Multi-Set Inoculation: Assessing Language Model Robustness Across Multiple Challenge Sets

Anonymous ACL submission

Abstract

Language models, characterized by their black-box nature, often hallucinate and display sensitivity to input perturbations, causing concerns about trust. To enhance trust, it is imperative to gain a comprehensive understanding of the model's failure modes and develop effective strategies to improve their performance. In this study, we introduce a framework designed to examine how input perturbations affect language models across various scales, including pre-trained models and large language models (LLMs). Utilizing fine-tuning, we enhance the model's robustness to input perturbations. Additionally, we investigate whether exposure to one perturbation enhances or diminishes the model's performance with respect to other perturbations. To address robustness against multiple perturbations, we present three distinct fine-tuning strategies. We also extend the applicability of our framework to LLMs through a chain of thought (CoT) prompting approach with exemplars. Our framework is applied to the Tabular-NLI task, demonstrating that the proposed strategies effectively train the model to handle various perturbations without compromising accuracy on an original set.

1 Introduction

Language models (LMs), which have become increasingly integrated into various aspects of daily lives, hold immense potential to revolutionize how we interact with technology. Their ubiquity underscores the importance of thoroughly examining their robustness and generalizability, which will be instrumental in fostering trust among users. One notable challenge is their sensitivity to even slight changes in input. For instance, while a human can easily interpret and understand a statement regardless of minor alterations, LMs struggle (Wang et al., 2023; Nie et al., 2020). This inconsistency becomes notably apparent when minor perturbations to the input, which do not inherently modify the underlying meaning, result in a marked decline in the performance of the model (Shankarampeta et al., 2022; Glockner et al., 2018). An example of such perturbations for the task of tabular inference (Gupta et al., 2020), is illustrated in Table 1.

Case Closed	
Written	Takahiro Arai
Publish	Shogakukan
Eng. Publish	SG Shogakukan Asia
Demographic	Shonen
Magazine	Weekly Shonen Sunday
Orig. Run	May 9, 2018 - present
Volumes	2 (List of volumes)

H_1: Takahiro Arai wrote 'Case Closed' comic series. (E)
H_1': Takahiro Arai wotte 'Case Closed' comci series. (E)
H_2: 'Case Closed' is a long-term comic series. (E)
H_2': 'Case Closed' isn't a long-term comic series. (C)
H_3: 'Case Closed' became the anime Detective Conan (N)
H_3': Detective Conan is 'Case Closed' anime version. (N)
H_4: 'Case Closed' has run over 5 years.(E)
H_4': 'Case Closed' has run over 10 years.(C)
H_5: Shogakukan Asia published 'Case Closed' (Eng). (E)
H_5': Shogakukan UK published 'Case Closed' (Eng). (C)

Table 1: Tabular premise and hypotheses example from INFOTABS (Gupta et al., 2020). Original (H_1-H_5) and perturbed hypotheses (H_1'-H_5') represent character, negation, paraphrasing, numeric, and location perturbations. Labeled as E, C, or N. **Bold** entries are keys; adjacent entries are their values.

Addressing these sensitivities to input perturbation is crucial for the advancement and reliability of LMs in real-world applications. Empirical evidence supports the effectiveness of fine-tuning models using perturbed input samples from challenge sets (Jiang et al., 2022; Fursov et al., 2021). For instance, Wang et al. (2020); Liu et al. (2019a) showcased that a pre-trained language model (PLM) utilizing Masked Language Modeling (MLM) and trained for a specific NLP task becomes significantly robust to input perturbations when further fine-tuned using a small set of challenge examples. However, the ability of these models to generalize across different types of perturbations is still a subject of investigation (Liu et al., 2020). The implications of fine-tuning

a model on a particular challenge set, especially concerning its impact on handling other perturbations, warrant further exploration. For example, refer to Figure 2, it remains unknown whether a model's enhanced robustness to character perturbations, post-fine-tuning, implies its competence in addressing challenges from distinct categories, such as those involving paraphrasing.

In this study, we address this challenge, seeking to answer the following two questions: *How does fine-tuning a model on one challenge set affect performance on others? Is it possible to guarantee consistent robustness across multiple distinct challenge sets?* In particular, we extend the *single-set inoculation* approach of Liu et al. (2019a), to a more generic multi-sets context, which we refer to as *multi-set inoculation*. First, we introduce a comprehensive framework designed to address the complexities of multi-set inoculation. To the best of our knowledge, we are the first to introduce and extensively study this challenge.

Our framework is adept at handling both (a) transformer-based pre-trained models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) [1], which are amenable to direct fine-tuning, and (b) large generative language models such as gpt-3.5-turbo(GPT-3.5)(Brown et al., 2020), GPT-4, and LLaMA, LLaMA-2 (Touvron et al., 2023), Flan-T5(Chung et al., 2022; Kanakarajan and Sankarasubbu, 2023) etc, which can't be fine-tuned freely. For these generative models, we leverage the few-shot Chain of Thought Wei et al. (2023) technique at inference, serving as an alternative parallel to traditional fine-tuning. This methodology circumvents the computational intricacies inherent in the fine-tuning of LLMs. It proficiently manages the tuning of a multitude of model parameters using a limited constrained set of training samples. To the best of our knowledge, we are the first to study Inoculation with LLM, prior studies (Liu et al., 2019b; Wang et al., 2021a; Liu et al., 2019a) limit to a more traditional BERT style model. Under our framework, we explore three distinct mulitple set fine-tuning methods, each aiming to strengthen model resilience across various challenge sets with different perturbations. Our study made the following contributions:

- We introduce the problem of "Multi-set Inoculation" which examines the implications of fine-tuning across multiple challenge sets.
- We proposed a framework to study the multi-set inoculation. Our framework encompasses traditional PLMs like RoBERTa, as well as LLMs including GPT-3.5 and LLaMA-2 etc.
- Using tabular NLI as our case study, we evaluated three distinct multi-set fine-tuning approaches which hold potential to ensure concurrent robustness to multiple challenge sets, i.e., several forms of input perturbations.

[1] Anonymous link code/ dataset of our framework.

2 Our Proposed Framework

In this section, we detail our framework for Multiset Inoculation. We evaluate the robustness of the model by subjecting it to different input perturbations. Subsequently, we propose our multiset fine-tuning techniques, which improve the model's performance on multiple challenging datasets. Figure 1 represents a high-level flowchart of our proposed framework.

Terminology. Given a PLM model, M, fine-tuned on the original (unperturbed) training set $\mathrm{O} = \{(x_i, y_i)\}_{i=1}^{N}$ for NLP task T. Let $\mathrm{O}_{S_j} = \{(x_i, y_i)\}_{i=1}^{n_j}$ be a sub-sample set S_j of the original training set O, where $n_j \ll N$. Let π_j represent an input perturbation applied to O_{S_j} to produce the challenge set $\Pi_j^{S_j} = \{\pi_j(x_i), \pi_j(y_i)\}_{i=1}^{n_j}$. If we have m distinct perturbations $= \{\pi_j\}_{j=1}^{m}$ applied to m sample subsets $\{S_j\}_{j=1}^{m}$ from the original set O, the final perturbation challenge set becomes $\{\Pi_j^{S_j}\}_{j=1}^{m}$. We use P_j as a shorthand for the final perturbation set $\Pi_j^{S_j}$.

We assess the performance of M using evaluation metric Θ across held out challenge sets Q_j for $j = 1, \ldots, m$. Q_j is test set for perturbation P_j, yielding the performance θ_j for each respective challenge set for perturbation P_j. Importantly, these metrics Θ serve as an evaluation metric, enabling a consistent comparison of model M's efficacy on diverse challenge sets.

2.1 Adopting MultiModel Uniset Inoculation

Drawing from the insights presented in our introduction, we employ challenge sets, denoted as P_j, as tools for refining a more robust model (Liu et al., 2019a). Specifically, we fine-tune our PLM model using K samples extracted from a challenge set P_j. This fine-tuning across different P_j sets results in an array of robust models, each designated as RM_j. We subsequently evaluate these models' performances across held-out challenge test sets, Q_j

for every $j \in N$. This evaluation probes the efficacy of inoculating models on a singular set in enhancing—or possibly undermining—performance on both test sets and different challenge sets. While this *multi-model single set* framework generates multiple robust models, a clear downside emerges: as the variety of perturbation types grows, managing multiple models becomes impractical.

2.2 Adopting UniModel Multiset Inoculation

The proposed methodology involves fine-tuning the PLM using samples drawn from every challenge set, aiming to cultivate a universally robust model, denoted as RM, based on the original model M. We put forth three distinct fine-tuning techniques:

- **Sequential** (SEQ)**:** This technique systematically fine-tunes the model by utilizing K samples from each challenge set P_j, aiming to evolve the model into RM. The fine-tuning proceeds in an order/sequence (specified by ORDER), which may be set at random or determined through particular criteria.
- **Mixed-Training** (MIX)**:** In this strategy, a composite dataset, termed P_M, is fashioned by randomly selecting K samples from all challenge sets, P_j. The cumulative size of P_M is given as mK, with K being a consistent sample size across all sets, P_j. Subsequently, the model M is fine-tuned using the aggregated P_M to give rise to RM. In our implementation, we adopt a uniform, random sampling approach, symbolized as Υ.
- **Dynamic Mix-Training** (DYNMIX)**:** This approach mirrors Mixed-Training but introduces variability in sample sizes across different challenge sets, denoted as K_1, K_2, and so on. Additionally, the sampling methodology Υ can be tailored for each perturbation challenge set ($\Upsilon_j \forall$ P_j).

Given that all three outlined techniques revolve around data sampling and culminate in a singular robust model RM, we refer this as the *Uni-model multi-set* paradigm.

2.3 Inoculation via. Prompting for LLM

Fine-tuning LLMs on challenge sets is costly. In contrast, prompt tuning is quicker and more effective for many NLP tasks (Shin et al., 2023). We design a prompt encapsulating the "task" description. We also add illustrative instances (as "exemplars")

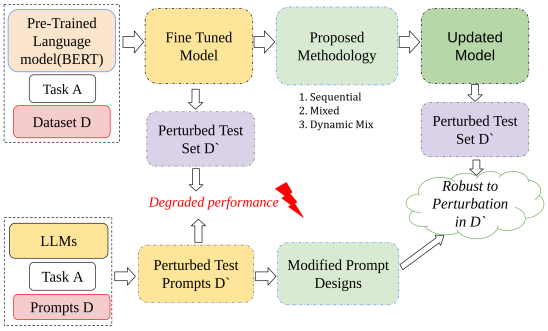

Figure 1: **Multi-Set Inoculation Framework.** High-level flowchart describing the proposed frameworks for PLMs (via fine-tuning) and LLMs (via prompt design).

from original sets (O) which serve as main guiding posts (a.k.a few shot). Each exemplar is enriched with a rationale, mirroring a *"chain of thought"* CoT prompting (Wei et al., 2023).

Original Prompt (OP)**.** This allows us to investigate the effectiveness of the perturbations π_j on LLMs. This is a baseline for LLM performance under input perturbations. Here, we consider two variants of LLM prompting:

- **Zero-shot** (OP_{ZS}). We create a prompt template consisting of only the description of the task, without any exemplars or reasoning chains. This serves as our baseline LLM.
- **Few-shot with CoT** (OP_{CoT})**.** Here, we consider NLI task description along with few shot exemplars taken from the original set O their reasoning chains a.k.a. chain of thoughts.

Single Exemplars Multiple Prompts (SEMP) : For each perturbation type, denoted as π_j, we construct a prompt that combines the task description, respective perturbation description, and exemplars from O and P_j. The exemplars are accompanied by corresponding labels and a reasoning chain (CoT). This results in multiple prompts, each tailored to a specific perturbation π_j. We call this approach *single exemplars multiple-prompts*, similar to *multi-model single set* framework in sec. 2.1.

Multiple Exemplars Single Prompt (MESP) **:** Here, we consider descriptions and exemplars of all perturbations ($\forall \pi_j$) in a single prompt. We create a prompt by combining multiple exemplars corresponding to each perturbation π_j, sampled from P_j. Here, the prompt contains the task description, a description of all perturbations, and exemplars from the original set O and each of the challenge sets ($\forall_j$ P_j). Given token length constraints, we tradeoff between the detail of perturbation descriptions and the number of perturbation exemplars:

- **Mixed-Prompting-Instructional** (MESP_{MPI}) : In this prompt, we emphasize the perturbation description while reducing the number of exemplars. Still, we ensure at least one exemplar is sampled from each perturbation.
- **Mixed-Prompting-Exemplar** (MESP_{MPE}): In this prompt, we prioritize more perturbation exemplars and condense each perturbation's description. However, we ensure that each brief description captures the core logic of the perturbation.

3 Case Study on Tabular Inference

For our framework evaluation, we focus on semi-structured tabular inference task as a case study.

Original Dataset (O)**.** We utilize the tabular-NLI dataset, INFOTABS, cited from (Gupta et al., 2020), along with its adversarial perturbations as detailed in Shankarampeta et al., 2022. The INFOTABS dataset features a wide range of table domains, categories, and keys, covering various entity types and forms. It includes three test splits: α_1 (original test set), α_2 (adversarial set), and α_3 (zero-shot or out-of-domain set), detailed in Gupta et al., 2020.

Class	Key	Type
char	C	Character-level perturbation
neg	N	Negation-type perturbation
num	M	Numeric perturbation
loc	L	Location perturbation
stan	S	Paraphrasing as a perturbation

Table 2: **Perturbation Types:** We use 5 distinct perturbations as challenge sets for Tabular-NLI.

Perturb Challenge Datasets (P, Q)**.** Our dataset incorporates perturbations from Shankarampeta et al., 2022, enhanced using tools such as TextAttack (Morris et al., 2020) and NLP Checklist (Ribeiro et al., 2020), alongside manual adjustments. Each perturbation specifically targets the hypothesis of an input sample. For every perturbation type, we curate challenge sets of up to 1,500 samples. Only those samples that are pertinent post-perturbation are selected. When the number of such samples exceeds 1500, we narrow down to the most diverse 1500 using Fixed-Size Determinantal Point Processes (k-DPPs) (Kulesza and Taskar, 2011). Perturbations pertinent to the Tabular-NLI tasks in our framework are delineated in Table 2.

Train/Test for BERT's Models. For any perturbation type, we represent Q_j consisting of 1000 examples for testing and P_j consisting of 500 examples for fine-tuning. We define the union of all challenge test sets as $\text{Q} = \{\cup_j^m \text{Q}_j\}$ and the corresponding training set as $\text{P} = \{\cup_j^m \text{P}_j\}$.

Train/Test for LLMs. As LLMs inference is costly we limit our evaluations to 300 samples from Q_j, which we named $\text{R}_j (\text{R}_j \subset \text{Q}_j)$. R_j is used to describe the challenge set comprising of premise and perturbed hypothesis pairs and R'_j contains the same premise along with the corresponding unperturbed hypothesis as pairs. Specifically, $\text{R}_j = \{\text{P}: \text{P}= \Pi_j^{-1}(x), x \in \text{R}_j\}$. We define union of all evaluation sets as $\text{R} = \{\cup_j^m \text{R}_j\}$. Demonstrations for prompts are sampled from training sets P_j and P'_j, where P'_j is the set containing the original unperturbed hypothesis and premise pairs for the examples in P_j.

Table Representation. Adapting from Neeraja et al., 2021, we used alignment methods (Yadav et al., 2020) to remove distracting rows (DRR). We consider the top-8 rows for table representation as a premise (DRR@8). This facilitates a more accurate representation by evidence grounding of the premise for information pertinent to the hypothesis that requires labeling.

Evaluation Metric. We use accuracy which is equivalent to the micro-f1 score for the NLI task where the label for each example can be only one of entailment **E**, contradiction **C**, neutral **N**. The improvement over the multi-challenge sets is considered by taking the average of the improved performance over each challenge set Q_{π_j} and this is used as the score(μ) for multi-perturbation setting.

Implementation and hyperparameter details for all experiments are mentioned in Appendix A.2.

3.1 Modeling: Fine-tuning Bert's Model

We use ROBERTA-LARGE (Liu et al., 2019b) as the baseline model (Wolf et al., 2020) fine-tuned on INFOTABS train set. This baseline model is henceforth referred to as $\text{ROBERTA}_{\text{INTA}}$.

We test the baseline model (and all other evaluations) on both O and Q, by testing on Q we attempt to demonstrate the effect of the different perturbations $\pi_C, \pi_N, \pi_M, \pi_L, \pi_S$ on $\text{ROBERTA}_{\text{INTA}}$.

MultiModel Uniset Inoculation $\text{ROBERTA}_{\text{INTA}}$ is further fine-tuned on different types of challenge sets(P_j), resulting in multiple robust models.

UniModel Multiset Inoculation. Sequential (SEQ)**:** We perform sequential fine-tuning of

RoBERTa$_{\text{INTA}}$ across various challenge sets. The training order (ORDER) for fine-tuning is based on average baseline model performance across challenge sets. Our sequencing strategy aims to minimize the potential for catastrophic forgetting (Kirkpatrick et al., 2017; Goodfellow et al., 2013) induced by subsequent fine-tuning on challenge sets.

Mixed-Training (MIX): Here, the RoBERTa$_{\text{INTA}}$ is fine-tuned samples obtained by mixing K instances drawn from each of the challenge sets P_M, P_N, P_L, P_C, P_S. Here, K is an hyper-parameters, set equal to 500 examples, as discussed in section 3.1.

Dynamic Mix-Training (DYNMIX): This is similar to MIX, except the number of samples drawn from each of the challenge sets is different. The number of samples is determined by the inverse of the baseline performance for RoBERTa$_{\text{INTA}}$ for challenge sets P_j.

3.2 Modeling: LLM Prompting

We used GPT-3.5 with low temperature of 0.3, LLaMA-2 after quantization using QLoRA (Dettmers et al., 2023), and Flan-T5 series. We develop methodologies for LLMs that rely solely on prompting and exclude fine-tuning(except for GPT-3.5 where we also report fine-tuning results). The LLM prompt design for our experiments, is detailed in Table 3, comprises five sections, with the Demonstration section being optional. The prompt first defines the NLI task, explaining entailment, contradiction, and neutrality. It then describes potential sentence perturbations and instructs the model to rely solely on common knowledge provided in the premise. Finally, it outlines the answer format, followed by an optional number of demonstrations.

Broad Prompt Template	
NLI Task Explanation	In this task, we will ask you to make an inference about the information presented as the premise. We will show you a premise and a hypothesis...
Perturbation Awareness	The concept of numeric and character typos in questions is important for maintaining the integrity and meaning of a sentence...
Description of Limitation	It is very important and critical that you do not use information other than the premise that you may know if you believe that it is not generally known...
Answering	(Restriction for Answering) Answer with an explanation in the following format, restricting the answer to only one of the following: "yes" or "no" or "it is not possible to tell" + Answering Format
Demonstrations	Demonstrations from different sets with reasoning (CoT).

Table 3: Prompt Structure used in LLMs

Original Prompt (OP). This is the original prompt zero shot (OP$_{\text{ZS}}$) setting with NLI task description. In CoT setting (OP$_{\text{CoT}}$), we define our few shot setting, where exemplars are sampled from original training dataset O.

Single Exemplars Multiple Prompts (SEMP). For a designated perturbation π_j from the set $\{\pi_C, \pi_N, \pi_M, \pi_L, \pi_S\}$, our prompts integrate the NLI task outline, a brief on the perturbation π_j, and its Chain of Thought (CoT) exemplars sourced from the respective challenge set P_j.

Multiple Exemplars Single Prompt (MESP). These prompts contain NLI task description, description of all perturbations $\pi_j \in \{\pi_C, \pi_N, \pi_M, \pi_L, \pi_S\}$ and exemplars sampled from each challenge set $P_j \in \{P_M, P_N, P_L, P_C, P_S\}$. Here , we consider two different prompts settings MESP$_{\text{MPI}}$ and MESP$_{\text{MPE}}$, as described earlier in section 2.3.

4 Results and Analysis

Our experiments answer the following questions:-

- Are input perturbations challenging for LMs?
- How does multi-model single-set inoculation and perturbation-specific fine-tuning affect LLM performance across perturbations?
- Are perturbation descriptions, exemplars, and Chain of Thought (CoT) prompting beneficial for LLM effectiveness?
- How does UniModel Multi-Set Inoculation fare against single-set inoculation?
- Which is more vital for LLM prompting: descriptions quality or exemplars quantity?

4.1 Results: Bert Style Models

MultiModel Uniset Inoculation. The baseline performance of RoBERTa$_{\text{INTA}}$ original and challenge sets is shown in Table 4. We also report the performance after fine-tuning each challenge set in the same table.

	Original Test Sets			Challenge Test Sets				
Train/ Test	α_1	α_2	α_3	**char**	**neg**	**num**	**loc**	**stan**
baseline	72.72	**64.83**	62.33	57.30	46.90	67.20	70.20	67.10
char	**75.28**	63.83	**63.33**	**59.20**	43.70	64.30	66.00	**68.30**
neg	66.94	64.56	58.06	52.80	**71.90**	69.60	69.70	62.40
num	62.06	60.83	52.50	47.30	49.60	**85.40**	83.00	57.60
loc	55.78	58.67	49.67	47.40	53.90	84.60	**86.10**	53.50
stan	73.56	62.61	60.44	58.30	40.80	70.30	67.80	66.80

Table 4: **Multi-model Uniset Inoculation:** RoBERTa$_{\text{INTA}}$ when fine-tuned on one of the challenge sets (P_j), but tested on all challenge sets (Q_j) with number of sample used equal 500.

Analysis. (a.) Baseline performance of RoBERTa$_{\text{INTA}}$ on challenge sets is notably lower

than on original sets, emphasizing PLMs' vulnerability to input perturbations. (b.) Fine-tuning via single-set inoculation significantly bolsters the model against specific perturbations, improving negation accuracy by +25 points from baseline. (c.) Despite fine-tuning, the model's robustness to paraphrasing remains largely unchanged. (d.) While the fine-tuned model excels against specific perturbations, it struggles with others. Interestingly, character perturbations inadvertently boost its proficiency in challenges like paraphrasing. (e.) Inoculation effects vary: character set inoculation enhances performance on original test sets, while number and location decrease performance in both original and challenge test sets.

UniModel Multiset Inoculation. We present results on Sequential training (SEQ), Mixed Training (MIX), and Dynamic Mixed Training (DYNMIX) in Table 5, 6, 7 respectively.

SEQ. Table 5 presents the results using Sequential Training (SEQ). The method trains ROBERTA$_{\text{INTA}}$ on varied challenge sets in distinct sequences. For instance, ORDER MNLCS with K samples implies training sequentially on subsets of $\{P_M, P_N, P_L, P_C, P_S\}$ of size K. This is denoted as SEQ$_{\text{MNLCS}}$.

	Original Test Sets			Challenge Test Sets					
Type	α_1	α_2	α_3	**char**	**neg**	**num**	**loc**	**stan**	μ
baseline	72.72	**64.83**	**62.33**	**57.30**	46.90	67.20	70.20	67.10	-
COL-ASC	61.67	60.94	50.11	48.80	54.60	**85.40**	85.40	56.60	**4.42**
COL-DSC	**74.67**	62.72	60.44	58.90	**57.30**	56.10	65.30	**68.00**	-0.62
ROW-ASC	55.00	58.11	47.22	46.80	50.90	84.50	**85.90**	51.30	2.14
ROW-DSC	73.44	63.39	57.44	56.50	45.10	60.00	71.60	65.80	-1.94

Table 5: **SEQ Results:** ROBERTA$_{\text{INTA}}$ Sequential Training SEQ with 500 samples from each P_j. Here, COL-ASC: CSNLM, COL-DSC: MLNSC, ROW-ASC: SCNML, ROW-DSC: LMNCS. μ is the average improvement.

Terminology. To define the sequence we consider (a.) *Column Wise Average.* This configuration assesses the aggregate impact of fine-tuning across all perturbations on each individual perturbation., (b.) *Row Wise Average.* This configuration evaluates the aggregate impact of fine-tuning on an individual perturbation against all other perturbations.

(a.) *Column Wise Average.* The column-wise average (COL) for a given perturbation π_d is the average performance improvement over the baseline on Q_j (Table 4) for models fine-tuned on all other perturbation P_j, FOR $j \neq d$ (except itself).

(b.) *Row Wise Average.* The row-wise average (ROW) for a given perturbation π_d is the average performance improvement over the baseline performance (Table 4) for the model fine-tuned on P_d on other challenge dataset sets Q_j, FOR $j \neq d$.

We compute both COL and ROW values for each perturbation. By sorting these values, we derive sequences in ascending and descending order, yielding the COL-ASC, COL-DSC, ROW-ASC, ROW-DSC as the ORDER sequences[2].

Analysis. Sequential training introduces the forgetting issue (He et al., 2021; Chen et al., 2020a), where models forget sets trained on earlier in the sequence. (a.) With column-wise averages, we capture how easy a perturbation π_j is to learn by fine-tuning on other perturbations by testing improvement in accuracy on set Q_j. Therefore in the ORDER COL-ASC, an "easier" perturbation appears later and hence improves the average performance. (b.) With row-wise averages, we capture how much fine-tuning on P_j improves the overall performance of other perturbation types. Hence, in the ORDER ROW-ASC with samples from P_j wherein π_j has a higher score appearing later, benefit other better perturbation more effectively.

MIX. Table 6 presents the outcomes from multi-set inoculation using mixed training.

	Original Sets			Challenge Sets					
K	α_1	α_2	α_3	**char**	**neg**	**num**	**loc**	**stan**	μ
baseline	**72.72**	64.83	**62.33**	**57.30**	46.90	67.20	70.20	**67.10**	-
100	70.40	**65.16**	59.48	56.00	58.48	78.78	78.50	66.04	5.82
200	70.42	65.06	59.21	56.86	59.50	80.94	80.36	64.68	6.73
300	71.92	64.54	59.49	56.50	61.30	81.22	79.68	65.12	7.02
400	72.11	64.48	59.78	56.58	63.70	81.60	80.38	64.64	7.64
500	72.62	64.34	59.20	56.98	**66.06**	**82.02**	**80.52**	65.64	**8.50**

Table 6: **MIX Results :** ROBERTA$_{\text{INTA}}$ fine-tuned on mixed samples from different perturbation sets P_j.

Analysis. Models trained via mixed training outperform those from SEQ. As we increase the number of samples for fine-tuning, we notice consistent gains across most challenge sets and original test sets. The most prominent improvements are seen in the negation and location sets. While there's a minor performance dip in some original and challenge sets, it's less pronounced compared to results from single-set inoculation and SEQ.

DYNMIX. Table 7 displays the results from dynamic mixed training. The sample ratio of $0.201 : 0.245 : 0.182 : 0.200 : 0.172$ for $C : N : M : L : S$ was determined based on the inverse of baseline performance values (i.e., poorer baseline performance warrants more samples from that perturbation set).

[2]These sequences represent the most commonly observed orders after conducting experiments as outlined in Table 4, capturing the most frequent positions of perturbations in the constructed sequences.

Analysis. Though the dynamic mixed training surpasses SEQ, it only edges out the mixed training approach when utilizing a total of 1000 and 1500 samples for fine-tuning (Table 6 for K = 200, 300). This shows that dynamically altering challenge set size improves unimodel multi-set inoculation.

	Original Sets			Challenge Sets					
K	α_1	α_2	α_3	char	neg	num	loc	stan	μ
baseline	**72.72**	**64.83**	**62.33**	**57.30**	46.90	67.20	70.20	**67.10**	-
500	71.28	64.42	60.39	56.26	59.22	77.84	76.24	65.38	5.25
1000	71.07	64.72	59.60	57.04	63.24	79.94	79.06	65.50	7.22
1500	72.07	64.81	59.73	56.50	**65.42**	**80.84**	**79.54**	65.64	**7.85**

Table 7: **DYNMIX Results:** ROBERTA$_{\text{INTA}}$ fine-tuned on K samples taken from all perturbation sets in ratios mentioned above.

In conclusion, multi-set inoculation produces robust models than single-set. Further, the MIX and DYNMIX strategies for fine-tuning stand out as more resilient compared to SEQ.

4.2 Results: LLMs

Original Prompt. Table 8 and 9 shows the results for OP$_{\text{ZS}}$ and OP$_{\text{CoT}}$, respectively.[3]

Set	Model	char	neg	num	loc	stan	avg.
R'	Flan-t5-XXL	70.60	77.30	69.00	74.00	79.00	73.98
	LLaMA-2-13b	51.33	54.00	49.67	62.33	53.00	54.07
	LLaMA-2-70b	59.00	63.60	64.60	67.00	60.00	62.84
	GPT-3.5	68.00	69.00	68.66	71.60	70.00	69.45
R	Flan-t5-XXL	63.00	70.00	63.00	65.00	69.30	66.06
	LLaMA-2-13b	39.67	39.33	45.67	56.67	44.67	45.20
	LLaMA-2-70b	54.00	51.60	49.60	57.00	54.30	53.30
	GPT-3.5	51.00	53.00	62.66	61.00	60.30	57.59

Table 8: **Zero Shot Results** (OP$_{\text{ZS}}$)**:** Baseline accuracy for LLMs for Original prompts in zero-shot setting.

Type	Model	char	neg	num	loc	stan	R'
BASE	LLaMA-2-70b	54.00	51.60	49.60	57.00	54.30	62.84
	LLaMA-2-13b	39.67	39.33	45.67	56.67	44.67	45.20
	GPT-3.5	51.00	53.00	62.66	61.00	60.30	69.45
OP$_{\text{CoT}}$	LLaMA-2-70b	63.00	60.00	63.00	61.30	66.00	70.82
	LLaMA-2-13b	61.33	57.00	57.67	59.33	60.00	64.27
	GPT-3.5	63.00	69.60	59.30	61.00	68.00	72.18

Table 9: **Few-shot with CoT** (OP$_{\text{CoT}}$)**.** Results using CoT prompting with exemplars sampled from O.

Analysis. On the Original Zero-Shot Prompts we observe that, (a.) Comparing the results of challenge datasets R$_j$ and their unperturbed version sets R'$_j$ reveals that LLMs similar to PLMs are also sensitive to input data perturbations. (b.) However, the Flan-T5 series, specifically XL and XXL, performs significantly better than other LLMs as it's fine-tuned specifically for the NLI task (Chung et al., 2022). Even the drop in performance due to data perturbation is relatively less. (c.) The poor performance of relatively smaller LLMs, such as LLaMA-2-13b, demonstrates the ineffectiveness of such models in responding to an instruction prompt.(d.) One reason for performance on original numerical set (R'$_M$), is due to model inability to handle mathematical reasoning (2019; 2021; 2021; 2023). Additionally, we find that all models enhanced with CoT (Table 9) outperform those using Zero Shot original prompts. This suggests that simply adding exemplars can enhance a model's resilience to perturbations.

Single Exemplars Multiple Prompts (SEMP): Table 10 presents results for GPT-3.5, with diagonal elements as an analogue to single set inoculation. LLaMA-2 results can be found in Table 11.

Prompt/ Test	char	neg	num	loc	stan	R'
baseline	51.00	53.00	62.66	61.00	60.30	69.05
char	**67.60**	65.30	**66.00**	**69.00**	**67.60**	68.05
neg	60.30	64.60	58.00	59.60	63.30	71.62
num	62.30	66.30	61.00	60.60	64.30	70.24
loc	62.60	63.60	61.00	59.30	64.00	71.30
stan	59.00	**67.60**	61.30	61.00	67.30	**73.76**

Table 10: **SEMP Results on GPT-3.5:** The last column is the average performance on all sets of R′.

Type	set/ π_j	char	neg	num	loc	stan
BASE	R'$_j$	59.00	63.60	64.60	67.00	60.00
	R$_j$	54.00	51.60	49.60	57.00	54.30
SEMP	R'$_j$	69.00	71.00	72.00	72.30	68.60
	R$_j$	53.00	58.00	62.00	62.00	68.30

Table 11: **SEMP LLaMA-2-70b:** Self-testing on perturbation π_j with prompt for π_j and test on R$_j$ and R$'_j$.

Analysis. From Tables 10 and 11, it's evident that incorporating an input perturbation explanation within the prompt enhances the model's accuracy. The results in Table 10 suggest that even a singular perturbation explanation prompts the model to anticipate other perturbations, essentially priming it for a noisy environment. This adaptability is especially pronounced for character perturbations, where improvements span across all challenge sets. Comparisons with instructional prompts and few-shot results show that demonstrations with perturbation explanations improve performance.

Multiple Exemplars Single Prompts (MESP): The results for **MPI** and **MPE** are in Table 12.

Model	Type	char	neg	num	loc	stan	R'
LLaMA	**Base**	39.67	39.33	45.67	56.67	44.67	45.20
	MESP$_{\text{MPI}}$	58.33	51.33	**59.67**	55.00	**62.33**	62.53
	MESP$_{\text{MPE}}$	**60.97**	**56.67**	59.33	**60.67**	62.00	**65.67**
GPT-3.5	**Base**	51.00	53.00	**62.66**	**61.00**	60.30	69.45
	MESP$_{\text{MPI}}$	59.60	**65.30**	62.00	59.00	59.60	68.76
	MESP$_{\text{MPE}}$	**70.00**	63.60	58.30	60.00	**66.80**	**72.52**

Table 12: **MESP** Results on LLaMA-2-13b and GPT.

Analysis. Both models show marked improvement with mixed prompting, indicating that LLMs, when guided with perturbation descriptions and examples, yield more stable outputs. The superior

[3]Results on other open source models in Appendix A.4.

performance of MPE over MPI suggests that including more examples in prompts is more beneficial than detailed perturbation descriptions.

In conclusion, LLMs too face challenges with input perturbations. Simply explaining one perturbation primes the LLM to consider others. Our findings show that a mixed prompting approach with several perturbation instances and brief explanations improves prompt resilience.

Fine-tuning GPT-3.5: We report fine-tuning of GPT-3.5 results in Table 13. This included taking 50 examples from each of the perturbed sets P_j and 50 from the unperturbed set $\bigcup P'_j$ and fine-tuning the model. Mixed fine-tuning outperforms prompts with perturbed exemplars by approximately 7 points. Thus, fine-tuning on perturbed cases improves LLM resilience to input perturbation too. We exclusively compare our results with **MESP$_{\text{MPE}}$**, which is the preferred approach, as elaborated in §4.1.

Model	char	neg	num	loc	stan	R'
Base	51.00	53.00	62.66	61.00	60.30	69.45
MESP$_{\text{MPE}}$	70.00	63.60	58.30	60.00	66.80	72.52
Fine-Tune	**71.67**	**78.33**	**66.67**	**67.33**	**71.00**	**76.40**

Table 13: **GPT-3.5** fine-tuning using MIX.

5 Related Works

Model Robustness Issues. Deep learning models in vision and language domains have exhibited sensitivity to adversarial examples and input distribution shifts, as highlighted in prior studies (Mahmood et al., 2021; Elsayed et al., 2018; Chang et al., 2021; Ren et al., 2019; McCoy et al., 2019; Wang et al., 2021a; Gupta et al., 2023; Zheng and Saparov, 2023). The quest for model robustness in the language domain has led to investigations involving contrast sets (Li et al., 2020a), Checklist (Ribeiro et al., 2020), and attack algorithms (Li et al., 2020b, 2018). The importance of ensuring model robustness (Wang et al., 2022, 2020) cannot be overstated, as even minor input perturbations can significantly impact performance due to model complexity and susceptibility to distribution overfitting (Glockner et al., 2018; Rice et al., 2020; Zhu and Rao, 2023; Moradi and Samwald, 2021).

Improving Model Robustness. Utilizing adversarial examples during training provides a degree of mitigation (Tong et al., 2022; Liu et al., 2019a; Yuan et al., 2023; Kotha et al., 2023; Liu et al., 2023), it falls short of a comprehensive solution for achieving widespread robustness, as it deals only with one facet, i.e., single-set inoculation. Our proposed framework is adept at evaluating model robustness across multiple challenge sets.

Our research complements and extends the work on robustness explored in (Liu et al., 2023; Lu, 2022; Zheng and Saparov, 2023). While (Liu et al., 2023) focuses on integrating consistency loss and data augmentation during model training, our framework uniquely applies to models that are already in use or deployed for specific tasks. Similarly, Lu (2022) addresses the issue of dataset artifacts in natural language inference (NLI) and introduces a multi-scale data augmentation method for mitigation. In contrast, our work concentrates on limited fine-tuning of already trained models and expands the scope to encompass additional dimensions of robustness. On the other hand, (Zheng and Saparov, 2023) examines the robustness of LLMs to perturbed inputs by enhancing noisy exemplars quantity, whereas our study offers a broader, more versatile framework for assessing the robustness of both PLMs and LLMs (using fine-tuning, enhancing instruction quality, and enhancing exemplars both in diversity and quantity).

6 Conclusion and Future Works

We demonstrate that input perturbation poses difficulties for LMs at all scales. While fine-tuned models on a single challenge set can produce robust models, their generalizability to unfamiliar perturbations remains questionable. This motivates the problem of multi-set inoculation, aiming to train a singular model resilient to a myriad of distinct perturbations. We introduce a comprehensive framework to systematically evaluate LM robustness against multiple input perturbations. In addition, we propose three strategies to fine-tune the model on multiple challenge sets. Our results underscore the superiority of mixed fine-tuning in training robust models. Furthermore, we expend our framework to LLMs, leveraging a *COT* prompting enriched with exemplar demonstrations.

Future Directions: We consider the following future directions: (a.) **Complex Sample Selection:** Future plans include adopting advanced sample selection strategies to boost model robustness during fine-tuning, inspired by (Roh et al., 2021; Swayamdipta et al., 2020). (b.) **Composite Perturbation:** We aim to explore the successive application of multiple perturbations on a single sample, represented as $\pi_i(\pi_j(x))$, to understand their combined impact on model performance.

Limitations

While our framework exhibits promising results for language models at different scales, there are several limitations to consider. We study five different perturbations in our framework. The effectiveness of our method, however, is contingent on the availability of data and definitions of these perturbations, which may not be available for unique unencountered perturbations. In addition, the process of sequential fine-tuning presents a challenge in terms of catastrophic forgetting. This necessitates maintaining a repository of both current and historical data and perturbations, which in turn leads to an increase in computational storage. Although our system performs well for tasks in English, processing and adapting to multilingual input data and accompanying models is an area that has to be researched further. We also recognize the opportunity for investigating parameter-efficient fine-tuning and other domain adaptation strategies to potentially enhance the robustness of the model. Finally, it is pertinent to note that the current evaluation of our framework has been limited to specific natural language processing tasks. Its performance in other tasks, such as question-answering and sentiment classification, has not yet been explored. These limitations underscore the need for further research to address these challenges.

Ethics Statement

We, the authors of this work, affirm that our work complies with the highest ethical standards in research and publication. In conducting this research, we have considered and addressed various ethical considerations to ensure the responsible and fair use of computational linguistics methodologies. We provide detailed information to facilitate the reproducibility of our results. This includes sharing code, datasets (in our case, we deal with publicly available datasets and comply with the ethical standards mentioned by the authors of the respective works.), and other relevant resources to enable the research community to validate and build upon our work. The claims in the paper match the experimentation results. However, a certain degree of stochasticity is expected with *black-box* large language models, which we attempt to minimize by keeping a fixed temperature. We describe in the fullest detail the annotations, dataset splits, models used, and prompting methods tried, ensuring the reproducibility of our work. For grammar correction, we use AI-based writing assistants, and for coding, we utilized Copilot. It's important to note that the genesis of our ideas and the conduct of our research were entirely independent of AI assistance.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Kai-Wei Chang, He He, Robin Jia, and Sameer Singh. 2021. Robustness and adversarial examples in natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 22–26, Punta Cana, Dominican Republic & Online. Association for Computational Linguistics.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020a. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020b. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial examples that fool both computer vision and time-limited humans. *Advances in neural information processing systems*, 31.

Ivan Fursov, Alexey Zaytsev, Pavel Burnyshev, Ekaterina Dmitrieva, Nikita Klyuchnikov, Andrey Kravchenko, Ekaterina Artemova, and Evgeny Burnaev. 2021. A differentiable language model adversarial attack on text classifiers.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Ashim Gupta, Rishanth Rajendhran, Nathan Stringham, Vivek Srikumar, and Ana Marasović. 2023. Whispers of doubt amidst echoes of triumph in nlp robustness. *arXiv preprint arXiv:2311.09694*.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1121–1133, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. MathPrompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, Toronto, Canada. Association for Computational Linguistics.

Nupur Jain, Vivek Gupta, Anshul Rai, and Gaurav Kumar. 2021. TabPert : An effective platform for tabular perturbation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 350–360, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lan Jiang, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, and Rui Jiang. 2022. ROSE: Robust selective fine-tuning for pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2897, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2023. Understanding catastrophic forgetting in language models via implicit inference. *arXiv preprint arXiv:2309.10105*.

Alex Kulesza and Ben Taskar. 2011. k-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1193–1200.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020a. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop*

on Analyzing and Interpreting Neural Networks for NLP, pages 126–135, Online. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Jiachi Liu, Liwen Wang, Guanting Dong, Xiaoshuai Song, Zechen Wang, Zhengyang Wang, Shanglin Lei, Jinzheng Zhao, Keqing He, Bo Xiao, et al. 2023. Towards robust and generalizable training: An empirical study of noisy slot filling for input perturbations. *arXiv preprint arXiv:2310.03518*.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models.

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

Zhenyuan Lu. 2022. Multi-scales data augmentation approach in natural language inference for artifacts mitigation and pre-trained model optimization. *arXiv preprint arXiv:2212.08756*.

Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. 2021. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations.

John 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. 2021. Incorporating external knowledge to enhance tabular reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2799–2809, Online. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Leslie Rice, Eric Wong, and Zico Kolter. 2020. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2021. Sample selection for fair and robust training.

Abhilash Shankarampeta, Vivek Gupta, and Shuo Zhang. 2022. Enhancing tabular reasoning with pattern exploiting training. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 706–726, Online only. Association for Computational Linguistics.

Jiho Shin, Clark Tang, Tahmineh Mohati, Maleknaz Nayebi, Song Wang, and Hadi Hemmati. 2023. Prompt engineering or fine tuning: An empirical assessment of large language models in automated software engineering tasks.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Shoujie Tong, Qingxiu Dong, Damai Dai, Yifan song, Tianyu Liu, Baobao Chang, and Zhifang Sui. 2022. Robust fine-tuning via perturbation and interpolation from in-batch instances.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021a. Infobert: Improving robustness of language models from an information theoretic perspective.

Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. 2023. Adversarial demonstration attacks on large language models.

Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021b. SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online. Association for Computational Linguistics.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CAT-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online. Association for Computational Linguistics.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4514–4525, Online. Association for Computational Linguistics.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. *International Conference of Learning Representation*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2023. HyPe: Better pre-trained language model fine-tuning with hidden representation perturbation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3246–3264, Toronto, Canada. Association for Computational Linguistics.

Shuo Zhang and Krisztian Balog. 2019. Auto-completion for data cells in relational tables. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 761–770, New York, NY, USA. ACM.

Hongyi Zheng and Abulhair Saparov. 2023. Noisy exemplars make large language models more robust: A domain-agnostic behavioral analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4560–4568.

Bin Zhu and Yanghui Rao. 2023. Exploring robust overfitting for pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5506–5522, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Related Works:- Tabular Datasets and Models.

Research on semi-structured tabular data has delved into tasks like tabular natural language inference, fact verification (Chen et al., 2020b; Gupta et al., 2020; Zhang and Balog, 2019), and more. Techniques for improving tabular inference include pre-training methods (Yu et al., 2018, 2021; Eisenschlos et al., 2020; Neeraja et al., 2021). Moreover, recently shared tasks such as SemEval'21 Task 9 (Wang et al., 2021b) and FEVEROUS'21 (Aly et al., 2021) have expanded upon these topics.

A.2 Implementation Details

For RoBERTA-LARGE: For creating a baseline model the RoBERTA-LARGE model is fine-tuned on INFOTABS for 10 epochs with a learning rate of $1e^{-5}$ with batch size of 4 and adagrad optimizer. (Shankarampeta et al., 2022; Jain et al., 2021). For fine-tuning on challenge set P_i, we use a learning rate of $3e^{-5}$. This learning is selected after experimenting with various learning rates(specifically $5e^{-4}$, $1e^{-4}$, $5e^{-5}$, $3e^{-5}$, $1e^{-5}$, $5e^{-6}$, $1e^{-6}$) and observing their performance on single set inoculation for various training dataset sizes(specifically 100, 300 and 500). We have used NVIDIA RTX A5000(24 GB), NVIDIA RTX A6000(48 GB) and Google Colab GPU(A100) for conducting different experiments. For API calls on GPT-3.5, we have used CPU only. The cost for fine-tuning is: $0.008 for training,$0.012 for usage input, $0.016 for usage output for 1k tokens. The cost for prompting is $0.008 for 1k tokens. The number of examples are highlighted in the Section 3 and 4.2.

For the mix fine-tuning we ran the evaluation for 5 different random seeds for each challenge set combination. Average metrics for calculating the final accuracy of mix training to avoid random noise.

For LLMs: We used GPT-3.5 model and LLaMA-2 models for our experiments. GPT-3.5 has been used with a temperature setting of 0.3 (to preserve reproducibility) and 1000 maximum new tokens. LLaMA-2 model has been used after quantization with QLoRA (Dettmers et al., 2023), with *nf4* 4-bit quantization. Double quantization has been employed and *torch.bfloat16* has been used for computations during the quantization.

An interesting observation for LLaMA-2 was made which led to the empirical observation that too many examples within the system prompt may also hurt model performance as evidenced from examples here and here(*anonymized for submission*). This observation influenced our decision to demonstrate the model using its past conversational history and to limit the system prompt to instructions specific to the model.

For SEMP, we utilized three demonstrations from the challenge set and three from the original set. We used six demonstrations for OP_{CoT}. We use ten demonstrations for GPT-3.5 in the $MESP_{MPI}$ setting and fifteen in the $MESP_{MPE}$ setting. For LLaMA-2, we used eight demonstrations in $MESP_{MPI}$ setting and eleven in the $MESP_{MPE}$ setting. There are minor differences in the NLI Task Explanation for prompts chosen for GPT-3.5 and LLaMA-2 models, these can be found in the corresponding data and examples are given below. This was done as LLaMA-2 performs better with labelling neutral examples as "it is not possible to tell" instead of "neutral".

For the Flan-T5 series, the model has been pre-trained on the NLI/RTE task. We used the same format for getting the results for zero shot setting (OP_{ZS}) as used in Huggingface inference API example for premise-hypothesis.

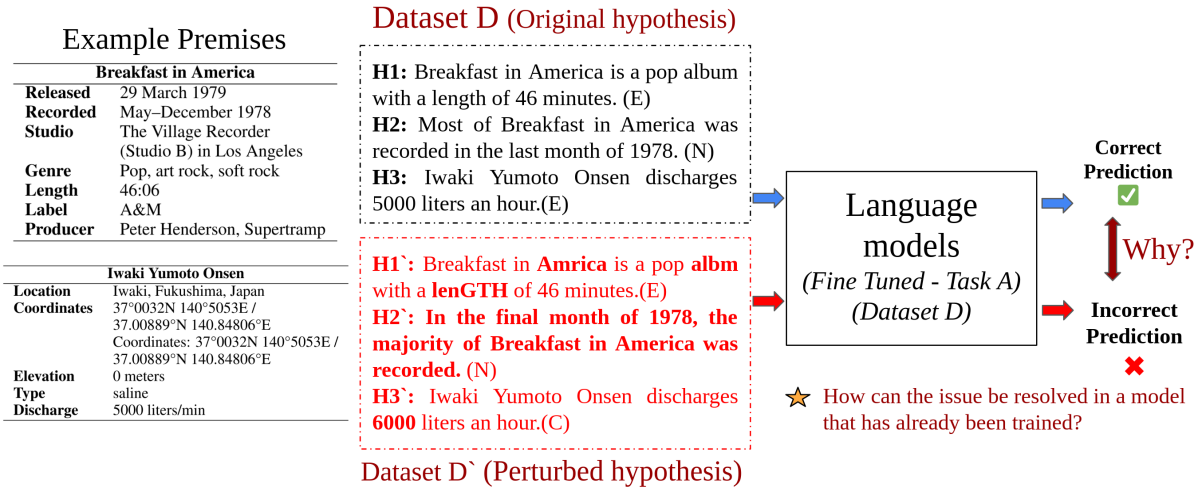

Figure 2: **Language Models Sensitivity to Input Perturbations.** Language models trained on Tabular-NLI (*Task A*) with Original Hypothesis(Dataset D) are not reliable for perturbed hypotheses (Dataset D' for character, paraphrasing, or numeric perturbations examples).

Example for OP_{ZS} on Flan-T5 series
Premise: At my age you will probably have learnt one lesson.
Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

Fine-Tuning on GPT-3.5: The system prompt was provided with the NLI task explaination and mixed perturbation awareness prompt consisting of a brief explanation of all the perturbation types as used in $MESP_{MPI}$ for the model gpt-3.5-turbo-0613. The answering scheme does not require an explaination here. A total of 300 samples are used for fine-tuning. Auto hyper-parameters yielded a batch size of 1, 3 epochs and learning rate multiplier of 2[4]. An example is given below:

Listing 1: Example for fine-tuning GPT-3.5

```
{
  "messages": [
    {
      "role": "system",
      "content": "In this task, we will ask you to make an
           inference about the information presented as the
           premise..."(Prompt containing NLI task
           description, perturbation awareness and
           Description of limitation adepted from
           MESP_MPI as in GPT-3.5).
    },
    {
       "role": "user",
       "content": "Premise:  The region of WIMA is Worldwide
            .  WIMA was founded in 1950.  The location of
           WIMA is the United States.  The website of WIMA
           is www.wimaworld.com. Hypothesis: WIMA is
           located in Gambia."."
    },
    {
      "role": "assistant",
      "content": "Answer: No"
    }
  ]
}
```

[4]More details can be found on the openAI documentation for fine-tuning.

A.2.1 MESP Prompting Example

Below an **example prompt for LLaMA-2 for** $MESP_{MPE}$.

NLI Task Explanation
In this task, we will ask you to make an inference about the information presented as the premise. We will show you a premise and a hypothesis. Using only the premise and what you believe most people know about the world, you should choose one of the following options for the premise-hypothesis pair:
1."yes": Based on the information in the premise and what is commonly known, the hypothesis is definitely true, in such a case respond with "yes".
2."no": Based on the information in the premise and what is commonly known, the hypothesis is definitely false, in such a case respond with "no".
3."it is not possible to tell": Based on the premise, the hypothesis could be true, but could also be false. We need additional information that is neither commonly known, nor explicitly mentioned in the premise which makes us come to a conclusion. We cannot make an inference about the hypothesis in such a case respond with "it is not possible to tell".

The next part, *perturbation awareness* contains the brief explanation of the respective perturbations. Explanation for one of the perturbation is as below. We have mentioned the prompt for other perturbations later in this section.

Perturbation Awareness
About Typos: When labelling sentences based on a premise, it's crucial to recognize and address errors and typos that may occur during hypothesis writing. Typos encompass mistakes like spelling errors and punctuation errors that commonly appear in written content. While numeric typos, involving number replacements, should generally be left uncorrected as they may still make sense in context, character typos, such as misspellings or incorrect word formations, should be corrected to ensure clarity. Maintaining this distinction is essential for preserving hypothesis meaning and readability. It is very important that if you suspect a typo in the hypothesis, attempt correction using premise hints without prompting the user and then attempt to label it yourself.
About Attention to Numbers: ...
About the Concept of Negation: ...
About Attention to Locations: ...
About Paraphrasing: ...

Description of limitation It is critical that you do not use information other than the premise. Take the premise to be ground truth and known to be correct. Use no external knowledge.

Answering
Answer with an explanation in the following format, restricting the answer to only one of the following: "yes" or "no" or "it is not possible to tell"
E: <explanation>
A: <answer>

There are multiple *demonstrations* based on the method. We have specified the number of demonstrations used in the implementation details section. In case of the MESP, the demonstrations contains instance of unperturbed as well as perturbed hypothesis NLI tasks. A single instance of a demonstration is shown below:

Demonstrations
Premise: The official languages of Hong Kong Special Administrative Region of the People's Republic of China are Chinese, English. The regional language of Hong Kong Special Administrative Region of the People's Republic of China is Cantonese. The official scripts of Hong Kong Special Administrative Region of the People's Republic of China are Traditional Chinese, English alphabet. The government of Hong Kong Special Administrative Region of the People's Republic of China is Devolved executive-led system within a socialist republic.
Hypothesis: The Hong Kong Special Administrative Region of the People's Republic of China grants official status to more than one language.
E: To make an inference about the hypothesis, we need to either know directly or deduce how many languages are official in Hong Kong Special Administrative Region of the People's Republic of China. We can see in the premise that There are two official languages: English and Chinese. As the hypothesis says "more than one". As two is more than one, the answer is Yes.
A: yes
Premise: ...
Hypothesis: ...
E: ...
A: ...
.
.
.

We have shown the prompt in the raw text format but depending on the model the prompt may be changed to adapt to the model's specific behaviour. For example in case of LLaMA-2 model, the NLI task explanation, Perturbation awareness and Description of limitation section are provided as the system prompt, which is consistent with the paper Touvron et al. 2023.

The only difference between MESP_{MPE} and MESP_{MPI} is that the former has more number of CoT examples of each perturbation in the demonstration section whereas the later has more detailed description of each perturbation in the perturbation awareness section. The perturbation awareness for each type of perturbation for both of the method is at the end of this section.

A.2.2 SEMP Prompting

For the SEMP method, the perturbation awareness section contains only description of only one kind of perturbation adapted from the *perturbation awareness* section as in MESP_{MPI} and the demonstration section contains demonstrations of only one type of perturbation demonstration and with unperturbed demonstrations.

A.2.3 OP_{ZS} Prompting

In case of zero-shot prompting we only explain the NLI task to the model briefly and provide it with

the answering format. We have provided example of OP_{ZS} below as used in **GPT-3.5**:

NLI Task Explanation for GPT-3.5
In this task, we will ask you to make an inference about the information presented as the premise. We will show you a premise and a hypothesis. Using only the premise and what you believe most people know about the world, you should choose one of the following options for the premise-hypothesis pair:
Based on the information in the premise and what is commonly known, the hypothesis is definitely true, in such a case respond with Yes.
Based on the information in the premise and what is commonly known, the hypothesis is definitely false, in such a case respond with No.
Based on the premise, the hypothesis could be true, but could also be false. We need additional information that is neither commonly known, nor explicitly mentioned in the premise which makes us come to a conclusion, in such a case respond with Neutral.

In the OP_{ZS} the perturbation awareness part is not given. So, model is not made aware of any perturbations explicitly.

Description of limitation
Avoid using information that you may know if you believe that it is not generally known.

Answering
Now classify the following Premise-Hypothesis pair. Answer only with one word: Yes or No or Neutral.

As this is the zero-shot prompting no demonstration is provided.

A.2.4 OP_{CoT} Prompting

In case of the few-shot with CoT prompting(OP_{CoT}), we will also provide examples of the NLI task on unperturbed examples along with its chain of thought explanation as a part of demonstrations. The prompt for OP_{CoT} on **GPT-3.5**.

NLI Task Explanation
Same as in for OP_{ZS}.

Note, that there is no perturbation awareness for CoT prompts.

Description of limitation
It is very important and critical that you do not use information other than the premise that you may know if you believe that it is not generally known. This restriction should not prevent you from exploring the premise repeatedly and making some assumptions and deeper inferences from the information within the premise.

Demonstration
Here are some examples:
Premise: Jerusalem is a city. The jewish of Jerusalem is 64%. The time zone of Jerusalem is UTC+02:00 (IST, PST). The area code of Jerusalem is +972-2.
Hypothesis: Christians comprise a big part of the population of Jerusalem.
To make an inference about the hypothesis, we need to either know directly or deduce the population division in Jerusalem. As stated in the premise, Jewish (religion) constitutes 64 percent of the population in Jerusalem. Hence the hypothesis must be false as the Christians(religion) can't possibly constitute a big part of the population, as the majority is taken up by the Jewish. The answer is No.
Premise: ...
Hypothesis: ...
CoT with answer: ...
.
.
.

Note that in all of the methods the premise-hypothesis pair for NLI task will be at the end of the prompt which will be appended with the shown prompt of each method.

A.2.5 Detailed perturbation awareness prompts

All prompts for perturbation awareness for $MESP_{MPE}$:
Find below the prompt for *perturbation awareness* description for different perturbations.

Perturbation Awareness
About Typos: already shown in the MESP prompt.

About Attention to Numbers: Precise handling of numerical information is paramount in labelling tasks. Be diligent in ensuring numerical data accuracy, considering context, surrounding words, and arithmetic operators. Labels should reflect nuanced inferences drawn from numerical values and word usage. It is very important to recheck numeric calculations and arithmetic and mathematical operations.

About the Concept of Negation: Understanding negation is crucial as it can significantly alter sentence meaning. Single negation involves using negative words like "not" to express negativity, while double negation can turn a negative statement into a positive one. Triple negation is rare but also conveys a positive meaning. Close attention to context is essential to avoid misinterpretation.

About Attention to Locations: Location accuracy is a top priority in labelling tasks. Use abbreviations and basic location knowledge, but avoid historical facts. Verify location data with external resources when critical. Meticulously review and edit location-related responses for precision.

About Paraphrasing: In labelling tasks, hypotheses may not mirror the premise's wording exactly. Paraphrasing, or rephrasing with the same meaning, is common. Carefully analyze premise for similar meanings and core concepts, even if phrasing varies. Paraphrasing skills help identify and label relevant information accurately.

Prompts for perturbation awareness MESP_{MPI}:

Perturbation Awareness *About typos:* When performing a labelling task on sentences based on a premise, it's important to understand that errors and typos can occur during the writing of questions. Typos are mistakes made when typing or printing, which can include spelling errors and punctuation errors. These errors can commonly appear in written content and can sometimes affect the clarity and accuracy of a question.
The concept of numeric and character typos in questions is important for maintaining the integrity and meaning of a sentence or premise:
Numeric typos, where a number is accidentally replaced by another number, should generally not be corrected. This is because the new number may still make sense in the context and altering it could change the question's meaning significantly. It's crucial to recognize that the typo might convey a different question altogether.
On the other hand, character typos, such as misspellings or incorrect word formations, should be corrected. These typos often result in words that have no meaning or make the question unclear. Correcting character-based typos is essential to ensure the question remains coherent and can be understood by the reader.
Maintaining this distinction is vital for ensuring that the question retains its intended meaning and readability. Numeric typos, although errors, can sometimes add unique value to a question, whereas character typos usually hinder comprehension and should be rectified whenever possible.
While numeric typos (errors in numbers) may not always need correction, character-based typos (errors in letters or characters) should be corrected. Numeric typos when a number is replaced by another number, shouldn't be corrected as this can mean a different question altogether where the new number still makes sense. Character typos where the newly formed word (after a typo) has no meaning, should be corrected and attempted to be reformed to the original word hints of the original word may also be made from the premise.
The reason typos happen during typing is because our brains focus on conveying meaning rather than the fine details of individual characters. This phenomenon can lead to errors slipping through.
In a labelling task, it's crucial to be vigilant about character-based typos as they can affect the interpretation of the premise and the accuracy of labelling.

About attention to locations: Here is some additional information which may help. Prioritize Location Accuracy: In this labelling task, it is of utmost importance to ensure the precise handling of location-related information. Pay close attention to locations and prioritize accuracy over other details. Use Abbreviations and Basic General Knowledge: Allow for the use of abbreviations like "NY" (New York) or "IND" (Indianapolis or India either may work depending on context). Basic general knowledge about locations, such as their geographical features and neighboring regions, is acceptable. However, do not include historical facts or general events about the place. Verify with External Resources: Encourage the utilization of external resources for verification when dealing with critical location data. Whenever possible, cross-reference the provided information with reliable sources such as maps, atlases, or official websites to ensure correctness. Review and Edit Meticulously: Emphasize the importance of reviewing and editing location-related responses meticulously before finalizing the answer. Double-check the spelling, coordinates, and other location-specific details to guarantee precision.

About attention to numbers: Please pay meticulous attention to numerical information. When performing labelling tasks, it is crucial to handle numerical data with precision. Ensure that the responses contain specific numerical values and context. Emphasize the importance of self-rechecking critical numerical information, and remind yourself to thoroughly review and edit numerical responses for accuracy before finalizing the answer.
In labelling tasks, the hypotheses may contain numerical values. When encountering such cases, carefully identify the numerical data and ensure that it is accurately labelled. Pay close attention to the context and surrounding words as well as arithmetic operators (e.g., +, -, *, /) that may influence the meaning of the numerical value.
Your goal is to provide labels that infer the answer from correct numerical values and comparisons and also reflect the nuanced inferences made from the presence of more or less types of words and arithmetic operators. This entails understanding the role of numerical data in the context of the hypothesis and accurately capturing its significance in the labels.
Remember that precision and accuracy in handling numerical information are paramount in labelling tasks. Take your time to review and edit your numerical responses, double-checking for any potential errors or omissions to ensure the highest quality labelling results.

About the concept of negation: It may also be necessary to understand the concept of negation to make correct inferences. Negation in sentences is the process of expressing the opposite or denial of something. When someone has to pay close attention to statements, understanding negation is crucial because it can change the meaning of a sentence significantly.
Single Negation: In a sentence with a single negation, a negative word like "not" or "no" is used to express a negative statement. For example, "I do not like ice cream" means the person dislikes ice cream.
Double Negation: While less commonly used than single negation, this occurs when two negative words are used in a sentence, such as "I don't want no ice cream." In this case, the double negative creates an affirmative or positive meaning, so the sentence means "I want ice cream.
Triple Negation: While used very rarely, triple negation involves the use of three negative words in a sentence, like "I don't need no help." In this case, it also conveys a positive meaning, indicating that the person doesn't require any assistance.
For someone paying close attention to statements, it's essential to recognize double or triple negations to accurately understand the speaker's intended meaning. These constructions often appear in colloquial speech, so close attention to context and word usage is necessary to avoid misinterpretation.

About paraphrasing: When performing a labelling task where you need to analyze a sentence or a piece of text, it's crucial to understand that the question posed may not always be presented in the exact same words as the information you are reading. This is where the concept of paraphrasing comes into play.
Paraphrasing involves rephrasing a sentence or passage while retaining its original meaning. It's a common practice in various contexts, including academic writing, as it allows for the expression of the same idea in different words. Paraphrasing can help you better understand and articulate information, and it's especially important when dealing with labelling tasks where the wording might not match exactly.
In the context of a labelling task, you should be aware that the question you're trying to answer might be a paraphrased version of the information presented in the text or a sentence in the premise. This paraphrasing may not be perfect, and there could be slight variations or synonyms used. Therefore, it's essential to carefully read and analyze the text, looking for similarities in meaning rather than relying solely on identical phrasing. By doing so, you can effectively identify and label the relevant information, even if it's not presented verbatim. Paraphrasing skills are valuable in such tasks as they allow you to recognize the core concepts and convey them accurately, regardless of the wording used in the question.
If you feel like the hypothesis may have a typo, you should attempt to correct it yourself by taking hints from the premise to guess the actual hypothesis and then attempt to label it. Do not prompt the user to correct the hypothesis, attempt it yourself.

A.3 LLM Answer Extraction Module

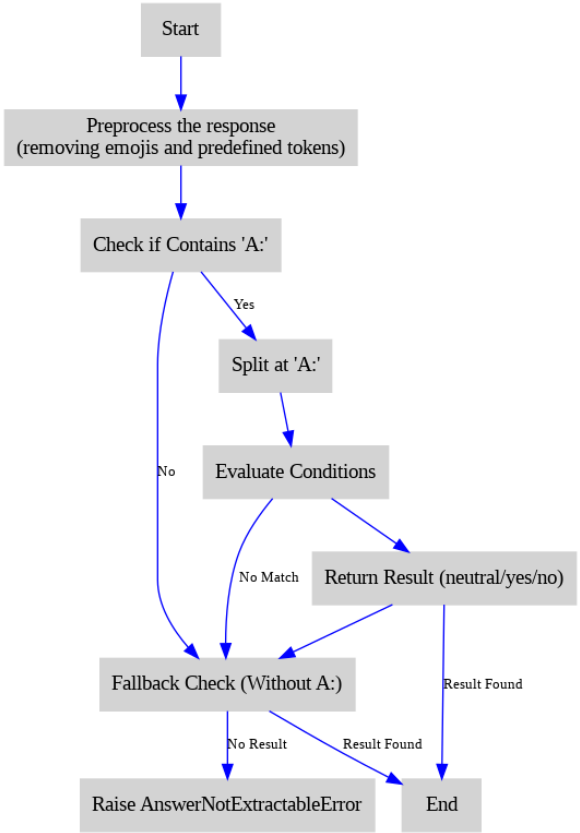

Figure 3: **Flowchart for answer extraction**

The outputs of the large language models are not necessarily in the required format even after explicitly specifying the format. Thus, we needed to design a method to extract out the answer from the very verbose outputs of the model. So, we have shown the flow of the answer extraction module in the Fig 3. The module begins by removing non-essential elements such as emojis from the text, enhancing text clarity for analysis. It then searches for a key marker ('A:'), indicating the start of a relevant response. Upon identification, this section is isolated for evaluation.

The module's functionality is centered on categorizing responses into affirmative, negative, or neutral based on specific phrases. In cases where the marker is missing, it reassesses the entire text, ensuring comprehensive analysis. If the response remains ambiguous, the module raises an error.

A.4 Additional Results on Zero-shot

The Table 14 shows zero shot (OP_{ZS}) accuracy for different language models.

Set	Model	char	neg	num	loc	stan	avg.
UNPERTURBED R'	Flan-T5-small	39.30	48.60	39.30	59.60	47.00	46.76
	Flan-T5-base	55.60	63.60	55.60	68.00	58.60	60.28
	Flan-T5-large	70.60	75.00	64.60	77.00	71.60	71.76
	Flan-T5-XL	72.30	76.30	66.70	78.60	75.30	73.84
	Flan-T5-XXL	70.60	77.30	69.00	74.00	79.00	73.98
	LLaMA-2-13b	51.33	54.00	49.67	62.33	53.00	54.07
	LLaMA-2-70b	59.00	63.60	64.60	67.00	60.00	62.84
	GPT-3.5	68.00	69.00	68.66	71.60	70.00	69.45
PERTURBED R	Flan-T5-small	33.00	40.00	49.30	71.00	47.00	48.06
	Flan-T5-base	44.00	54.00	55.60	68.60	58.00	56.04
	Flan-T5-large	54.00	66.00	62.30	65.00	67.60	62.98
	Flan-T5-XL	63.00	68.00	64.00	66.00	71.30	66.46
	Flan-T5-XXL	63.00	70.00	63.00	65.00	69.30	66.06
	LLaMA-2-13b	39.67	39.33	45.67	56.67	44.67	45.20
	LLaMA-2-70b	54.00	51.60	49.60	57.00	54.30	53.30
	GPT-3.5	51.00	53.00	62.66	61.00	60.30	57.59

Table 14: **Zero Shot Results** (OP_{ZS})**:** Baseline accuracy for LLMs for Original prompts in zero-shot setting.

A.5 Confusion Graphs

The confusion graph below represents the confusion matrix values for char, neg, num, loc, stan perturbation for a particular method in the results section. This results provide the insights on which type of hypothesis out of entailment, contradiction and neutral are more difficult for the model with given method. The arrow from A to B represents the percentage of examples which has true label A and has been predicted as B. All the graphs are on perturbed sets.

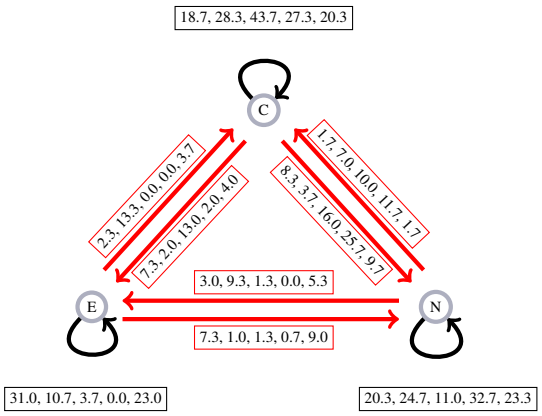

Figure 4: Confusion graph MESP_{MPE} for GPT-3.5 on char, neg, num, loc and stan respectively.

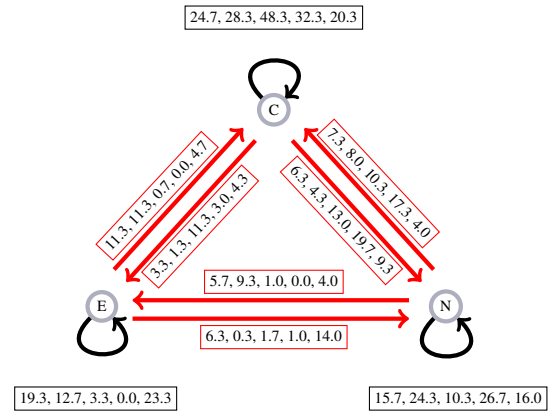

Figure 5: Confusion graph MESP_{MPI} for GPT-3.5 on char, neg, num, loc and stan respectively.

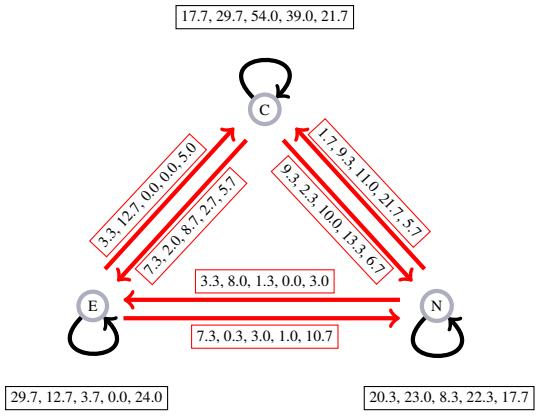

Figure 6: Confusion graph $\text{SEMP}_{\text{CHAR}}$ for GPT-3.5 on char, neg, num, loc and stan respectively.

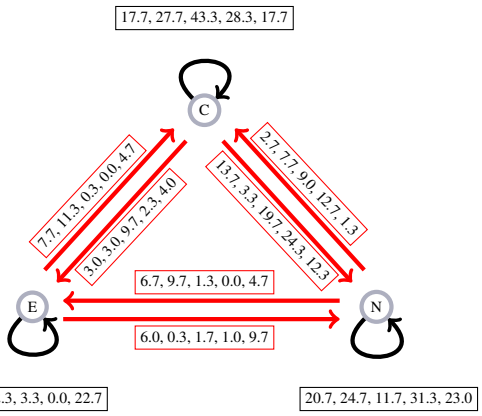

Figure 7: Confusion graph SEMP_{NEG} for GPT-3.5 on char, neg, num, loc and stan respectively.

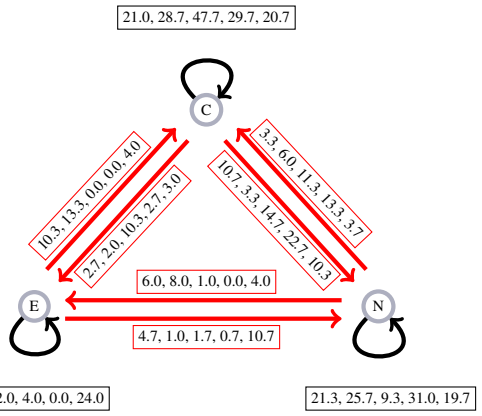

Figure 8: Confusion graph SEMP_{NUM} for GPT-3.5 on char, neg, num, loc and stan respectively.

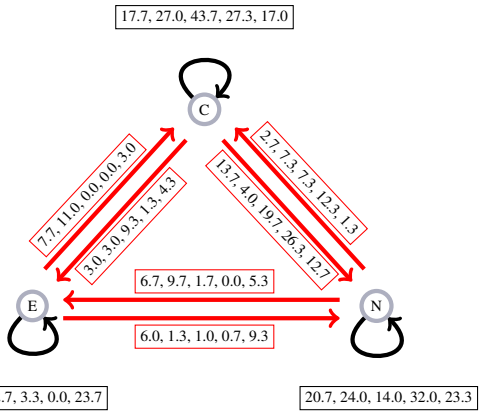

Figure 9: Confusion graph SEMP_{LOC} for GPT-3.5 on char, neg, num, loc and stan respectively.

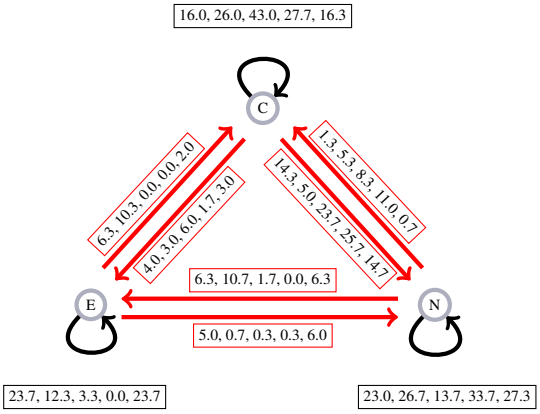

Figure 10: Confusion graph $\text{SEMP}_{\text{STAN}}$ for GPT-3.5 on char, neg, num, loc and stan respectively.

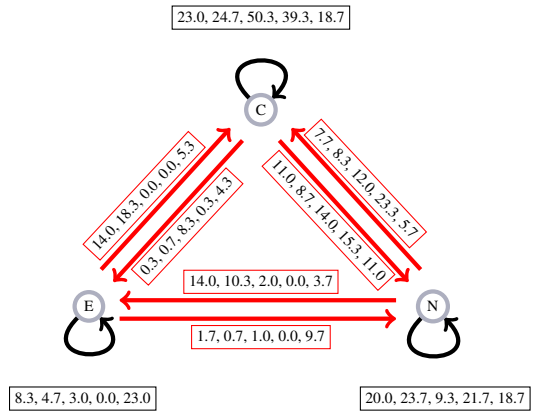

Figure 11: Confusion graph OP_{ZS} for GPT-3.5 on char, neg, num, loc and stan respectively.

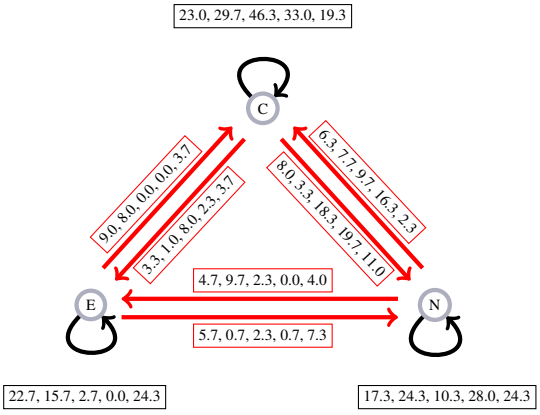

Figure 12: Confusion graph OP_{CoT} for GPT-3.5 on char, neg, num, loc and stan respectively.

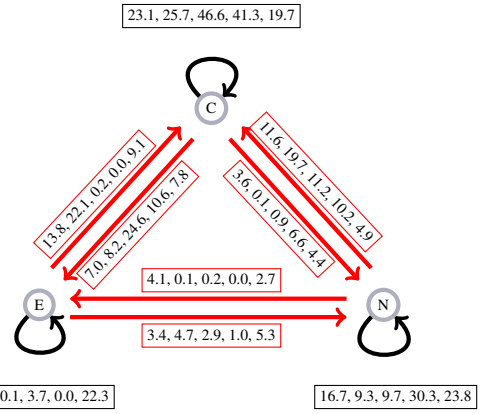

Figure 13: Confusion graph $\text{SEQ}_{\text{COL-ASC}}$ for $\text{RoBERTA}_{\text{INTA}}$ on char, neg, num, loc and stan respectively.

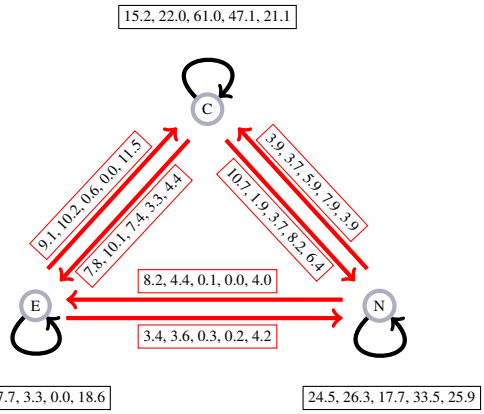

Figure 14: Confusion graph MIX with 500 examples each for $\text{RoBERTA}_{\text{INTA}}$ on char, neg, num, loc and stan respectively.

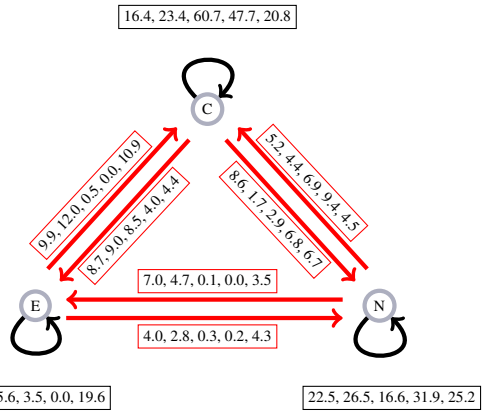

Figure 15: Confusion graph DYNMIX with total 1500 examples for $\text{RoBERTA}_{\text{INTA}}$ on char, neg, num, loc and stan respectively.