples of textual perturbations

We can use this idea to generate new train and test dataset by only altering the original dataset by a small and controlled way. It was introduced by (Gardner et al., 2020) and it will reveal whether a model relied on spurious features or not. However, manually constructing these contrast sets is tideous. For some datasets like SNLI, (Li et al., 2020) introduced an automated way of generating large datasets. Linguistically Informed Transformation (LIT) method enables practitioners to explore linguistic phenomena of interests as well as compose different phenomena. It is HPSG based (Copestake and Flickinger, 2000), optimized by Answer Constraint Engine (ACE) (Packard, 2018) and configurable by English Resource Grammar (ERG) (Flickinger et al., 2014). Auto-generated sentences can be further refined by more capable LLM like GPT-OSS-20B (OpenAI, 2025) Q4 quantized model to pick top-k best ERG candidates.

2.2 CheckList

CheckList (Ribeiro et al., 2020) is a task-agnostic behavioral testing methodology inspired by software unit tests. It defines a matrix of linguistic capabilities (e.g. vocabulary, negation, coreference, temporal reasoning) and test types (minimum functionality tests, invariance tests, directional expectation tests). Using a mix of templated generation and manual crafting, it produces many test cases targeting specific phenomena.

2.3 Adversarial Challenge Sets

Adversarial challenge sets (Jia and Liang, 2017) are evaluation datasets constructed to trick models by exploiting their learned shortcuts. Unlike contrast sets (which are minimal edits preserving correctness), adversarial examples often add distracting or confusing content to cause incorrect answers. This work directly prompted the creation of SQuAD 2.0 (adding unanswerable questions) to make models focus on when no answer is present. For NLI, adversarial sets have exposed models' reliance on annotation artifacts. The HANS dataset (McCoy et al., 2019) consists of inference examples designed to break models that use lexical overlap heuristics. Adversarial NLI (Nie et al., 2020) had humans iteratively craft NLI examples that fool the current model, yielding three rounds of increasingly difficult data.

2.4 Dataset Cartography

Dataset cartography (Swayamdipta et al., 2020) is an approach to map out the training data based on training dynamics calculated by comparing different training checkpoints. The main idea is that not all training examples are equal – some are learned easily by the model, some are consistently mislabeled, and some are in between. By tracking each example's model confidence in the correct label and variability of that confidence across epochs, one can visualize data in a 2D map. This reveals three regions: easy examples (high confidence, low variability – model learns them early and never falters), ambiguous examples (medium confidence or high variability – sometimes right, sometimes wrong), and hard examples (low confidence, often never correctly learned). Then it can filter the original dataset by defining certain threshold ratios to reduce the training dataset and improve robustness. They found that the ambiguous region contributes most to out-of-distribution generalization. In other words, they found the easy region, while often the largest portion of dataset, can contain many artifact-driven samples that the model memorizes without truly generalizing. By comparison, the hard region often contains wrongly mislabeled samples.

2.5 Counterfactually Augmented Data

NLI models are known to provide spurious results, but there is no standard mechnism that defines how the models are generating the spurious or non-spurious results. Causal frameworks provide a rigorous definition of spuriousness, attributing such associations to confounding factors rather than to genuine causal mechanisms. Without leveraging the mathematic behind causality, the team working to create contrafactual Augmented dataset using human reviewers in the data generation pipeline.

2.6 Hypothesis Only Baselines in Natural Language Inference

The paper (Poliak et al., 2018) proposes a hypothesis only baseline for diagnosing Natural Language Inference (NLI). Especially when an NLI dataset assumes inference is occurring based purely on the relationship between a context and a hypothesis, it follows that assessing entailment relations while ignoring the provided context is a degenerate solution.

The Hypothesis Only Baselines in Natural Language Inference (Poliak et al., 2018) lingusitic reason as to why the NLI models have failures, the models typically perform very well with entailment following the lexical patterns, however tweaks to the lexical patterns breakdown when analyzing the similar statements for non-entailment. This is where this paper is able to highlight inherent weakness in the NLI models, and can be used a guideline for creating datasets designed to break lexical dependence, to move towards meaningful reasoning of the hypothesis entailment. This dataset though only has entailment and non-entailment, since the lexical change performed resulted in ambiguous label, and neutral vs contradiction labeling could not be ascertained with reasonable certaintity.

2.7 Uncertain Natural Language Inference (UNLI)

Uncertain Natural Language Inference (UNLI), (Chen et al., 2019) a refinement of Natural Language Inference (NLI) that shifts away from categorical labels, targeting instead the direct prediction of subjective probability assessments.

Uncertain Natural Language Inference (UNLI)(Chen et al., 2019) refines traditional NLI by transitioning from discrete categorical labels to the direct prediction of subjective human probability assessments. We demonstrate that these elicited probabilities reveal granular distinctions regarding the likelihood of a hypothesis given a premise—nuances that UNLI captures

with significantly greater fidelity than the rigid labeling schemes of popular NLI datasets.

3 Proposals

We proposed a new pipeline to modernize and ruggedize 5 years old ELECTRA-small model by analyzing various dataset artifacts and trying to mitigate these issues.

- First we finetuned the baseline ELECTRA-small model using SNLI train dataset split, recorded the loss and accuracy using SNLI test dataset split.

- Second, we revisited LIT paper and adjusted the script to accommodate the recent changes of ACE engine and ERG grammar. Then utilized multiple workstations to generate several gigabytes of SNLI ERG raw transfer data, post-processed them using various LIT predefined rules(such as it clefts, pasive, progressive, tense, including a new GPT-OSS-20B-Q4-GGUF top-k selection of best candidates. This reduced augmented SNLI dataset to several hundreds of megabytes.

- Third, we fine-tuned previous baseline model using these new augmented datasets and compared the loss and accuracy, both using standard SNLI test and new augmented test dataset.

- Fourth, we then fine tuned the model on anli, cnli-augmented and unli datasets.

- Fifth, we implemented a simplified version of dataset cartography method which can process 3 per-epoch checkpoints to calculate the confidence and variability and separated the augmented dataset into 3 categories: easy/hard/ambiguous. Then we retrained the baseline model with only ambigous subset and did further analysis and comparison between baseline model, augmented model, cartography model.

4 Implementation

We implemented a complete pipeline of training and evaluating scripts to characterize the dataset artifacts impacts and dataset augmentation improvements.

4.1 Hardware

Out of curiosity we tested our scripts on 3 popular GPU/APU systems.

- 1x Alienware R16 with Nvidia 4070 Ti GPU
- 2x Supermicro AMD Threadripper PRO with RX7900 / W7900 / RX9070 GPUs
- 1x Apple MacBookPro with M4 Pro APU

Even though the datasheet performance was similar, the real performance difference was huge and each system was better suited for different tasks. Before running the actual workload, we benchmarked 3 systems using final project (Bostrom et al., 2021) SNLI finetuning baseline as performance indicator. With default training batchsize of 8, Nvidia 4070 GPU obtained 26.9 it/s, whereas AMD GPU RX7900 33.9 it/s, W7900 34.2 it/s, RX9070 23.9 it/s, Apple M4 APU 26.1 it/s. As comparison, Intel Core i7-14700F CPU on Dell Alienware only achieved 4.3 it/s. We have to use workstation CPU to do LIT transfer and workstation GPU to do SNLI finetuning on ELECTRA-small. M4 APU also had a competitive edge, but due to the laptop form factor, we only did short trainings on it.

4.2 Dataset Generation and Selection

Generating Augmented Dataset is the main idea of dataset artifacts analysis and mitigation. We implemented a new pipeline based on prior work LIT (Li et al., 2020) and Cartography (Swayamdipta et al., 2020), and integrate every steps into NLP final project (Bostrom et al., 2021).

4.2.1 LIT

LIT (Li et al., 2020) is our foundation of SNLI dataset artifacts survey. We improved both transfer and post-processing script dramatically to better suit the latest software stack. And this work took almost half of our development time. The main drawback of LIT paper is the delicate nature of ACE engine and ERG grammar, we will discuss in details in the following subsections.

4.2.2 ACE

ACE (Packard, 2018) is a C++ Boost based text manipulation library used by LIT to generate sentences. It provides an optimized way of generating flexible grammatically correct SNLI sentences based on predefined grammars obtained from HPSG project (Copestake and Flickinger, 2000). Now it only supports Linux x86_64 and

MacOS arm64 binary. We tried to build ACE from source but it's very old and only supported Ubuntu 22.04 due to C++ Boost library limitation. We decided to use old version 0.9.31 binary to support old ERG 1214 on both Linux and Mac. The ACE generation phase for SNLI train dataset took 2 workstations more than 20 hours.

4.2.3 ERG

ERG (Flickinger et al., 2014) is an actively maintained and constantly evolved prebuilt English-only grammar rules collection. We tried to use the latest ERG build from 2025 and it failed miserably with LIT's manually constructed transfer grammar, especially for it-cleft and passive rules. Since version 2018 ERG changed a lot and it-cleft and passive grammar rules were no long valid, we had to fallback to old version 1214 attached on ACE website (Packard, 2018). We tried to adjust LIT it-cleft and passive grammar to both ERG version 2018 and 2025, but this was a far too demanding and error prone pure linguistic task for NLP class final project. We spent a week to carefully write LIT rules with ERG 2025 grammars, but the transfer dataset and training result of ERG 2025 was far worse than LIT's original ERG 1214 rules. Therefore this feature was abandoned and we stick to original LIT implementation.

4.2.4 GPT-OSS

LIT paper originally used GPT2 (Li et al., 2020) which was 5 years old and outdated. We modified the script and introduced modern GPT-OSS-20B model (OpenAI, 2025) to do the same work. In order to fit into our gaming GPU's limited VRAM, we have to use Q4_K_M GGUF (Gerganov, 2023a) quantization. However, the testing result was not very exciting. ERG generated SNLI premise and hypothesis sentences mostly achieved very similar perplexity score, which meant they were equally bad even though grammatically correct. We found top-k selection method actually degraded the accuracy compared to simply select the first candidate. Therefore we disabled this feature and put it to future use.

4.3 Dataset Augmentation

We improved final project's original run.py with new capability to read and write multiple jsonl files, in this way we can feed previous steps augmented ERG SNLI dataset to further finetuning baseline ELECTRA-small model. Because some NLI dataset might contain invalid labels out of range of SNLI's 0/1/2, we added another mapping function so that before training or testing all dataset would be sanitized. This feature was inspired by a GPU hardware exception due to out of range label value which could only be reproduced on AMD GPU but not Nvidia GPU. The signature was kernel name nll_loss_forward_reduce_cuda_kernel from rocdevice.cpp queue aborting with error HSA_STATUS_ERROR_EXCEPTION, an HSAIL operation resulted in a hardware exception, code 0x1016. This revealed the difference between CUDA and ROCM library in dealing with hardware exceptions.

4.4 Dataset Cartography

Dataset cartography (Swayamdipta et al., 2020) builds data maps by logging the model's behavior on each training example over epochs. For each training example i and epoch t, let:

- $p_t^{(i)}$ = model's probability of the gold label at epoch t for example i.

From these, define:

- **Confidence:**

$$\mu^{(i)} = \frac{1}{T}\sum_{t=1}^{T} p_t^{(i)}$$

  (average gold-label probability across training)
- **Variability:**

$$\sigma^{(i)} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(p_t^{(i)} - \mu^{(i)})^2}$$

  (std-dev of the gold-label probability)
- **Correctness:**
  fraction of epochs where the predicted label == gold label.

These three dimensions separate examples into:

- Easy-to-learn: high confidence, low variability (model gets them right early & consistently).
- Ambiguous: mid confidence, high variability (model flips back and forth; tends to help OOD generalization).
- Hard-to-learn: low confidence, low variability (often mislabeled/noisy).

Use these metrics computed in previous step to further separate and trim the augmented dataset:

- Remove the "easy" examples that mostly encode SNLI artifacts, we add –drop_easy_ratio parameter and preset the default value to 0.4
- Trim obviously noisy "hard" garbage, this is controlled by parameter –drop_hard_ratio with default value 0.1
- Emphasize the ambiguous and challenge region that tends to help OOD robustness, we relax –top_ambiguous_ratio default parameter to 0.7

For simplicity, we didn't reuse any source code from (Swayamdipta et al., 2020) but implement our own version of confidence and variation calculation and integrate them into this final project fine-tuning run.py (Bostrom et al., 2021). Therefore we can add do_cartography right after do_train and do_eval functions. For SNLI specifically, do_cartography with 3 epochs only needs less than 1 hour, much faster than do_train phase's 2 hours for standard SNLI train dataset or 4 hours for augmented ERG SNLI train dataset.

4.5 Experiments Result

We ran expriements for the following set of experiments

- Contrast Set it:o
- Contrast Set o:it
- Contrast Set p:o
- Contrast Set o:p
- Contrast Set ss:o
- Contrast Set o:ss
- Contrast Set n:n
- Contrast Set it:it
- Contrast Set f:p
- Contrast Set f:p+it
- Contrast Set f:ss
- CNLI
- ANLI

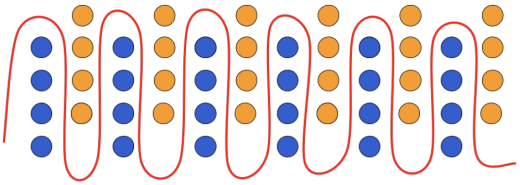
The table presented the interesting results of the curated datasets that had some interesting obersavaations and chararcteristics

5 Analysis

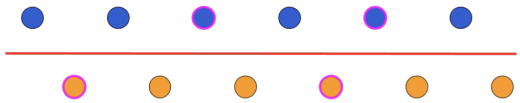
For the analysis of the Electra-small model we have have looked into using multiple different approaches: Contrast Sets, Adverseral Data Sets and Contrerfactually Generated and Augmented Data

Table 1: Summary of Total Errors

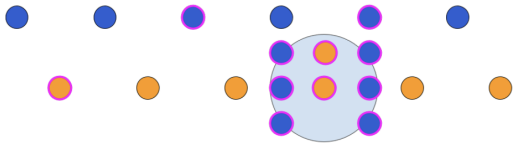
Experiment	Total	Errors	%
CNLI			
SNLI	9,842	1,069	10.86%
CNLI on SNLI Base	2,000	504	25.20%
CNLI on Tuned	2,000	401	20.05%
SNLI on Tuned	9,842	1,419	14.42%
ANLI Experiments			
SNLI	9,842	1,069	10.86%
ANLI on SNLI Base	3,200	2,248	70.25%
ANLI on ANLI Tuned	3,200	1,734	54.19%
SNLI on ANLI Tuned	9,842	1,655	16.82%
f:p Experiments			
SNLI	9,842	1,069	10.86%
f:p on SNLI Base	1,667	1,137	68.21%
SNLI Base (f:p)	9,842	3,728	37.88%
f:p on Tuned	1,667	120	7.20%
Future:Prog + its cleft			
SNLI	9,842	1,069	10.86%
f:p+its on SNLI Base	1,667	1,137	68.21%
SNLI Base (f:p+its)	9,842	3,728	37.88%
f:p+its on Tuned	1,667	120	7.20%
Future:Subswap Experiments			
SNLI	9,842	1,069	10.86%
(f:ss) on SNLI Base	1,683	1,231	73.14%
SNLI Base (f:ss)	9,842	3,858	39.20%
(f:ss) on Tuned	1,683	118	7.01%
it:o Experiments			
SNLI	9,842	1,069	10.86%
SNLI Base (it:o)	9,842	1,136	11.54%
Contrast (it:o) on SNLI Base	8,248	948	11.49%
Contrast (it:o) on Tuned	8,248	998	12.10%



(a) A two-dimensional dataset that requires a complex decision boundary to achieve high accuracy.



(b) If the same data distribution is instead sampled with systematic gaps (e.g., due to annotator bias), a simple decision boundary *can perform well on i.i.d. test data* (shown outlined in pink).



(c) Since filling in all gaps in the distribution is infeasible, a *contrast set* instead fills in a local ball around a test instance to evaluate the model's decision boundary.

Figure 2: An illustration of how contrast sets provide a more comprehensive model evaluation when datasets have systematic gaps.

Figure 2: Contrast Sets and Boundary Conditions

Sets. Each of these datasets provides different mechanisms that can be used to do further analysis independently.

5.1 Contrast Sets

Due to the nature of the model, to achieve high level of performance, the model tightly modeled boundaries2 for inference. The aim of the contrast sets to generate enough test cases that will cause the models to trip over these boundaries.

Perturbed test sets need only be large enough to yield reliable insights into model performance. However, as noted in the paper, based on the study (Gardner et al., 2020) this still requires about one person week of effort to generate 1000 examples.

NLI models have systematic gaps. With contrast sets, a tight collection of data (Test and Training) instances is created to target a systemic gap. This contrast set tests the local accuracy of the model's decision boundary relative to the pivot. This specific example highlights a failure in the model's simple boundary. We iterate this process across many pivots to build complete evaluation datasets. There are many interpretations of the gaps in the NLI model, and few examples can help to highlight, but this process is not universal, or scalable.

Contrast sets are not intentionally adversarial in nature, but can cause models to perform dramatically lower, especially when measured for evaluation consistency.

5.2 Linguistically Informed Transformations

Due to the time constraints for this final paper, lack of expertise in contrast set generations through perturbations, and the volume of data in the SNLI, even with only 10% failure rate against the SNLI dataset (with over 100K of training data). It seemed a monumental challenge to manually analyze and perturb enough data that could address these boundary conditions and attack the systemic gaps. To facilitate programmatic generation of Contrast sets Linguistically-Informed Transformations (Li et al., 2020) was leveraged for generating augmented SNLI data set, using ACE and ERG libraries that introduce linguistic perturbations such as passive voice, its cleft of the sentence, altering the tense of the sentences, and other linguistic transformations. LIT contains seven phenomenon-specific transformation rules for modifying the parse results and can be further extended; LIT also allows the composition of different transformation rules for complicated perturbations involving multiple linguistic phenomena. In comparison to the other methodologies for creating datasets the LIT methodology provides the following attributes

- Flexibility - Provides a simple template based approach
- Plausibility - Uses a single Lingustic backbone which creates a control for data plausibility
- Modularity - Since this has distinct modules for parsing, generation, transformation and post-selection, it lends itself for extension. This is a property we are leveraging
- Model Agnostic - this approach is limited to selecting the best string based on ERG representation, the archietecture and/or dataset do not influence the outcome.

5.3 Transform Process

The transform process parses the provided grammar in the SNLI dataset, then applies the requested linguistic transformations, producing the finally result which is added along with the original sentence in the augmented data set. the LIT process

does go through a process of de-duplication to ensure that the resulting set does not result in too many transformations.

The resulting sentences produced however, are not all stylistically recognizable as English language sentences, but can be lexically parsed to the provide the intended meaning and linguistic interpretation.

This process though programmatic is still time consuming, as it requires a lot of compute, but overall still feasible in the time we had for this project.

The still are limitations to this process from our perspective, as we are not able to identify or change labels, the golden labels which could be affected based on the combination of sentences used.

5.4 Applying LIT to SNLI

To apply the Linguistically informed transformation (Gardner et al., 2020) the code referenced utilizes the py-delphin library developed by DELIPH-IN. The py-delphin library utilizes the ERG library, which is a general-purpose computational grammar that, in combination with specialized processing tools, can map running English text to highly normalized logical-form representations of meaning. This library is used in conjunction with Answer Constraint Engine (ACE) runtime parser for parsing the sentence structure provided in the SNLI jsonl dataset.

For this transformation process, python libraries in addition with platform specific libraries for ace and erg binaries are required to be downloaded.

5.5 SNLI Data Augmentation

To perform the SNLI Data augmentation, the code from the github repository was downloaded. Using the transfer_example.py provided the provided library code was updated to ensure that a single example could perform all the linguistic transformations it was capable of.

To allow for parallel generation. the SNLI provided jsonl was split into 128 files to allow the generation to be run across 2 workstations in batches of multiple 32 (due to 16 cores 32 threads CPU restriction).

Even with this compute power, due to some unforeseen issues, this process still took over 48 hours to complete if only using a single workstation.

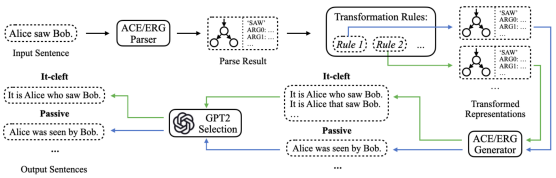


Figure 3: Linguistically Informed Transformation Process

This was still significantly faster than if we attempted to do linguistic transformations manually.

5.6 Augment Data Feature Extraction

For the feature extractions of perturbation we went the following set of transformations

- It clefts;Original
- Original;It clefts
- It clefts:It clefts
- Passive;Original
- Original;Passive
- Passive clefts:Passive
- SubjSwap;Original
- Original;Subj swap
- Sub Swap:Sub Swap
- Negation:Negation

These perturbations were chosen as these leave the label unchanged.

- Future Progressive;Progressive
- Future Progressive;Progressive + its cleft
- Future Progressive;Subjswap / its cleft

5.7 Analysis of Adversarial NLI systems

Analysis of Adversarial NLI systems (Chien and Kalita, 2019) show case a few heuristics that models

They have three categories of heuristics (each is a special case of the one before).

- Lexical overlap: The model is likely to answer *entailment* if the premise and hypothesis share a lot of words. It would trick bag-of-words (no word order) models.
- Subsequence: The hypothesis is a contiguous string of words from the premise. *The ball by the bed rolled.* $\nvdash$*The bed rolled.* It could confuse sequence models too.
- Constituent: The hypothesis is a syntactic constituent in the premise. *If the boys slept, they would not eat.* $\nvdash$*The boys slept.* It could confuse models that know about syntax.

All three heuristics involve the model thinking the answer is entailment when it is not, i.e. the non-entailment examples are the ones that contradict the heuristic

The paper has additional forms of analysis done on the Adversarial Challenge Set that can be used for further analysis, but using the Contrast Sets, these heuristics can be targeted.

5.8 Counterfactually Augmented Dataset

Counterfactually Augmented (Kaushik et al., 2020) Data devised a dataset creation mechanism in which humans counterfactually revise documents. To provide structure to the humans revising the data set, they were provided with the following set of instructions for the interventions

- Revise the letter to make it more positive
- Edit the second sentence so that it appears to contradict the first

These edits might be thought of as intervening on only those aspects of the text that are necessary to make the counterfactual label applicable. For the NLI task, these interventions were done either to the premise or the hypothesis without altering the golden label of the provided data.

Table 3: Analysis of edits performed by humans for NLI hypotheses. P denotes *Premise*, OH denotes *Original Hypothesis*, and NH denotes *New Hypothesis*.

Types of Revisions	Examples
Modifying/removing actions	**P:** A young dark-haired woman crouches on the banks of a river while washing dishes. **OH:** A woman washes dishes in the river **while camping**. (Neutral) **NH:** A woman washes dishes in the river. (Entailment)
Substituting entities	**P:** Students are inside of a lecture hall. **OH:** Students are **indoors**. (Entailment) **NH:** Students are **on the soccer field**. (Contradiction)
Adding details to entities	**P:** An older man with glasses raises his eyebrows in surprise. **OH:** The man **has no glasses**. (Contradiction) **NH:** The man **wears bifocals**. (Neutral)
Inserting relationships	**P:** A blond woman speaking to a brunette woman with her arms crossed. **OH:** A woman is talking to **another woman**. (Entailment) **NH:** A woman is talking to **a family member**. (Neutral)
Numerical modifications	**P:** Several farmers bent over working on the fields while lady with a baby and four other children accompany them. **OH:** The lady has **three** children. (Contradiction) **NH:** The lady has **many** children. (Entailment)
Using/Removing negation	**P:** An older man with glasses raises his eyebrows in surprise. **OH:** The man **has no** glasses. (Contradiction) **NH:** The man **wears** glasses. (Entailment)
Unrelated hypothesis	**P:** A female athlete in crimson top and dark blue shorts is running on the street. **OH:** A woman is **sitting on** a white couch. (Contradiction) **NH:** A woman **owns** a white couch. (Neutral)

Figure 4: Prominent Examples of Counterfactually Augmented data for NLI

Refinement This dataset generation can be further refined by using large language models to better format the sentence to make it more human-like for better inference characteristics to help evaluate the model in more realistic human scenarios

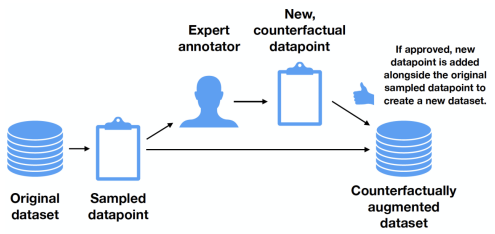


Figure 1: Pipeline for collecting and leveraging counterfactually-altered data

Figure 5: Pipeline for collecting and leveraging counterfactually-altered data

Premise ⇝ Hypothesis	SNLI	*u*-SNLI	Predicted
A man perched on a row of aquariums is using a net to scoop a fish from another aquarium. ⇝ A man is standing by the aquariums.	ENT	1.0	0.119
A man and woman are drinking at a bar. ⇝ A couple is out on a date.	NEU	0.755	0.377
Couple walking on the beach. ⇝ The couple are holding hands.	NEU	0.808	0.308
An elderly woman crafts a design on a loom. ⇝ The woman is a seamstress.	NEU	0.923	0.197
Two girls riding an amusement park ride. ⇝ The two girls are screaming.	NEU	0.909	0.075
A man and woman sit at a cluttered table. ⇝ The table is neat and clean.	CON	4.91×10^{-4}	0.262
A race car sits in the pits. ⇝ The car is going fast.	CON	2.88×10^{-7}	0.724
A guy is standing in front of a toilet with a coffee cup in one hand and a toilet brush in the other. ⇝ A man is attempting to brew coffee.	CON	8.32×10^{-6}	0.504

Figure 6: Samples of UNLI data diverted from BERT

Inference With the new Contrast Sets generated with can be used to evaluate whether a model's prediction changes or not when input is altered in a salient detailed way.

5.9 Cartography Data Map Analysis

After dataset cartography filtering, we obtained one third of the original augmented dataset as ambiguous "gold" subset. We also reduced the training time by almost 62% to achieve similar accuracy result. The only problem is ERG 2025 result with modified grammar, our new itcleft and passive rules cannot generate enough data compared to LIT's original rules, thus it failed standard SNLI test with only 36% accuracy. As comparison, baseline model cannot pass LIT augmented test and accuracy dropped from 89% to 82% because it did more wrong predictions to slightly revised test set. Table 2 showed the detailed accuracy. From Figure 7 we knew intuitively why accuracy was similar between ERG generated dataset and Cartography filtered dataset. The ambiguous data were so few that most of SNLI dataset was either easy or hard. That also indicated 10 years old SNLI 1.0 dataset was not good enough for modern finetuning tasks.

6 Discussion

Hardware Compatibility Issues We were working to the tune the model across multiple machines that had different hardware and software

Model	STD acc	LIT acc	DS Line Ct
Baseline	89.39%	82.24%	550,152
LIT 1214	89.76%	88.75%	7,699,395
ERG 2025	35.84%	88.18%	425,941
Cartography	89.33%	86.73%	2,142,901

Table 2: Performance comparison across different models. Note the trade-off between standard SNLI test accuracy and LIT augmented SNLI test accuracy and the vast line count difference between different dataset.

characteristics that seem to have contributed to the erratic nature of the results. Underlying difference in python libraries based on Intel CPU, Apple M4, Nvidia and AMD GPU, these could be affecting the experiments. A more controlled set of hardware to be used for running the experiments would provide a more stable base for producing results.

Device Form Factor Based on the device the batch sizes seem to affect the result. On the Mac M4 even the baseline dataset with base ELECTRA-Small model was affected and produced only 60% accuracy when using non default device batch size of 8. This did not seem to be an issue when using other hardware.

Incorrect Methodology The methodology we followed was as follows, first tune the base SNLI model, this was done with the provided run.py file with not additional changes to remove bias, and run the this and save the results, and checkpoints for our usage. For each of the experiments we then used the saved trained SNLI model as the base model and further ran training on the specially curated dataset for the experiment. Then we ran the evaluation against the SNLI dataset, subsequently against the test dataset for the specially curated test data that was created for the experiment. Based on results in some of the experiments, SNLI model performed much worse when tested against the SNLI test data, the expectation was that there would be drops in inference, but some of the drops were hard to explain.

LIT Data Creation The LIT data creation was slow, so to improve the speed of the LIT pipeline some shortcuts were taken to limit the kind of outputs the ace/erg combination would produce. This shortcut ended up hampering the kinds of perturbations that we were able to create. In the first iteration we included only the present tense, this limits the inference to being unchanged, thus not meaningfully perturbing the results. However if the even onl future tense are included in the experiment mix, it showed issues in the ELECTRA-small model that could have been finetuned to improve.

7 Conclusion

Contrast Sets The Contrast set experiments resulted were inconclusive, we did get an overall model to show minor improvements, but this result did not seem to be meaningful.

LIT Data Transform The LIT Transformation Pipeline worked though slowly, to produce data that could be used for creating contrast datasets. Given more experience with this, more robust set of experiments could be created and analyzed.

Data Cartography Using the Data Cartography results for trimming did produce a reduced set of training data that looked promising for fine tuning the model. In the initial stages, this only provided a small improvement. The reason was because cartography did trim the dataset, analogue to jpeg compression, it removed useless data and reduced training time by several times. If computation resource was fixed, cartography saved resource for more useful dataset, thus improve the accuracy. Unfortunately, that's not the case for our final project because we only had a simple SNLI 1.0 dataset.

References

Kaj Bostrom, Jifan Chen, and Greg Durrett. 2021. fp-dataset-artifacts: Final project starter code for nlp classes at UT austin. Code available at https://github.com/gregdurrett/fp-dataset-artifacts.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Van Durme. 2019. Uncertain natural language inference.

Tiffany Chien and Jugal Kalita. 2019. Adversarial analysis of natural language inference systems.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In

Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00), Athens, Greece. European Language Resources Association (ELRA).

Dan Flickinger, Emily M. Bender, and Stephan Oepen. 2014. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 875–881. European Language Resources Association (ELRA).

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets.

Georgi Gerganov. 2023a. GGUF: Gpt-generated unified format. Accessed: 2025-11-24.

Georgi Gerganov. 2023b. llama.cpp: Port of facebook's llama model in c/c++. Accessed: 2025-11-24.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data.

Chuanrong Li, Lin Shengshuo, Leo Z. Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (lit): A method for automatically generating contrast sets. Code available at https://github.com/leo-liuzy/LIT_auto-gen-contrast-set.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data?

OpenAI. 2025. gpt-oss-20b model card.

Woodley Packard. 2018. ACE: The answer constraint engine. Last Accessed: 2025-11-19.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. Code available at https://github.com/allenai/cartography.

A Supplemental Material

Please review and check the detailed LIT ACE 0.9.31 ERG-1214 Generated Augmentation SNLI Dataset Cartography Data Map as Figure 7

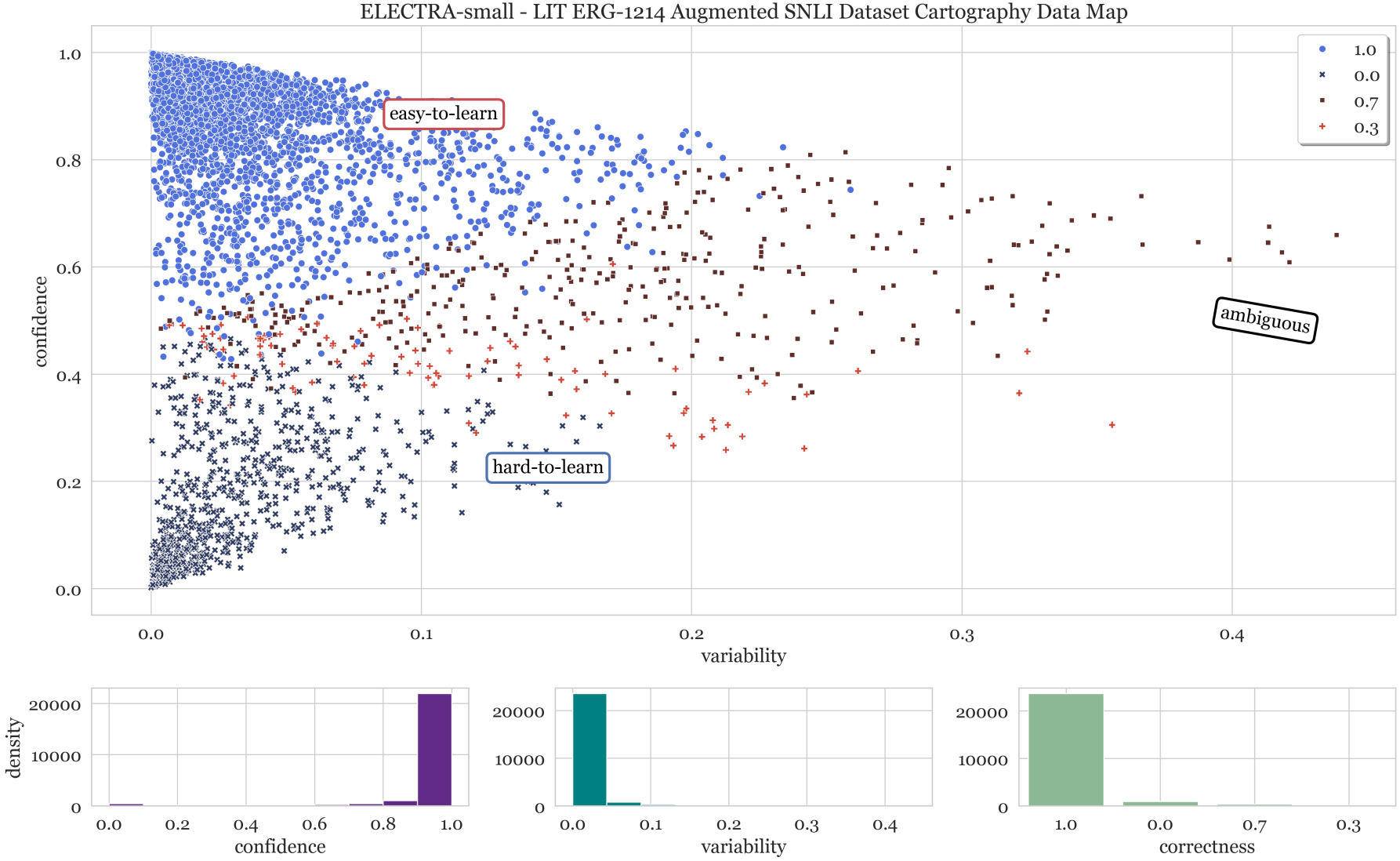


Figure 7: LIT Original ERG-1214 Dataset Cartography Data Map