Illuminating Blind Spots of Language Models with Targeted Agent-in-the-Loop Synthetic Data

Philip Lippmann, Matthijs T.J. Spaan, and Jie Yang
Delft University of Technology

Abstract

Language models (LMs) have achieved impressive accuracy across a variety of tasks but remain vulnerable to high-confidence misclassifications, also referred to as unknown unknowns (UUs). These UUs cluster into blind spots in the feature space, leading to significant risks in high-stakes applications. This is particularly relevant for smaller, lightweight LMs that are more susceptible to such errors. While the identification of UUs has been extensively studied, their mitigation remains an open challenge, including how to use identified UUs to eliminate unseen blind spots. In this work, we propose a novel approach to address blind spot mitigation through the use of intelligent agents – either humans or large LMs – as teachers to characterize UU-type errors. By leveraging the generalization capabilities of intelligent agents, we identify patterns in high-confidence misclassifications and use them to generate targeted synthetic samples to improve model robustness and reduce blind spots. We conduct an extensive evaluation of our method on three classification tasks and demonstrate its effectiveness in reducing the number of UUs, all while maintaining a similar level of accuracy. We find that the effectiveness of human computation has a high ceiling but is highly dependent on familiarity with the underlying task. Moreover, the cost gap between humans and LMs surpasses an order of magnitude, as LMs attain human-like generalization and generation performance while being more scalable.

1 Introduction

Language models (LMs) have achieved remarkable accuracy across a wide range of predictive tasks, but remain vulnerable to out-of-distribution data (Papernot et al., 2016; Wang et al., 2019; Brown et al., 2020). Small, lightweight LMs – while easier to train and run on limited hardware, and therefore favored in domain-specific applications – are especially prone to UUs due to their reduced robustness (Wang et al., 2022; Du et al., 2023). Larger LMs, although generally more robust, require significant computational resources for both training and inference, limiting their usability (Touvron et al., 2023). This vulnerability often leads to prediction errors, including in high-stakes applications such as suicide prevention (Large et al., 2017) and criminal justice sentencing (Crawford, 2016), where reliable and unbiased predictions are critical. A particularly challenging class of errors, referred to as *unknown unknowns* (UUs), occurs when the model confidently misclassifies an input as the incorrect label (Attenberg et al., 2015). These UUs tend to *cluster* into *blind spots* in the feature space, areas where the model consistently produces high-confidence misclassifications due to biases in the training data (Lakkaraju et al., 2017; Liu et al., 2020). On the left side of figure 1 we show an example of a mispredicted label at a high confidence, resulting in a UU, that forms part of a blind spot.

The identification of UUs and blind spots has been extensively studied (Attenberg et al., 2015; Bansal & Weld, 2018; Vandenhof, 2019; Liu et al., 2020), including approaches involving human oversight to aid in detection (Cabrera et al., 2021; Han et al., 2021). *Mitigating* blind spots – especially how to move from identified blind spots to unseen ones – remains an unresolved challenge. Simple approaches to tackling only *already discovered* blind spots, such as relabeling previously identified UUs and using them for additional training (Han

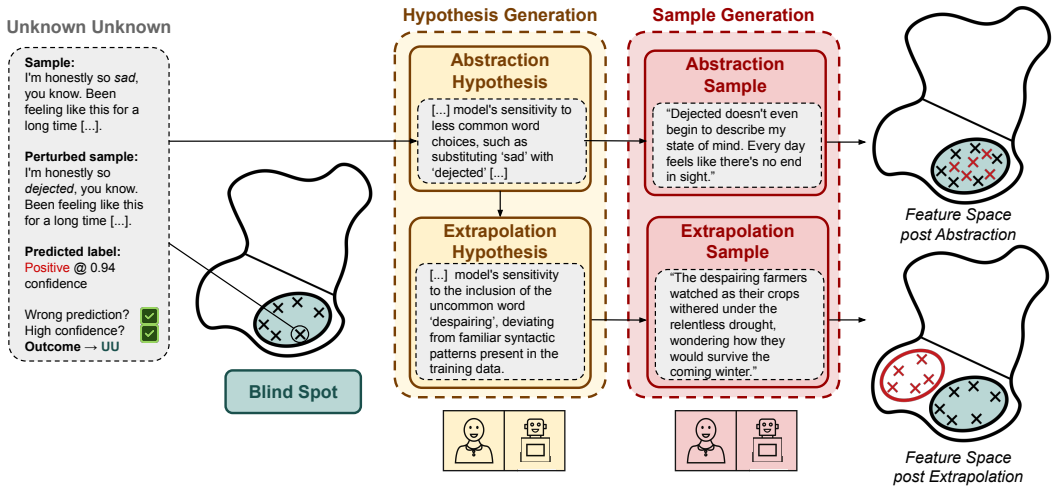


Figure 1: In a sentiment classification task, we begin with a UU resulting from a perturbation – denoted by a cross in the feature space. This UU is then used to generate an initial hypothesis via abstraction through human computation or an LM. This abstraction hypothesis can then either be used to generate a synthetic samples that target the existing blind spot or to generate a new hypothesis via extrapolation, which in turn is then used to generate synthetic samples targeting an unseen blind spot.

et al., 2021), do not scale and fall short of ensuring a holistic reduction in blind spots. Thus the only blind spots of the model that can be illuminated using such reactive approaches are those that correspond to seen data, with those that correspond to unseen data remaining out of reach.

In this paper, we introduce an agent-in-the-loop workflow that proactively mitigates blind spots of LMs by employing intelligent agents – either humans or large LMs – to *characterize* blind spots and subsequently generate targeted synthetic data. We pose that the key to mitigating these blind spots lies in the generalization abilities of the agent, allowing them to hypothesize patterns of discovered UUs and similarities between seen and unseen UUs using prior knowledge (Gluck et al., 2011; Banich & Caccamise, 2010; Allaway & McKeown, 2020). To this end, we guide agents to formulate these hypotheses in natural language, either describing the found blind spot consisting of discovered UUs (abstraction) or reasoning about undiscovered blind spots (extrapolation), as is shown in figure 1. Using these hypotheses, we guide agents toward the generation of synthetic samples targeted at blind spots, improving the robustness of LMs through subsequent retraining by reducing the number of high-confidence misclassifications without sacrificing overall predictive accuracy. Our workflow is designed to flexibly integrate intelligence from both humans and LMs, with specific mechanisms to incorporate human computation or LMs. Additionally, the workflow can incorporate existing adversarial attack methods to proactively illuminate blind spots, further enhancing its adaptability and effectiveness.

Our workflow proves to be a viable means of distilling knowledge from intelligent agents to small LMs, making them more robust while maintaining their lightweight advantages. Through our comprehensive experiments, we find that our method is capable of substantially reducing the number of high-confidence misclassifications without decreasing accuracy. On average, we are able to reduce the number of UUs by 19.08%. Further, we show that for our method LMs are more effective overall than human agents, achieving a 22.37% reduction in UUs compared to a 15.78% reduction when using human-generated data. Additionally, LM-generated data are far more economical, making them a more scalable solution for improving the robustness of small models. Finally, we observe that humans surpass LMs in certain tasks, particularly those that align more closely with human intuition due to participants' greater familiarity with them.

2 Agent-in-the-loop targeted data generation

Our proposed approach to blind spot mitigation involves engaging a human or LM in three tasks: *hypothesis generation via abstraction*, *hypothesis generation via extrapolation*, and *synthetic sample generation*. These tasks are designed to characterize and mitigate blind spots, ultimately reducing high-confidence misclassification. The workflow is schematically illustrated in figure 1. The human computation component of our study is implemented through a survey study, the details of which are provided in appendix A, while the equivalent LM prompts are given in appendix B.

2.1 Problem formulation

For UU discovery, let the dataset be $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x is the original text sample and y the original ground truth label. Without having access to y, a predictive model θ is tasked with generating a label prediction $y_p = \theta(x)$ at a confidence $c \in [0, 1]$. Formally, a UU occurs when (1) θ predicts the wrong label $y_p \neq y$ and (2) the prediction is made with high confidence $c \geq \tau$.

In this work, in addition to dealing with the blind spots that naturally occur in models as a result of training, we make use of adversarial UU discovery, where we increase the number of misclassifications by introducing perturbations. For this, a black-box adversarial perturbation model G generates perturbed samples $\bar{x} = G(x)$, where $\bar{x} \neq x$. The model θ is then used to predict new labels $y'_i = \theta(\bar{x}_i)$ at a confidence c. The resulting perturbed dataset, denoted $\mathcal{P}$, consists of the new samples and predicted labels $(\bar{x}, y')$. If a perturbation occurs, there is an additional requirement for a misclassification to be considered a UU: (3) $\bar{x}$, regardless of its label indicated by θ, maintains the same underlying true label y as x post perturbation.

Given a predictive model θ trained on a dataset $\mathcal{D}$, our objective is to mitigate UUs produced by θ. To systematically reduce high-confidence misclassification, we seek to identify patterns in discovered UUs and generate targeted synthetic data $\{x^s, y^s\}$ for a set of UUs, where x^s is the synthetic sample and y^s represents the corresponding ground truth label for the synthetic sample. This data is then used to further train θ and thus reduce the blind spots present.

2.2 Generalization via hypothesis creation

For UU mitigation, we employ intelligent agents (humans or large LMs) to generalize from identified UUs to create hypotheses in natural language regarding the underlying causes of these UUs. As we use perturbations, such hypotheses are based on pairs of original and perturbed samples, $(x_i, y_i) \sim \mathcal{D}$ and $(\bar{x}_i, y'_i) \sim \mathcal{P}$. Humans are adept at using sparse data to generalize (Lake et al., 2015), and this task exploits that capability by focusing on subsets of UUs. Each hypothesis describes the shared characteristics that explain why certain UUs occur and how these characteristics might generalize to other, unseen UUs. The goal is not merely to explain individual failure cases but to construct hypotheses that address multiple UUs clustering together into a blind spot. In doing so, we can illuminate patterns within the feature space that the model is consistently misclassifying. To this end, we pursue two distinct but complementary strategies: abstraction and extrapolation.

Abstraction Abstraction involves generating a hypothesis on why a specific UU occurred that generalizes across a set of closely related UUs, revealing underlying patterns within a blind spot. In this step, the intelligent agent is provided with an original sample (x_i, y_i) and, if adversarial perturbations are used, its perturbed counterpart $(\bar{x}_i, y'_i)$. Then the agent is tasked with reasoning abstractly about the factors leading to this UU. Specifically, we instruct them to consider whether these factors involve semantics, syntax, specific words, or something else in the samples that could be the cause of the high-confidence misclassification. This is to guide the agent to identify what most likely contributes to the UU without prescribing rigid criteria, leaving room for creative thinking and allowing the agent to explore unforeseen or nuanced factors. The hypothesis is in natural language and should generalize across other UUs that share these characteristics, expanding our

understanding of the particular blind spot the UU corresponds to. Compared to a mitigation approach that only makes use of a simple reactive relabeling of found UUs, our method comes with the additional advantage that it builds up a corpus of human-interpretable error reports on seen errors of the classification model.

Extrapolation Extrapolation extends the process of hypothesis creation beyond trying to describe discovered blind spots, encouraging the agent to use existing hypotheses and sample pairs (used during abstraction) to uncover new blind spots. This task emphasizes extrapolation, asking the agent to hypothesize new failure modes – also in natural language – that differ from those previously identified. Extrapolative thinking has previously been shown to be a human strong suit (Bartlett, 1958). By ensuring that the new hypotheses are dissimilar from those used for abstraction, we aim to discover new regions in the feature space where the model may be prone to high-confidence misclassification. To avoid the agent overextrapolating, we specifically instruct them to focus on the same topic but to reason about whether a different factor from semantics, syntax, specific words might be responsible that was not mentioned in the abstraction hypothesis. In this step, we present only human-generated hypotheses to human participants and vice versa. An example of hypothesis generation via abstraction and extrapolation is shown in figure 2.

2.3 Synthetic sample generation

Once hypotheses have been generated via abstraction or extrapolation, the agent is tasked with generating synthetic samples. These synthetic samples must align with the structure and context of the original dataset while reflecting the characteristics of the generated hypotheses. For instance, if the dataset consists of movie reviews, the synthetic samples should maintain the form and tone of movie review-related text. The goal of this step is to create new data points that correspond to the blind spots identified during hypothesis generation. These synthetic samples are added to the training dataset, resulting in a dataset that is extended for each synthetic sample and its corresponding label $\mathcal{E} = \mathcal{D} \cup \{x_i^s, y_i^s\}$, where the label is provided by the agent. By incorporating these new samples into training, we aim to enhance the robustness of the predictive model θ by reducing its susceptibility to high-confidence misclassifications. The sample generation process is uniform, regardless of whether the hypothesis was obtained through abstraction or extrapolation. Humans generate samples based on human-created hypotheses, and LMs do the same for LM-generated hypotheses. An example of this type of sample generation from human and LM agents for abstraction and extrapolation is shown in figure 2.

3 Experimental setup

In this section, we present an overview of our experimental design. A schematic illustration of the workflow can be found in figure 3. First, we obtain our initial set of UUs of the fine-tuned classification model from the validation set. Following this, we characterize the blind spots corresponding to these UUs by making the intelligent agent perform generalization as described in section 2, culminating in new synthetic data that we use to retrain the model. Finally, we evaluate this retrained model with respect to accuracy and UU count. As a preliminary study, to verify that our method does indeed address blind spots, we successfully demonstrate that it is possible to artificially create blind spots by hand (*i.e.*, ground truth blind spots) in a model and then illuminate these using our approach in appendix C. In our main study, our experiments instead address mitigating both natural blind spots that occur during normal model training and those created by adversarial attacks. For this, we do not have access to the ground truth blind spots and as such just have indirect evidence that some blind spots are illuminated as the number of occurring UUs is decreased.

3.1 Datasets, models, and perturbations

To evaluate the generality and effectiveness of our approach, we select a diverse set of classification tasks, each representing varying levels of task complexity.

Specifically, we focus on sentiment analysis (SA) using the IMDB dataset (Maas et al., 2011), semantic equivalence (SE) using the MRPC dataset (Dolan & Brockett, 2005), and natural language inference (NLI) using the QNLI dataset (Rajpurkar et al., 2016). The statistics of the dataset for each task are shown in table 1. For blind spot mitigation, we use the validation set to obtain our UUs that are then used to perform the hypotheses generalization. These hypotheses are then used in turn to generate synthetic samples and extend the training set, as shown in figure 3. We limit the number of hypotheses derived from each of abstraction and extrapolation to 1% of the training set size, leading to an additional 73, 500, and 2095 training samples after applying our method for MRPC, IMDB, and QNLI, respectively. These values are treated as hyperparameters and are chosen to balance computational efficiency and effectiveness. We leave further optimization of this split between abstraction- and extrapolation-derived hypotheses to future work. We employ two classification models in our experiments, finetuned for each classification task: BERT (`bert-base-uncased`) (Devlin et al., 2019) and Llama 2 7B (Touvron et al., 2023), selected for their contrasting architecture and size. We choose BERT for its known performance on sentence-level classification tasks and its low number of parameters, while Llama 7B was chosen for its larger (but still manageable) scale and capability in handling more complex language understanding tasks. GPT-3.5 Turbo is incorporated as the teacher model to perform hypothesis and sample generation, as it is superior to both classification models that we use.

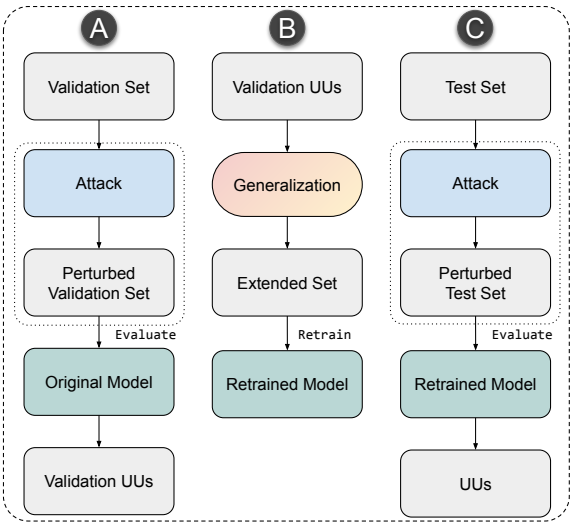


Figure 3: Workflow: (A) Obtain UUs from the validation set on the original finetuned model; (B) use UUs to extend the training data via generalization (figure 1) and thus obtain a more robust model; (C) evaluate this retrained model. Adversarial perturbations in dotted box are optional.

In a black-box setting, where we assume no access to the model's internal parameters, we employ adversarial perturbation techniques to yield more UUs for our method to use. Note that while perturbations aid proactive discovery of blind spots, they are not strictly necessary to our overall approach. Perturbations are generated using TextAttack (Morris et al., 2020), specifically with TextFooler (TF) (Jin et al., 2020) for word-level perturbations and DeepWordBug (DWB) (Gao et al., 2018) for character-level perturbations. Using these two methods, we cover a wide spectrum of adversarial attack types, revealing additional blind spots. We focus on perturbations that maintain semantic integrity, ensuring that the true underlying label remains consistent after perturbation. Manual inspection of 100 random perturbed samples revealed that none had a different underlying true label, affirming that our perturbations are faithful.

3.2 Baseline

As a baseline, we use a reactive relabeling approach based on the previous work by Han et al. (2021), where identified UUs are given a ground truth label, before being reintroduced to the classification model for additional training. This method directly targets blind spots by adding these correctly labeled samples to the extended set. While Han et al. (2021) perform this reintroduction in smaller, iterative batches to identify more UUs, we pool all relabeled UUs in a single batch, as we only concern ourselves with the mitigation of UUs and assume that we have knowledge of whether a sample is a UU or not post classification. This is similar to how we perform the retraining for our method. For a fair comparison, we apply this baseline approach with the same budgetary constraints as our proposed method,

Dataset	Task	#Classes	#Train	#Validation	#Test
MRPC	SE	2	3,668	408	1,725
IMDB	SA	2	25,000	12,500	12,500
QNLI	NLI	2	104,743	5,463	5,463

Table 1: Datasets used, including the task type, number of classes, and number of samples in each of the test, validation, and training sets. Note the split of the original IMDB test set into new validation and test sets.

with new samples making up 2% of the initial training set size. We pose that our method, which uses hypotheses to synthesize new data, will outperform this method by uncovering additional failure modes not captured by relabeling alone.

3.3 Implementation

Following Lakkaraju et al. (2017), we set the confidence threshold for determining high-confidence misclassifications to $\tau = 0.65$. All BERT models were trained for 10 epochs, using a learning rate of 2×10^{-4}, and a batch size of 64. We fine-tune all Llama 7B models using the Low-Rank Adaptation (LoRA) (Hu et al., 2021) method with the following configuration: a LoRA scaling factor of 16, dropout of 0.1, and rank $r = 64$. The target modules are all linear layers in the model, and no bias adjustment is applied. The training is performed over 3 epochs, with a batch size of 8, and gradient accumulation set to 8 steps. We employ AdamW (Loshchilov & Hutter, 2019) as our optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 2×10^{-4} with a warmup ratio of 0.1, followed by a cosine decay. We apply a maximum gradient norm of 0.3 to ensure stability during training and use a weight decay of 0.001 to prevent overfitting.

The human computation component of our study is implemented through a survey study, the details of which are provided in appendix A. A key procedural difference between human and LM-based experiments is the number of examples provided. The human participants receive two examples, while no examples are given to LMs (*i.e.*, zero-shot). This design choice aims to minimize guidance for the LM since few-shot prompting tends to result in overly homogeneous samples, even when using higher temperature settings. The LM prompts for the teacher model are given in appendix B. When prompting the teacher model, we always ask it to explicitly give its reasoning, which we find not only increases performance but also improves interpretability. To ensure the quality of human-generated hypotheses and synthetic samples, we include attention checks (Oppenheimer et al., 2009) in each survey to eliminate inattentive or low-effort responses. For both human- and LM-generated hypotheses and samples, we implement automated quality checks for this purpose. We do not focus on selecting the high-quality responses, but filter out bad-faith ones such as repeated or nonsensical submissions. To be included, all text entries are required to meet a minimum character threshold ($\text{char}_{\text{min}} = 40$) to ensure sufficient content. Additionally, we employed BERTScore (Zhang et al., 2020) to automatically evaluate the similarity of new samples against a reference set in the form of samples from the training set. If the similarity score falls below a threshold of $S_{min} = 0.5$, the sample is discarded.

3.4 Evaluation metrics

We use two key metrics to assess the effectiveness of our approach and the comparative approach. These include the accuracy of the model on the test set and the number of UUs observed during evaluation. Accuracy provides a basic measure of model performance, while the UU count reflects the model's robustness and allows us to reason about the prevalence of blind spots. Note that the accuracy we report is the accuracy of the model before any perturbations are applied, while the number of UUs is post perturbation. Ideally, our goal is to maximize accuracy while minimizing the number of UUs. Our evaluation compares the performance of the original finetuned model with that of the models retrained

		BERT				Llama 7B			
		TF		DWB		TF		DWB	
		Acc. (%) ↑	UUs (#) ↓	Acc. (%) ↑	UUs (#) ↓	Acc. (%) ↑	UUs (#) ↓	Acc. (%) ↑	UUs (#) ↓
MRPC	Original Model	82.38	952	82.38	936	**90.84**	301	90.66	293
	Relabelling Baseline	**82.49**	911	**82.55**	898	90.61	277	**90.73**	268
	Hypothesis (LM)	81.57	851	82.23	882	89.86	149	89.73	164
	Hypothesis (Human)	81.58	**418**	82.10	**802**	90.20	**144**	89.91	**140**
IMDB	Original Model	94.84	1882	95.40	1682	**95.20**	892	**95.33**	810
	Relabelling Baseline	93.94	1732	94.26	1621	94.86	781	95.10	742
	Hypothesis (LM)	**95.40**	**1241**	94.41	1448	94.96	**604**	95.13	**689**
	Hypothesis (Human)	94.43	1518	**95.74**	**1412**	94.67	658	94.90	702
QNLI	Original Model	**89.88**	1923	**89.88**	2597	**90.08**	879	**90.72**	952
	Relabelling Baseline	88.24	1796	88.98	1907	89.90	856	90.60	929
	Hypothesis (LM)	89.31	**1536**	89.21	**1746**	89.58	**741**	90.10	**890**
	Hypothesis (Human)	89.42	2028	89.38	2325	89.16	857	89.73	924

Table 2: Results of the blind spot study across datasets for BERT and Llama 7B as classification models. TF refers to the TextFooler perturbation method and DWB to DeepWordBug. For accuracy (Acc.) a higher percentage is preferable, while for UUs a lower count is better.

on their respective extended dataset $\mathcal{E}$. This allows us to quantify the impact of our approach on mitigating blind spots and improving model robustness.

4 Results

In this section, we report the experimental results on the effectiveness of our proposed method in reducing blind spots across the classification tasks. The results of our methods configured with human- and LM-generated data as well as those of the baselines are shown in table 2. Additionally, we compare human-generated samples to those produced by LMs in terms of effectiveness, scalability, and ease of use.

4.1 Impact of synthetic samples

Observation 1: Our approach leads to a significant and consistent UU reduction across tasks. As part of our evaluation, we find that our method successfully reduces UUs, with a maximum reduction of 56.09% when using human computation on the BERT model with TF for the MRPC task. LMs generally offer more consistent UU reductions, though performance varies by task. On average, across perturbation methods and classification models, our method with LM-based data generation reduced UUs by 23.43%, while human-based data generation led to a reduction of 21.68%. Similarly, regardless of what type of agent generates the data, our method achieves an average reduction in UUs of 35.71%, 21.27%, and 10.70% for MRPC, IMDB, and QNLI, respectively. The only configuration where our method does not reduce UUs is the BERT model on the QNLI dataset, where human-based retraining with TF actually increases UUs by 5.46%. We elaborate on this in observation 3.

Observation 2: Relabeling of UU samples is effective but not as impactful. Simply relabeling UU samples from the validation set and reintroducing them as the extended set leads to a decrease in the number of UUs, albeit a more modest one compared to our method. Relabeling achieves a decrease in UUs of 8.86% and 7.06% on average for BERT and Llama 7B, respectively. This compares to the average decreases achieved by our method: 21.68% when using human-generated data and 23.43% when using LM-generated data This confirms that only reactive illumination of blind spots using seen data is less effective than our method, regardless of agent type.While the average decrease is lower, the relabeling method is very consistent across tasks, as it is not dependent on an agent grasping the task and delivering high quality data. Additionally, it is very cost effective as no human computation or LM querying is necessary. The obvious limitation of this approach is that it only scales to blind spots that have been discovered and therefore has very little transfer learning potential, as it is unlikely that the found UUs will generalize to unseen UUs.

Observation 3: Human performance is very task dependent. We find that human-generated samples may outperform LMs in tasks that align with human intuition. For

tasks such as SE and SA – which are more intuitive to humans compared to NLI, as they more closely resemble everyday tasks – human performance tends to be better, yielding more significant reductions in UUs. In particular, on the MRPC dataset we see a greater reduction in UUs using human-generated data, 35.38% and 52.19% on BERT and Llama 7B, respectively, when compared to when using LM-generated hypotheses and samples 8.21% and 47.31%. In less intuitive tasks such as NLI, humans can generate data of poor quality, leading to a reduction in model robustness, which may even result in an increase in UUs. When analyzing participants' responses for QNLI, we find that several participants did not fully grasp the natural language inference task, which was not the case for SE and SA. Note that these are not purposefully low-effort responses and are therefore not filtered out as described in section 3.3. This shows that irrespective of classification model, there is a task-specific advantage of human computation compared to LM teacher models when there exists a higher degree of familiarity with the task and vice versa. Although LMs provide samples of acceptable quality consistently, rare but high-quality human responses, such as a crowdworker correctly identifying that changing the date "June 15" to "John 15" referenced a Bible verse – an insight that the LM missed – can significantly reduce UUs and thus be more impactful. This suggests that while human-generated responses can have a higher ceiling in certain contexts, LMs deliver more consistent results overall as incorrect responses from just a few human participants can reduce the effectiveness of our method.

Observation 4: Accuracy does not decrease despite improved robustness. In terms of accuracy, extending the training set with human- or LM-generated data did not have a significant effect. Across tasks, accuracy fluctuations of the models with extended training sets remain within $\pm 1\%$ compared to the original models. This contrasts with previous findings that improvements in robustness often come at the expense of accuracy (Tsipras et al., 2019). Detailed perturbation statistics are made available in appendix D, as well as a visualization of prediction confidences for misclassified samples after perturbation. We observe a reduction in high-confidence misclassifications, particularly at the highest prediction confidences. Additionally, there is a clear reduction across the entire confidence range towards lowering the classifier's confidence in its misclassifications. This, in combination with our overall results, indicates that we improve the calibration of the classification models.

4.2 Scalability and ease of use

Observation 5: Our method scales well per sample and by parameter count. Despite only adding a small amount (2% for each task) of synthetic data relative to the total training set size, we achieve significant results in the reduction of UUs. This indicates that our method can scale to large datasets, as only a small number of synthetic samples relative to the total dataset size are required have a significant impact in terms of improving robustness. We study classification models that use a different architecture and have an order of magnitude difference in size (110M parameters for BERT and 7B for Llama). Here, we find that models with a lower number of parameters achieve a performance similar to that of large generative LMs, with comparable accuracy on the IMDB and QNLI tasks, indicating that smaller models may be more suitable for text classification tasks when considering their other advantages, which corroborates previous findings (Yu et al., 2023). This is especially encouraging for use cases where computational resources are limited or speed and transparency are critical.

Observation 6: Obtaining samples via LM is easier and more cost effective. When considering the practical aspects of our study, significant insights emerge regarding the costs and time involved in conducting human- and LM-based generalization experiments. The human study, which included 168 participants, resulted in a total cost of $1072, with an hourly compensation rate of $12 per participant. In contrast, the LM experiment incurred a much lower cost of $46 for generating an equivalent number of generalizations and samples. Although it is challenging to provide precise estimates, the data collection process via human surveys also took substantially longer than the LM-based approach. This highlights the fact that when using LMs, our method is far more cost-effective and generates data almost instantaneously, in stark contrast to the considerable delays associated with human-based study design and data collection. Thus, from a scalability perspective, the LM-based procedure offers clear advantages, being both faster and less expensive. However, in certain

high-stakes or specialized applications such as suicide prevention and criminal justice sentencing, human involvement, including via a hybrid approach where human intuition supplements the efficiency of LM-generated data, may be more advantageous. This is especially true when considering that LM outputs come with no guarantees and may be biased.

5 Related work

Unknown unknowns Attenberg et al. (2015) introduce the concept of querying humans to find UUs in a game-like setting and show that there were patterns to the found UUs. Vandenhof (2019) proposes an approach to identify UUs where human-interpretable decision rules are learned to approximate how a model makes high-confidence predictions. Crowdworkers then contradict these rules by finding an instance that would classify as a UU. Cabrera et al. (2021) explore the use of crowdworkers to generate failure reports for computer vision models to describe how or why the model failed. Han et al. (2021) propose an approach where crowdworkers continuously extend a dataset with relabeled UUs, on which the chosen model is iteratively trained. Instead, we go beyond simple relabeling and characterize found blind spots and explore new, previously unseen blind spots. There are also algorithmic approaches to finding UUs, such as Lakkaraju et al. (2017), who propose utilizing an explore-exploit approach to find groups of UUs. Bansal & Weld (2018) extend this by proposing a utility model that rewards the degree to which the found UUs cover a sample distribution, thus encouraging the discovery of new blind spots. Instead, we do not find the UUs algorithmically, but instead use an LM or crowdworkers to find existing UUs, extrapolate from these to unseen UUs, and generate synthetic data targeting both of these.

Model calibration and robust training The concept of UUs and blind spots is connected to model calibration (Guo et al., 2017; Minderer et al., 2021; Tian et al., 2023). A model that is well-calibrated will have its prediction confidence aligned with the likelihood of the correctness of the prediction and, as such, a model with blind spots is a poorly calibrated model. In the case where the UUs are specifically generated through adversarial attacks, illumination of model blind spots is also related to robust training. UUs that populate these blind spots, when created by such attacks, may be identified as adversarial examples (Ribeiro et al., 2018; Wallace et al., 2019; Wang et al., 2020). This underscores the relationship between our proposed method and robust training practices with the aim of improving the robustness of the model (Madry et al., 2018; Pang et al., 2021). Our method focuses not on general robustness but rather on high-confidence misclassifications and is not limited to just adversarial samples, as we consider UUs that occur naturally without perturbation as well. Several approaches have been proposed to utilize synthetic data to expand training sets (Puri et al., 2020; Claveau et al., 2021). He et al. (2022) explore few-shot prompting LMs to generate task specific synthetic training data. Unlike prior work, we propose a method to generate targeted synthetic data with the purpose of eliminating blind spots that lead to high confidence misclassifications.

6 Conclusion

We propose a method to identify and mitigate blind spots in classification models by leveraging human- and LM-generated generalizations, followed by synthetic sample generation to target UUs and enhance model robustness. Our evaluation demonstrates that our method is effective at addressing model blind spots and achieves a significant reduction in UUs across datasets, while not altering the general performance of the model and therefore maintaining accuracy. Our study sheds light on the notable task dependency of the human ability to characterize blind spots and generate new data and how this ability compares to that of an LM. Future work will focus on optimizing the balance between accuracy and robustness to further enhance model performance.

References

Emily Allaway and Kathleen McKeown. Zero-Shot Stance Detection: A Dataset and Model using Generalized Topic Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8913–8931, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.717. URL https://aclanthology.org/2020.emnlp-main.717.

Joshua Attenberg, Panos Ipeirotis, and Foster Provost. Beat the machine: Challenging humans to find a predictive model's "unknown unknowns". *J. Data and Information Quality*, 6(1), mar 2015. ISSN 1936-1955. doi: 10.1145/2700832. URL https://doi.org/10.1145/2700832.

Marie T. Banich and Donna Caccamise. Generalization of knowledge: Multidisciplinary perspectives (1st ed.). 2010. doi: https://doi.org/10.4324/9780203848036. URL https://doi.org/10.4324/9780203848036.

Gagan Bansal and Daniel Weld. A coverage-based utility model for identifying unknown unknowns. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11493. URL https://ojs.aaai.org/index.php/AAAI/article/view/11493.

Frederic Bartlett. Thinking: An experimental and social study. 1958.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Ángel Alexander Cabrera, Abraham J. Druck, Jason I. Hong, and Adam Perer. Discovering and validating ai errors with crowdsourced failure reports. *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW2), oct 2021. doi: 10.1145/3479569. URL https://doi.org/10.1145/3479569.

Vincent Claveau, Antoine Chaffin, and Ewa Kijak. Generating artificial texts as substitution or complement of training data, 2021.

Kate Crawford. Can an algorithm be agonistic? ten scenes from life in calculated publics. *Science, Technology, & Human Values*, 41(1):77–92, 2016. doi: 10.1177/0162243915589635. URL https://doi.org/10.1177/0162243915589635.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1758–1770, 2023.

J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56, May 2018. doi: 10.1109/SPW.2018.00016.

Mark A. Gluck, Eduardo Mercado, and Catherine E. Myers. Learning and memory: From brain to behavior (2nd ed.). 2011.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Lei Han, Xiao Dong, and Gianluca Demartini. Iterative human-in-the-loop discovery of unknown unknowns in image datasets. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 9(1):72–83, Oct. 2021. doi: 10.1609/hcomp.v9i1.18941. URL https://ojs.aaai.org/index.php/HCOMP/article/view/18941.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. Generate, annotate, and learn: Nlp with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842, 2022.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025, Apr. 2020. doi: 10.1609/aaai.v34i05.6311. URL https://ojs.aaai.org/index.php/AAAI/article/view/6311.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050. URL https://www.science.org/doi/abs/10.1126/science.aab3050.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2124–2132. AAAI Press, 2017.

Matthew Large, Cherrie Galletly, Nicholas Myles, Christopher James Ryan, and Hannah Myles. Known unknowns and unknown unknowns in suicide risk assessment: Evidence from meta-analyses of aleatory and epistemic uncertainty. *BJPsych Bulletin*, 41(3):160–163, 2017. doi: 10.1192/pb.bp.116.054940.

Anthony Liu, Santiago Guerra, Isaac Fung, Gabriel Matute, Ece Kamar, and Walter Lasecki. Towards hybrid human-ai workflows for unknown unknown detection. In *Proceedings of The Web Conference 2020*, WWW '20, pp. 2432–2442, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380306. URL https://doi.org/10.1145/3366423.3380306.

Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pp. 342–351, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930469. doi: 10.1145/1060745.1060797. URL https://doi.org/10.1145/1060745.1060797.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15682–15694. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/8420d359404024567b5aefda1231af24-Paper.pdf.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 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*, pp. 119–126, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.16. URL https://aclanthology.org/2020.emnlp-demos.16.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Computing Surveys*, feb 2023. doi: 10.1145/3583558. URL https://doi.org/10.1145%2F3583558.

Daniel M. Oppenheimer, Tom Meyvis, and Nicolas Davidenko. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology*, 45(4):867–872, 2009. ISSN 0022-1031. doi: https://doi.org/10.1016/j.jesp.2009.03.009. URL https://www.sciencedirect.com/science/article/pii/S0022103109000766.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training, 2021.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy*, pp. 372–387, 2016. doi: 10.1109/EuroSP.2016.36.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5811–5826, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1079. URL https://aclanthology.org/P18-1079.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330.

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. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Colin Vandenhof. A hybrid approach to identifying unknown unknowns of predictive models. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7(1): 180–187, Oct. 2019. doi: 10.1609/hcomp.v7i1.5274. URL https://ojs.aaai.org/index.php/HCOMP/article/view/5274.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. 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)*, pp. 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://aclanthology.org/D19-1221.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 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)*, pp. 5141–5146, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.417. URL https://aclanthology.org/2020.emnlp-main.417.

Wenqi Wang, Run Wang, Lina Wang, Zhibo Wang, and Aoshuang Ye. Towards a robust deep neural network in texts: A survey, 2019. URL https://arxiv.org/abs/1902.07285.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 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*, pp. 4569–4586, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.339. URL https://aclanthology.org/2022.naacl-main.339.

Hao Yu, Zachary Yang, Kellin Pelrine, Jean Francois Godbout, and Reihaneh Rabbany. Open, closed, or small language models for text classification?, 2023. URL https://arxiv.org/abs/2308.10092.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

A User study for human computation

We use Prolific as a crowdsourcing platform for all our participants. Below, we present the structure followed by all survey participants for the generalization user study, consisting of an initial disclaimer, an instruction set, examples, and finally the questions. Here, we use the abstraction and extrapolation assignments on the IMDB dataset as an example. The workflow is very similar between the different generalization assignments and datasets (MRPC, IMDB, or QNLI), with only slight differences in the wording between the surveys to fit the task and dataset used, as they all present the crowd worker with some input and result in plain text output. For the generation assignment, crowdworkers are asked to perform the same steps, with relevant examples related to the structure of the dataset being shown, before finally contributing usable samples based on shown hypotheses.

A.1 Abstraction on IMDB

Disclaimer Crowdworkers were shown an initial disclaimer to inform them that our governing ethics body sanctions this survey and to remind them not to share personal information:

- "Welcome to the Hypothesis Extrapolation Survey! Please carefully read the following: You are invited to participate in our research study. This study is fully sanctioned by our governing ethics body, as is the handling and storing of the resulting data. This research study aims to use your creativity and generalization ability to come up with new abstractions. It will take you approximately 25 minutes to complete. As with any online activity, the risk of a breach is always possible. To the best of our ability, your answers in this study will remain confidential. We will minimize any risks by making this survey completely anonymous. Therefore, please do not provide any personal information anywhere. The anonymous results might be shared publicly in the future. Participation in this study is entirely voluntary, and you can withdraw anytime. Feel free to contact us with any questions or feedback you might have."

Instructions Crowdworkers were then introduced to the specific task (SE, SA, or NLI) as follows:

- "Please read the following examples carefully. All tasks in this survey are related to a single task, sentiment analysis, which tests the sentiment of a sentence is either positive or negative, applied to movie reviews. The goal here is to use your creativity and ability to generalize to spot patterns and come up with new possible samples. A fully worked-out example can be found below, with user-generated text, similar to what you are expected to write, in *italic* and instructions **bold**. You will receive all relevant instructions again when for each question."

Examples Then, they were presented with two examples that match the dataset used, as well as the task (abstraction, expansion, or generation), before being asked if they understood the examples:

- "**There is a sentence pair below, with one original sample (O) and a perturbed one (P), which is similar but had some things changed (shown in double square brackets). These changes may relate to a pattern, related to semantics, syntax, specific words, or something else in the samples, that leads to the wrong True or False label being predicted for semantic similarity.**
- Example 1 – The two samples are:

  O: There was an overarching [[story]] that was [[refusing]] to reveal itself to me. P: There was an overarching [[narrative]] that was [[unable]] to reveal itself to me.

  Formulate a hypothesis on what this pattern for O and P might be and enter it below. Try to be specific when formulating a hypothesis.

  The pattern that caused the wrong prediction may be related to the substitution of the word ""story"" with its synonym ""narrative"".

- Example 2 – The two samples are:
  O: Overall, I [[loved]] the cinematography of this through and [[through]]. P: Overall, I [[looved]] the cinematography of this through and [[thr0ugh]].
  Formulate a hypothesis on what this pattern for O and P might be and enter it below. Try to be specific when formulating a hypothesis.
  Several words have been misspelled in the samples, all related to the letter ""o"". Either more letters are added ""oo"" or the letter is substituted with a number ""0"" that looks similar, making it easy to misread."

Main Questions Finally, the actual questions preceding the text entry field used for data collection all have the same structure with the unique O and P sentences substituted in for each question:

- "The two samples are:
  O: {original sentence} P: {perturbed sentence}
  Formulate a hypothesis on what this pattern might be and enter it below. Try to be specific when formulating a hypothesis."

B Used LM prompts

We specifically instruct the LM to split its hypothesis from its reasoning because, in our experience, this leads to a clearer and more useful answer for further steps.

B.1 Abstraction prompt

- "There is a sentence pair below, with one original sample (O) and a perturbed one (P), which is similar but had some things changed. These changes may relate to a pattern, related to semantics, syntax, specific words, or something else in the samples, that leads to them being the reason the sample is misclassified by a classification algorithm. This misclassification is made at a high level of confidence.
  The model is not trained on the two samples. The two samples relate to {task} and are:
  O: {sentence[0]}
  P: {sentence[1]}
  Formulate a hypothesis on what this pattern might be. Try to be specific when formulating a hypothesis. Your response should always follow the format:
  Hypothesis: {hypothesis}
  Reasoning: {reasoning}"

B.2 Extrapolation prompt

- "There is a sentence pair, with one original sample (O) and a perturbed one (P), which is similar but had some things changed. These changes may relate to a pattern, related to semantics, syntax, specific words, or something else, that leads to them being the reason the sample is misclassified by a classification algorithm. This misclassification is made at a high level of confidence.
  The model is not trained on the two samples. The two samples relate to {task}
  There is an existing hypothesis regarding the samples, that may capture a pattern related to semantics, syntax, specific words, or something else in the sample pair. This pattern leads to a misclassification of the sample.
  The hypothesis is: {hypothesis}
  Formulate a new hypothesis regarding those sentence samples that is concerned with the same topic but is applied to a different possible pattern that could also lead to a misclassification. Try to be specific when formulating a new hypothesis. Your response should always follow the format:
  Hypothesis: {hypothesis}
  Reasoning: {reasoning}"

B.3 Generation prompt

- "There is a sentence pair, with one original sample (O) and a perturbed one (P), which is similar but had some things changed. These changes may be related to a pattern related to semantics, syntax, specific words, or something else that leads to them being the reason the sample is misclassified by a classification algorithm. This misclassification is made at a high level of confidence.

  The model is not trained on the two samples.

  A hypothesis has been formulated regarding the samples, that may capture a pattern related to semantics, syntax, specific words, or something else in the sample pair. These samples led to a classification algorithm misclassifying them at a high level of confidence.

  Given the samples and a previously generalized hypothesis, generate one new sample made up of one or more sentences that relate to {task} and could have a similar effect on the classification algorithm.

  The new sample should be varied and detailed. Follow the logic laid out in the given hypothesis and follow the format of the sample pair (O and P) exactly. Also include whether the new sample should be given a (positive) or (negative) label for the task: {task}.

  The hypothesis is: {hypothesis}

  Your response should always follow the format:

  Sample: {sample}

  Label: {label}

  Reasoning: {reasoning}"

C Synthetic blind spots

We use the synthetic blind spot study akin to a sanity check for our approach. As such, compared to the full natural blind spot study, we use only a single task, a simpler model architecture, and make other simplifications to our mitigation process. We select an LSTM (Hochreiter & Schmidhuber, 1997) as our model of choice due to the absence of pretraining and apply the TF perturbation method on the SA task. The LSTM used is the standard version of the Bi-LSTM provided by Morris et al. (2020).

	Original			**Retrain**		
	Accuracy (%)	Perturbation (%)	UUs (#)	Accuracy (%)	Perturbation (%)	UUs (#)
Clean	88.03	82.22	1725	88.03	82.21	784
Biased R	78.55	78.56	3785	78.61	78.58	2593
Biased P	75.10	75.02	4607	74.25	73.12	1201
Biased N	76.64	76.64	4394	77.38	77.35	845
Biased PN	74.17	73.94	9231	74.81	74.01	2331

Table 3: Results of synthetic blind spot study for accuracy, perturbation success rate, and number of UUs before and after retraining for all LSTM model variants. The used perturbation method is TF and the dataset is IMDB.

C.1 Blindspot creation and mitigation

To assess whether our method can tackle existing synthetic blind spots we perform a type of Controlled Synthetic Data Check (Nauta et al., 2023). We create synthetic blind spots by systematically excluding some data from training that have commonalities, namely containing a positive or negative term according to lexica by Liu et al. (2005). Here, we randomly subsample 600 of each as our selection of positive and negative terms, due to the extensive nature of the lexica.

We create a false positive blind spot by removing samples from the train set using our selection of negative terms, resulting in a *negatively biased* LSTM (N). Similarly, we create

a false negative blind spot, resulting in a *positively biased* LSTM (P), as well as a blind spot resulting from a selection of 50% randomly chosen terms from each, leading to a *positive/negative biased* LSTM (PN). For comparison, we also include a *randomly biased* LSTM (R), where samples were removed from the train set randomly to obtain a size comparable to the P, N, and PN ones.[1]

After creating the synthetic blind spots through biasing, the authors perform the generalization procedure and provide handcrafted hypotheses that precisely describe these, similar to golden labels. To generate the new samples from our handcrafted hypotheses, we prompt the teacher model to generate movie review-related sentences (to fit the chosen task) that follow a given hypothesis. This was done in an attempt to simplify the procedure by taking advantage of human strengths, generalization and extrapolative thinking, and LM strengths, low-cost text generation, simultaneously.

C.2 Synthetic blind spot study results

The mitigation results of this human-LM approach for our Controlled Synthetic Data Check can be seen in table 3. As can be seen in the first column of table 3, before retraining, the overall test accuracy declines in line with the degree to which the train set is biased. Interestingly, the percentage of successful perturbations by TF, i.e., the percentage of successful label flips, closely follows the overall accuracy. This mirrors the findings of Tsipras et al. (2019), that there is a strong relationship between high accuracy and brittleness – or a lack of robustness. The number of occurring UUs as a result of the perturbation does not follow this trend, instead increasing as the training data becomes more biased, as expected. This poses an interesting optimization problem since the model becomes most robust in general terms, i.e., the successful perturbation percentage falls, but simultaneously there is a significant uptick in blind spots as the training sets become more biased.

The effect of retraining on the overall accuracy and perturbation success rate is minimal, with accuracy changing by no more than $\pm$ 1% and perturbation success rate changing no more than $\pm$ 2%. However, the number of found UUs decreases drastically due to the retraining, with reductions of 73.93%, 80.77%, and 74.75% for the biased P, N, and PN models, respectively. The clean and randomly biased models also show a reduction, though less significant at 54.55% and 31.49%, respectively. These results confirm that our method can be used to target synthetic blind spots found in biased models through the use of hypotheses and generated instances, without significantly affecting the performance or general robustness of the model.

D Perturbation statistics and visualization

	MRPC_O	MRPC_L	MRPC_H	MRPC_R	IMDB_O	IMDB_L	IMDB_H	IMDB_R	QNLI_O	QNLI_L	QNLI_H	QNLI_R
Original Accuracy (%)	82.38	81.57	81.58	82.49	94.84	95.40	94.43	93.94	89.88	89.31	89.42	88.24
Accuracy Under Attack (%)	9.80	17.40	12.99	10.42	10.18	10.44	19.21	10.22	8.91	11.67	14.89	9.97
Attack Success Rate (%)	71.83	64.87	68.29	69.65	88.46	93.18	63.85	85.34	87.35	86.80	78.84	84.92
Perturbed Words (%)	7.70	9.9	8.51	7.98	4.59	7.62	9.02	5.50	6.12	8.80	9.57	7.33
Words per Input	39.3	39.3	39.3	39.3	230.0	230.0	230.0	230.0	37.9	37.9	37.9	37.9
Avg. Number of Queries	51.40	68.62	55.17	57.86	185.24	184.94	198.31	186.37	49.38	51.27	56.11	53.27

Table 4: Perturbation statistics across datasets and models for attacks with TF using BERT. Subscripts O, L, H, R denote the original, LM-retrained, human-retrained, and relabeled models, respectively.

To add additional context to the perturbation performed, we supply the detailed attack statistics across all performed perturbations. Specifically, we report *Original Accuracy* and *Accuracy Under Attack* are reported, which are the classifier accuracy on its own and while under attack. Further, *Attack Success Rate* is shown, which is the percentage of successful perturbation attempts to failed ones. Finally, we report the number of *Perturbed Words*, the percentage of words that are perturbed, the *Words per Input*, the average number of

[1]Size of training sets: $N_{Clean} = 25,000$, $N_R = 2,500$, $N_P = 2,439$, $N_N = 3,138$, and $N_{PN} = 2,438$.

	$MRPC_O$	$MRPC_L$	$MRPC_H$	$MRPC_R$	$IMDB_O$	$IMDB_L$	$IMDB_H$	$IMDB_R$	$QNLI_O$	$QNLI_L$	$QNLI_H$	$QNLI_R$
Original Accuracy (%)	82.38	82.23	82.10	82.55	95.40	95.41	95.74	94.26	89.88	89.38	89.38	88.98
Accuracy Under Attack (%)	7.78	13.73	11.94	10.42	9.54	21.43	15.32	12.51	8.21	9.90	7.30	8.67
Attack Success Rate (%)	72.00	70.38	72.64	72.35	59.41	50.59	79.70	56.87	77.54	79.74	82.08	79.27
Perturbed Words (%)	8.47	9.18	9.03	8.91	6.43	8.11	13.09	9.37	7.99	8.32	11.03	8.31
Words per Input	39.3	39.3	39.3	39.3	230.0	230.0	230.0	230.0	37.9	37.9	37.9	37.9
Avg. Number of Queries	56.92	64.37	58.61	58.23	199.32	211.65	201.44	204.12	34.91	33.53	49.09	35.75

Table 5: Perturbation statistics across datasets and models for attacks with DWB using BERT. Subscripts O, L, H, R denote the original, LM-retrained, human-retrained, and relabeled models, respectively.

	$MRPC_O$	$MRPC_L$	$MRPC_H$	$MRPC_R$	$IMDB_O$	$IMDB_L$	$IMDB_H$	$IMDB_R$	$QNLI_O$	$QNLI_L$	$QNLI_H$	$QNLI_R$
Original Accuracy (%)	90.84	89.86	90.20	90.61	95.20	94.96	94.67	94.86	90.08	89.58	89.16	89.90
Accuracy Under Attack (%)	13.85	18.31	12.43	14.09	20.97	18.22	15.09	17.55	12.64	15.29	14.53	13.67
Attack Success Rate (%)	68.70	65.24	69.54	66.89	71.32	75.64	78.31	70.55	83.42	79.12	75.87	81.34
Perturbed Words (%)	9.23	8.12	9.68	8.97	6.45	7.54	10.88	8.36	7.34	8.69	9.11	7.92
Words per Input	39.3	39.3	39.3	39.3	230.0	230.0	230.0	230.0	37.9	37.9	37.9	37.9
Avg. Number of Queries	53.92	62.34	57.92	55.76	191.34	192.85	198.21	194.43	48.22	49.98	52.89	50.76

Table 6: Perturbation statistics across datasets and models for attacks with TF using Llama 7B. Subscripts O, L, H, R denote the original, LM-retrained, human-retrained, and relabeled models, respectively.

	$MRPC_O$	$MRPC_L$	$MRPC_H$	$MRPC_R$	$IMDB_O$	$IMDB_L$	$IMDB_H$	$IMDB_R$	$QNLI_O$	$QNLI_L$	$QNLI_H$	$QNLI_R$
Original Accuracy (%)	90.66	89.73	89.91	90.73	95.33	95.13	94.90	95.10	90.72	90.10	89.73	90.60
Accuracy Under Attack (%)	16.35	14.79	13.87	15.68	21.78	20.32	19.12	22.19	11.78	10.95	14.28	12.44
Attack Success Rate (%)	66.40	63.89	67.56	65.78	70.42	68.55	71.32	74.65	79.78	77.24	82.43	80.34
Perturbed Words (%)	9.11	8.76	9.02	8.86	7.18	6.92	11.54	9.29	8.06	9.11	10.24	8.76
Words per Input	39.3	39.3	39.3	39.3	230.0	230.0	230.0	230.0	37.9	37.9	37.9	37.9
Avg. Number of Queries	60.22	65.14	62.03	61.76	203.56	199.42	204.29	208.23	45.29	43.87	50.77	47.83

Table 7: Perturbation statistics across datasets and models for attacks with DWB using Llama 7B. Subscripts O, L, H, R denote the original, LM-retrained, human-retrained, and relabeled models, respectively.

words per input, and the *Average Number of Queries*, which is how many tries it took the perturbation method to find the best attack. For BERT, the attack statistics for TF attacks are shown in table 4 while the ones for DWB attacks are shown in table 5. For Llama 7B, the attack statistics for TF attacks are shown in table 6 and for DWB in table 7.

To visualize the effect of the retraining across the perturbed samples, we show the confidence at which predictions are made in figure 4 and figure 5. We observe a reduction in high confidence misclassifications; specifically, predictions with confidence greater than 90% are reduced as a result of our method. Additionally, we observe similar results across perturbation methods, indicating that the performance of our approach is not dependent on the type of perturbation method employed.

Example of full workflow for a single UU sample using human computation and LMs on the IMDB Dataset

Original Sample:
Footage of busy highways, planes flying overhead, poor lightning effects, soldiers doing their thing, they're all used in a [[haphazard]] manner, sometimes repeatedly, and they often only have a loose connection to the story Wood is trying to tell. [...]

Perturbed Sample:
Footage of busy highways, planes flying overhead, poor lightning effects, soldiers doing their thing, they're all used in a [[thoughtless]] manner, sometimes repeatedly, and they often only have a loose connection to the story Wood is trying to tell. [...]

Human Abstraction Hypothesis:
"Haphazard can be intentional, vs thoughtless is unplanned. It feels more of an unwillingness rather than a lack of ability."

LM Abstraction Hypothesis:
"The replacement of specific adjectives in the perturbed sample contributes to misclassification. The adjective "thoughtless" in the perturbed sample, may change the perception of the director's skills and use of stock footage. This alteration in sentiment could lead to a different classification outcome."

Human Extrapolation Hypothesis:
"The hypothesis could be that the words were potentially swapped or replaced, such as "bad" being replaced by "evil" which could lead to an incorrect prediction as it is similar but could have a slightly different meaning in some contexts."

LM Extrapolation Hypothesis:
"The use of the word "hypocrite" may have caused the misclassification. The word "hypocrite" implies that the person is saying one thing but doing another, which may be considered a negative trait. However, some people may not interpret the person in the sample as a hypocrite, leading to a difference in sentiment analysis. This difference in interpretation may have caused the misclassification by the algorithm."

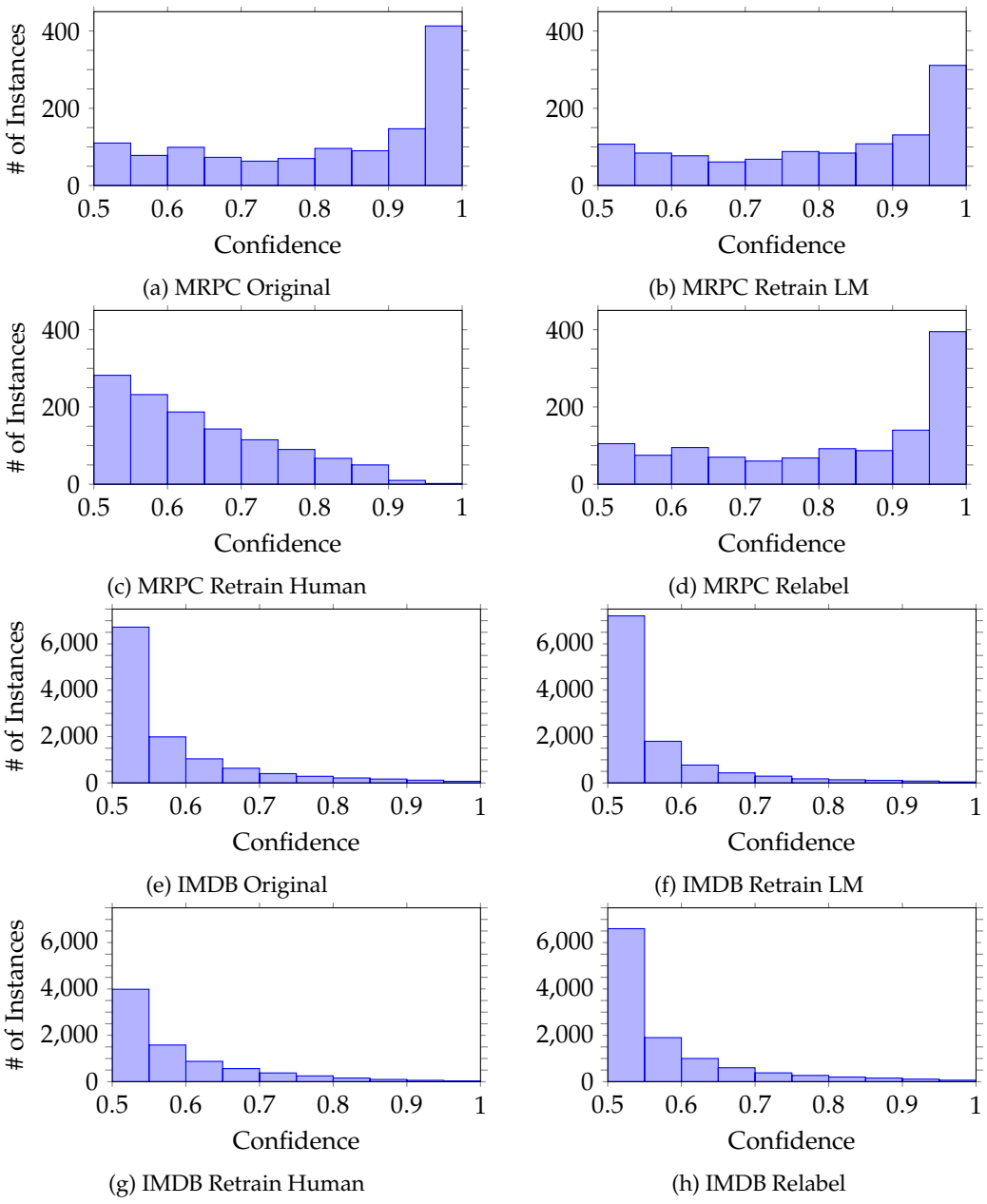
Human Abstraction Sample:
"The scenes felt like they were slapped together with no care at all. It's not that the director didn't have the skill they just didn't bother to try harder. Haphazard is at least creative but thoughtless just shows they didn't wanna put the effort in."

LM Abstraction Sample:
"Skyward Dreams had potential, but the director's use of stock footage feels thoughtless. Scenes of bustling cities and planes flying overhead are inserted without any real purpose, disrupting the flow of the narrative. [...]"

Human Extrapolation Sample:
"The CGI in was straight up evil. The way the effects looked completely ruined the immersion for me, and it felt like the creators didn't even care about quality. I get that sometimes budget is an issue, but this was just on another level. [...]"

LM Extrapolation Sample:
"The protagonist of The Final Betrayal is a true hypocrite. Throughout the film, he preaches loyalty and honesty to his friends, yet secretly manipulates and betrays them behind their backs. This hypocrisy is central to the film's conflict, as the character's outward morality sharply contrasts with his deceitful actions. Despite this glaring flaw, some viewers may interpret his behavior as a survival tactic in a harsh world, rather than outright hypocrisy. [...]"

Figure 2: Example of hypothesis generalization using *abstraction* for the IMDB dataset. The abstraction is performed by a human or LM based on original and perturbed samples.



(a) MRPC Original

(b) MRPC Retrain LM

(c) MRPC Retrain Human

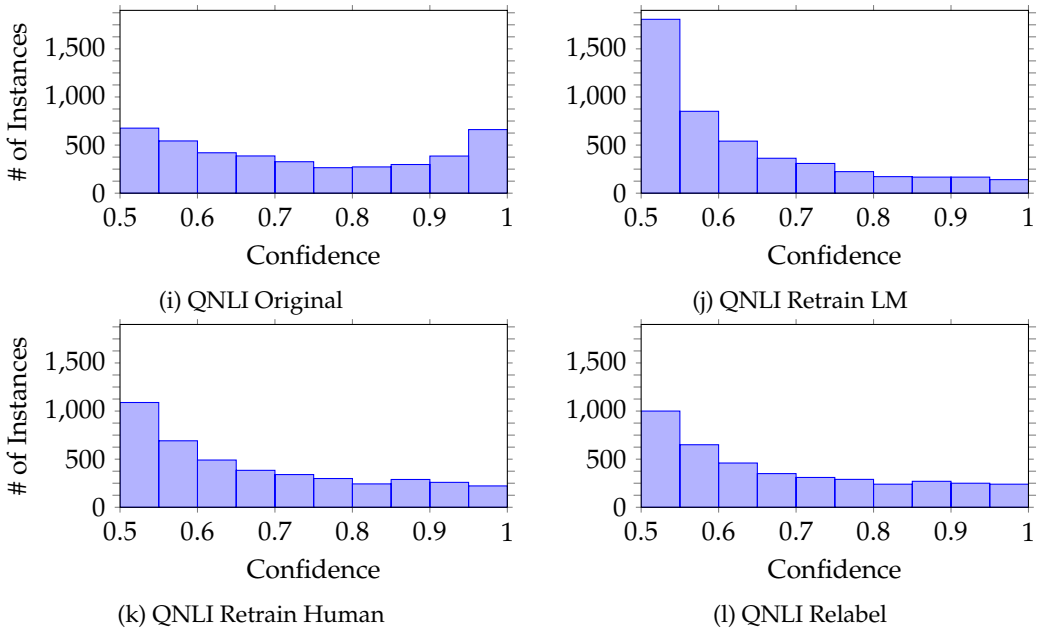
(d) MRPC Relabel

(e) IMDB Original

(f) IMDB Retrain LM

(g) IMDB Retrain Human

(h) IMDB Relabel



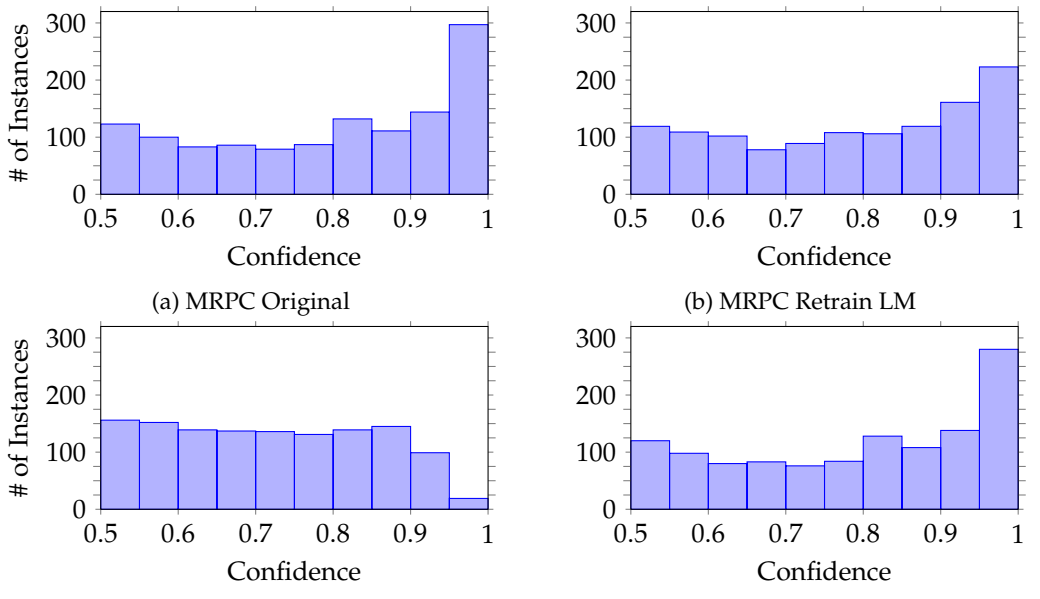
(i) QNLI Original

(j) QNLI Retrain LM

(k) QNLI Retrain Human

(l) QNLI Relabel

Figure 4: Plots of successful perturbations for all datasets when using TF, showing the distribution of the number of instances across confidence bins.

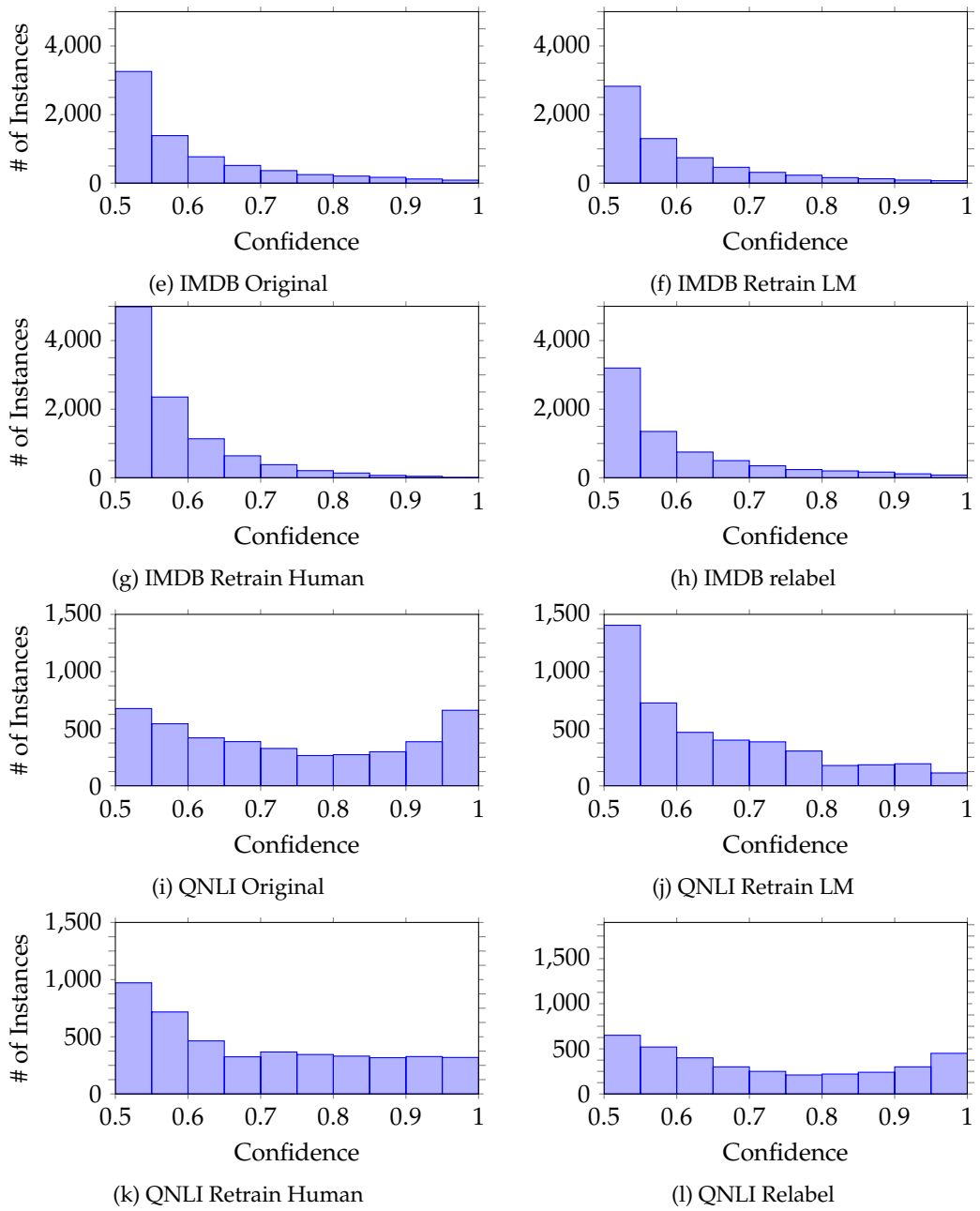


(a) MRPC Original

(b) MRPC Retrain LM

(c) MRPC Retrain Human

(d) MRPC Relabel



(e) IMDB Original (f) IMDB Retrain LM

(g) IMDB Retrain Human (h) IMDB relabel

(i) QNLI Original (j) QNLI Retrain LM

(k) QNLI Retrain Human (l) QNLI Relabel

Figure 5: Plots of successful perturbations for all datasets when using DWB, showing the distribution of the number of instances across confidence bins.