What Learned Representations and Influence Functions Can Tell Us About Adversarial Examples

Shakila Mahjabin Tonni and **Mark Dras**
School of Computing, Macquarie University
shakila.tonni@mq.edu.au, mark.dras@mq.edu.au

Abstract

Adversarial examples, deliberately crafted using small perturbations to fool deep neural networks, were first studied in image processing and more recently in NLP. While approaches to detecting adversarial examples in NLP have largely relied on search over input perturbations, image processing has seen a range of techniques that aim to characterise adversarial subspaces over the learned representations.

In this paper, we adapt two such approaches to NLP, one based on nearest neighbors and influence functions and one on Mahalanobis distances. The former in particular produces a state-of-the-art detector when compared against several strong baselines; moreover, the novel use of influence functions provides insight into how the nature of adversarial example subspaces in NLP relate to those in image processing, and also how they differ depending on the kind of NLP task.

1 Introduction

The high sensitivity of deep neural networks (DNNs) to slight modifications of inputs is widely recognised and makes DNNs a convenient target for adversarial attacks (Szegedy et al., 2014). Creating malicious inputs or adversarial examples by adding small perturbations to the model's inputs can cause the model to misclassify the inputs that would be predicted correctly otherwise. Such adversarial attacks are highly successful in both image and Natural Language Processing (NLP) domains.

In the image domain, due to the straightforwardness of creating adversarial images by calibrating noise to the original records, researchers have explored many high-performing adversarial attacks (Papernot et al., 2016b; Moosavi-Dezfooli et al., 2016; Carlini and Wagner, 2017, for example). The perturbations of the input images degrade the model's performance with a high success rate and are generally imperceptible to a human.

Work in the NLP space has followed that in image processing. Here, in addition to the goal of impacting the model's prediction, adversarial text examples need to be syntactically and semantically sound to the reader. Consequently, adversarial attack techniques on text use semantics-preserving textual changes at the character level, word level and phrase level or sentence level (Pruthi et al., 2019; Alzantot et al., 2018; Li et al., 2020, for example). Table 1 illustrates two examples, showing different types of attack formulation in NLP.

In the image domain, defence against adversarial attack can be 'proactive' or 'reactive' (Cohen et al., 2020), where proactive defence refers to improving the model's robustness (Madry et al., 2018; Gopinath et al., 2018; Cohen et al., 2019) and reactive defence focuses on detecting real adversarial examples before they are passed to neural networks (Feinman et al., 2017; Ma et al., 2018; Lee et al., 2018; Papernot and McDaniel, 2018). Broadly speaking, for reactive methods, the detection of adversarial examples involves taking a conceptualisation of the space of learned representations and the adversarial subspaces within them (Tanay and Griffin, 2016; Tramèr et al., 2017), and then characterising the differences in some function of the learned representations between the actual and the adversarial inputs produced by the DNN; for example, Ma et al. (2018) applied a local intrinsic dimensionality (LID) measure to the learned representations and used that to successfully distinguish normal and adversarial images.

In the NLP space, relatively fewer adversarial defence techniques have been proposed. Among them, many focus on enhancing the models' robustness proactively through adversarial training (Jia et al., 2019; Pruthi et al., 2019; Jin et al., 2020); generating textual samples for proactive adversarial training is computationally expensive because of necessary search and constraints based on sentence encoding (Yoo and Qi, 2021). Reactive adversarial

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 392–411
November 1–4, 2023. ©2023 Asian Federation of Natural Language Processing

text detection techniques have mostly been different from their image counterparts, in that they typically modify the input by e.g. repeatedly checking word substitutions (Mozes et al., 2021; Wang et al., 2022; Zhou et al., 2019) rather than trying to characterise the learned representations; consequently, they focus on detecting synonym-substitution adversarial examples. An exception is the work of Liu et al. (2022), which both adapts LID to the text space and proposes the new MultiDistance Representation Ensemble (MDRE) method; their state-of-the-art results suggest that the detection methods based on learned representations drawn from the image processing domain are a promising source of ideas for NLP.

The particular focus of the present paper is the use of influence functions in adversarial detection methods, proposed for image processing by Cohen et al. (2020). They propose that distances to nearest neighbors (used by previous methods) and influence functions, which measure the impact of every training sample on validation or test set data, can be used complementarily to detect adversarial examples: they argue, with support from the strong results from their method, that adversarial examples locate in different regions of the learned representation space of their neighbors with respect to influence functions, compared to original datapoints (Fig 1). Specifically, in the image space, for original datapoints, nearest neighbors and influence function training points overlap, but for adversarial examples, they do not. Influence functions have only relatively recently begun to be explored in NLP, with Han et al. (2020) finding that, with the variety of classification tasks in NLP, the information provided by influence functions differs from image processing and is task-dependent. In this paper, noting significant differences between inputs in NLP and image processing (continuous versus discrete) and attack types, we explore whether and how they can help in NLP in detecting adversarial examples using learned representations, and what this can tell us about the nature of adversarial subspaces.

We also adapt a second method from the image processing literature, by Lee et al. (2018), which uses a Mahalanobis-based confidence score; this was a strong baseline for Cohen et al. (2020), giving an additional perspective on the nature of adversarial subspaces in NLP.

The contributions of this paper are as follows:

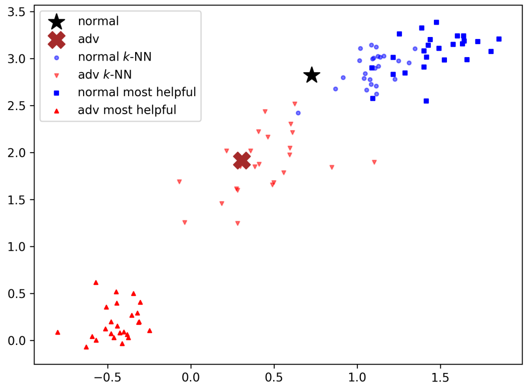


Figure 1: Adversarial examples characterised by divergence in learned representations between nearest neighbors and training points selected by influence functions, unlike original examples (from (Cohen et al., 2020)).

- An adaptation of two adversarial detection techniques from the image processing literature, Mahal confidence (Lee et al., 2018) and Nearest Neighbor Influence Functions (NNIF) (Cohen et al., 2020), into the text domain; we show that we can achieve SOTA results relative to several strong, recent baselines.
- An analysis of how influence functions work in this context, contributes to understanding both the nature of adversarial subspaces in the text space and what information influence functions can provide.

2 Related Work

Adversarial Defences for Image An intuitive adversarial defence is to train a deep neural network to be robust against adversarial input samples by e.g. mixing adversarial samples with the training data (Goodfellow et al., 2015; Madry et al., 2018; Xie et al., 2019); popular platforms like Cleverhans (Papernot et al., 2016a) are available to support robust training. However, such defences, termed as 'proactive', are expensive and vulnerable to optimisation attacks (Cohen et al., 2020).

In contrast, others have proposed 'reactive' defences that identify the variations in the representations learned by the DNN on the original input images to separate the adversarial samples; typically, these posit that adversarial examples can be characterised as belonging to particular subspaces (Tramèr et al., 2017), and the different approaches aim to capture the nature of these subspaces in different ways, with detectors such as logistic regression classifiers built over the learned representations. Feinman et al. (2017) built detectors us-

Original Text	at last, a movie that handles the probability of alien visits with the appropriate depth and loving warmth.	Positive
Char-level (Pruthi et al., 2019)	at last, a movie that handles the probability of alien visits with the appr0priate depth and loving warDmth	Negative
Word-level (Alzantot et al., 2018)	at last, a movie that handles the probability of alien trips with the adequate depth and loving warmth	Negative

Table 1: Examples of textual adversarial instances on IMDB and the prediction of BERT$_{\text{BASE}}$ on them

ing kernel density estimation on the last hidden layer of a DNN. Ma et al. (2018) characterised the dimensional properties of adversarial subspaces using Local Intrinsic Dimensionality (LID), applied to the distribution of distances to neighbors in the region around a sample. Papernot and McDaniel (2018), noting that DNNs are poorly calibrated (Guo et al., 2017), proposed Deep k-Nearest Neighbors (DKNN), a KNN classifier constructed over the hidden layers of a DNN classifier; such a DKNN classifier could match the performance of the DNN while also providing better confidence estimates of prediction, and these confidence estimates are used in identifying adversarial examples. Lee et al. (2018) constructed Mahalanobis distance-based confidence scores from DNNs, using these scores to construct a detection classifier. Cohen et al. (2020) investigated the use of influence functions in adversarial image detection that explain the decisions of a model by identifying influential training examples, and comparing these points to those found in a DKNN approach, using the differences in distributions between real examples and adversarial ones to construct classifiers that outperformed the approaches above. In this paper, we focus on the last two and adapt them to NLP.

Adversarial Defences for Text Improving adversarial robustness remains a widely used mechanism in defending textual adversaries (Li et al., 2016, 2017; Ribeiro et al., 2018; Jones et al., 2020). In NLP, however, there have been fewer reactive methods. To prevent character-level and word-level adversarial perturbations Zhou et al. (2019) proposed the learning to discriminate perturbations (DISP) framework that detects and replaces suspicious words. Mozes et al. (2021) emphasised word frequencies in the texts in determining adversarial perturbations, arguing that adversarially infused words are less likely to occur, and constructed a rule-based, model-agnostic frequency-guided word substitutions (FGWS) algorithm. The approach of Wang et al. (2022) voted the prediction label for a set of samples generated by random word substitutions from a sentence and matched the voted prediction label with the original sentence's prediction label to detect word-level adversaries. Anomaly Detection with Frequency-Aware Randomization (ADFAR) as proposed by Bao et al. (2021) adds anomaly detection as an additional optimization training objective and augments the training set with random rare-frequency word substitutions of the original sentences. Rather than focus on word substitution as the above methods, Mosca et al. (2022) trained an adversarial detector on Shapley additive explanations (Fidel et al., 2020).

In NLP, only Liu et al. (2022) has used the idea of constructing detectors over learned representations as in the image domain, which explored the idea of adapting the LID (Ma et al., 2018) method above. In addition, they proposed the MultiDistance Representation Ensemble Method (MDRE) algorithm that puts together learned representations from multiple DNN models to detect adversarial texts. Unlike other approaches, the same detector could apply to different types of attacks (character-based, word-based, syntax-based) and MDRE in particular improved over baseline methods across the range of attacks. This motivates our adaptation of more recent techniques from the image domain.

Influence Functions The influence function (IF) is a statistical method that captures the dependence of an estimator on any one of the sample (training) points. Koh and Liang (2017) were the first to adapt IFs to image DNNs as a method for interpreting the model's decision: the IF finds the most influential training samples, both helpful and harmful, contributing to each prediction. The essence of the approach is to consider a point z from the training set and compute the change to parameters θ if z were upweighted by a small ϵ; they then defined closed-form expressions $\mathcal{I}(z, z_{\text{test}})$ to identify the most influential points z on a test point z_{test}.

IFs were first applied to NLP deep architectures by Han et al. (2020), and compared with established gradient-based saliency maps as a way of interpreting input feature importance, using sentiment classification and natural language inference (NLI) as testbeds. Their first finding was that IFs are reliable for deep NLP architectures. Their second interesting finding was that while IFs and saliency measures were consistent for sentiment classification, they differed for NLI: they concluded that for more complex understanding tasks like NLI, IFs captured more useful interpretive information. They also found IFs to be useful for identifying and quantifying the effect of data artifacts on model prediction. A few other works have continued investigating the usefulness of IFs in NLP, such as Guo et al. (2021), who proposed a faster method for IF computation by restricting candidates to top-k nearest neighbors.

3 Methods

3.1 NNIF Detector

We follow Cohen et al. (2020)'s Nearest Neighbor Influence Function (NNIF) method and apply it to NLP architectures. The essence of it is, for some point z that may be regular or adversarial, to identify the training points that are most influential and those that are nearest neighbors to z, and to build a classifier based on those that will predict whether z is regular or adversarial based on differences in relative distributions (Fig 1).

We take a DNN classifier and dataset for some particular task (e.g. sentiment classification); we refer to this DNN as the TARGET MODEL. For each test sample z_{test}, we compute the influence scores $\mathcal{I}(z, z_{\text{test}})$ for all training points z, given the target model, and select the top M most helpful and M most harmful (details App B). We then construct a DKNN classifier in the style of Papernot and McDaniel (2018), using the hidden layers of the target model and the training points. For each z_{test} we find the ranks $\mathcal{R}$ and distances $\mathcal{D}$ using this DKNN for the training examples identified by the IFs; we denote by $\mathcal{R}^{M\uparrow}, \mathcal{D}^{M\uparrow}, \mathcal{R}^{M\downarrow}, \mathcal{D}^{M\downarrow}$ the ranks and distances of the $2M$ most helpful and harmful training examples, respectively. We finally construct a logistic regression classifier with features $(\mathcal{R}^{M\uparrow}, \mathcal{D}^{M\uparrow}, \mathcal{R}^{M\downarrow}, \mathcal{D}^{M\downarrow})$ to detect whether an input is adversarial or not.

Where the target model of Cohen et al. (2020) is a ResNet model, ours is a large language model (LLM) base with additional layers that are fine-tuned for the chosen tasks (§4.3). The hidden layers we use for NNIF are then the pre-final additional layers on top of the DNN (§4.5).

3.2 MAHAL Detector

Here we follow Lee et al. (2018), who build a detector that captures the variation in the probability density of the class-conditional Gaussian distribution of the learned representation by the model. Motivated, like Papernot and McDaniel (2018), by the problem that DNNs are poorly calibrated (Guo et al., 2017), they replace the final softmax layer with a Gaussian Discriminant Analysis (GDA) softmax classifier.

For a set of training points $\{(x_1, y_1), ..., (x_n, y_n)\}$ with the label $y \in \{1, 2, \ldots, C\}$, the class mean $\hat{\mu}_c$ and covariance $\hat{\sum}$ are computed for each class c to approximate the generative classifier's parameters from the pre-trained target DNN $f(x)$. Next, from the obtained class-conditional Gaussian distribution, the Mahalanobis distance between a test sample x and its closest distribution is measured to find the confidence score $M(x) = \max_c -(f(x) - \hat{\mu}_c)^T \widehat{\sum}^{-1}(f(x) - \hat{\mu}_c)$. Finally, we label the Mahalanobis scores for the test samples as positive and adversarial samples as negative and input this feature set to an LR detector.

Lee et al. (2018) propose two calibration techniques to improve the detection accuracy and make regular and out-of-distribution samples more separable: (1) *input pre-processing*, where they add a small noise in a controllable manner to the test samples; and (2) *feature ensemble*, which combines the confidence scores from all the hidden layers of the DNN including the final features. Both together substantially improve the performance of the base approach; each individually reaches almost the combination of the two. As for our NNIF detector in §3.1, our target DNN will have several hidden layers, and we explore models both with final layer-only representations and feature ensembles over all hidden layers. The input preprocessing of (1) is appropriate to the continuous space of images, but not in an obvious way to text, so we do not use that.

4 Experimental Setup

We broadly follow the setup of Liu et al. (2022), as the prior NLP work that has used learned representations to detect adversarial examples.

4.1 Tasks and Datasets

We work on the sentiment analysis and the natural language inference tasks, two widely tasks used in the adversarial example generation (Pruthi et al., 2019; Alzantot et al., 2018; Ribeiro et al., 2018; Ren et al., 2019; Iyyer et al., 2018; Yoo and Qi, 2021; Li et al., 2020, 2021; Jin et al., 2020). In addition, these are the two tasks that were used for the investigation of the use of influence functions in NLP (Han et al., 2020).

Sentiment Analysis For the sentiment analysis, we use the IMDB dataset (Maas et al., 2011) that has 50,000 movie reviews, split into 25,000 training and 25,000 test examples with binary labels indicating positive or negative sentiment. IMDB dataset has 262 words per review on average. In all experiments, we use 512 maximum sequence lengths for the language models on IMDB.

Natural Language Inference The Multi-Genre NLI (MULTINLI) dataset (Williams et al., 2018), used for the natural language inference (NLI) task, contains pairs of sentences annotated with textual entailment information. The test examples are mismatched with train examples and are collected from different sources. The dataset has 392,702 training and 9,832 testing examples labelled as three classes: entailment, neutral, and contradiction. Each text of the dataset has 34 words on average. On this dataset, we set the maximum sequence length to 256.

4.2 Attack Methods

We use the implementations from Liu et al. (2022) of two widely used attack methods that apply character-level and word-level perturbations to construct adversarial examples. We take a $\text{BERT}_{\text{BASE}}$ model (§4.3) as the target model. An adversarial attack is successful when the adversaries have different predictions than the target mode's original predictions. Our two methods are (more details in §A.1):

- CHARATT (Pruthi et al., 2019). This is a character-level attack that tweaks the original texts by randomly swapping, dropping and adding characters or adding a keyboard mistake.
- WORDATT (Alzantot et al., 2018). This is a word-level attack that allows the attacker to alter practically every word from the sentence if required with the context-preserving synonymous words. This implementation follows Jia et al. (2019) in speeding up the synonym search.

4.3 Target Model

Following (Liu et al., 2022), we use a pre-trained BERT-base-cased model, adding a fully connected dense layer of 768 nodes, a layer of 50% dropout, and another dense layer of 768 nodes. The dataset split is 80-20 train-test. We train the model for 3 epochs with $5e^{-5}$ learning rate and AdamW optimization without freezing any layer of the backbone model. This $\text{BERT}_{\text{BASE}}$ model achieves 92.90% and 82.01% test accuracies on the IMDB and MULTINLI datasets respectively. The accuracies of the clean model and the model under attack are given in Table 6; we note that in all the cases, CHARATT degrades the classifier's performance comparatively more than WORDATT. Sizes for IMDB and MULTINLI datasets and number of generated adversarial texts from them are in Table 5.

4.4 Detectors

For data to train the adversarial example detectors on, we follow standard practice in image processing (Ma et al., 2018; Cohen et al., 2020) and Liu et al. (2022) and use only those examples that are correctly classified by the target model (§4.3) from the overall test set. Adversarial attacks are then applied to these examples; the originals (labelled positive) and their adversarial alternatives (negative) then form the DETECTION DATASET. Due to the computational intensity of estimating the influential training records for the NNIF method, we limit our detectors to having 10k records (5k tests and 5k adversarial texts) and follow a similar data size for all the other detection methods for comparability. We split the detection dataset 80-20 train-test, and construct and evaluate logistic regression classifiers as detectors over this detection dataset split for our proposed methods (§4.5) and baselines (§4.6).

4.5 NNIF and Mahalanobis Methods

NNIF We adapt the standard NNIF implementation of Cohen et al. (2020). For influence score calculation, Cohen et al. (2020) uses the Darkon module for the image; we instead incorporate the influence function calculation from Han et al. (2020)[1] which uses Linear time Stochastic Second-Order Algorithm (Agarwal et al., 2017) for faster convergence, and makes several adaptations to NLP. We build the DkNN containing one layer with l_2 distance and brute-force search.

Because IF calculations are expensive, like Cohen et al. (2020) and Han et al. (2020) we only

[1] https://github.com/xhan77/influence-function-analysis

sample from among all neighbors: we compute the IF on 6K training datapoints uniformly randomly sampled (Cohen et al. (2020) sample 10K neighbors from 49K training points). We choose $M = 500$ for our main results, which is at the top end of the range of values of M selected by Cohen et al. (2020); we show in §5.2 that, unlike the image processing domain, results in our experiments are broadly monotonically increasing as M increases.

Note that we don't use the faster variant of IF computation of Guo et al. (2021), as NNIF requires *separate* perspectives from IFs and kNNs, and FastIF restricts IF search to subsets of kNNs.

Mahal As per §3.2, we compute the mean and covariance for each class and calculate the Mahalanobis distance score for each normal instance and its adversarial counterpart. Like Ma et al. (2018), we consider both using only the final layer of the model and stacking scores from each layer of the model (feature ensembling). Feature ensembling is always better, so we only include those in the main results, but do separately analyse the contribution of the feature ensembling.

Code For both of these, our code uses the implementation of Cohen et al. (2020) as a starting point and adapts as above.[2]

4.6 Baseline Detection Methods

We evaluate six adversarial text detection methods as our baseline detectors. The first four are from Liu et al. (2022) (we omit the language model, as it operates essentially at the chance), while the other two are also recent high-performing systems.[3] We give more details on the methods in §A.2.

DISP (Zhou et al., 2019). This is a system that aims to correct any adversarial perturbations before an example is passed to a classifier. Liu et al. (2022) adapt this to detecting the adversarial examples.

FGWS (Mozes et al., 2021). This algorithm uses a word frequency threshold and calibrated replacement approach to detect adversarial examples. It is only designed to work against word-level attacks.

LID (Liu et al., 2022). From among image processing detection methods, Liu et al. (2022) adapted the Local Intrinsic Dimensionality (LID) approach of Ma et al. (2018). This technique creates a distribution over local distances for a test record concerning its neighbors from the training set; it then applies these to the outputs of each layer from the target model to create a detection classifier.

Dataset	Detector	Char Attack	Word Attack
IMDB	DISP *	0.8936	0.7714
	FGWS	—	0.7546
	LID	0.814	0.675
	MDRE	0.846	0.7025
	RSV	—	0.8876
	SHAP	0.812	0.764
	NNIF	**1.0**	**0.899**
	Mahal	0.9167	0.8147
MultiNLI	DISP *	**0.7496**	0.6137
	FGWS	—	0.6112
	LID	0.7035	0.5838
	MDRE	0.687	0.6231
	RSV	—	0.6054
	SHAP	0.614	0.697
	NNIF	0.745	**0.7351**
	Mahal	0.6972	0.6211

Table 2: Accuracy of detection classifiers (**best**, second). DISP results reported from Liu et al. (2022).

MDRE (Liu et al., 2022). This has similarities to LID above but uses Euclidean distance rather than the LID measure, and creates an ensemble using different Transformer models (like Liu et al. (2022), we use BERT$_{\text{BASE}}$, RoBERTa$_{\text{BASE}}$, XLNet$_{\text{BASE}}$, BART$_{\text{BASE}}$).

RSV (Wang et al., 2022). In this Randomized Substitution and Vote approach, the assumption is that a word-level attacker aims to find an optimal synonym substitution that mutually influences other words in the sentence. Hence, Wang et al. (2022) randomly replaces words from the text with synonyms in order to destroy the mutual interaction between words and eliminate adversarial perturbation. Like FGWS, this is only designed to work against word-level attacks.

SHAP (Mosca et al., 2022). In this approach, an adversarial detector is trained using the SHapley Additive exPlanations (SHAP) values of the training data for each test data item using the SHAP explainer (Fidel et al., 2020). They experiment on multiple classifiers as the detectors: logistic regression, random forest, support vector and neural network. In our main results, we report the best classifier for each dataset and attack.

5 Evaluation

5.1 Main Results

Results on the detector baselines are in Table 2. (All SHAP detector classifiers in Table 8.) Overall, NNIF is the best, performing with 100% accuracy on CharAtt for sentiment analysis (more than

[2] Code: https://github.com/SJabin/NNIF.

[3] We do not include ADFAR (Bao et al., 2021), as it works and performs similarly to (and was proposed concurrently with) RSV, but has a more complex code implementation.

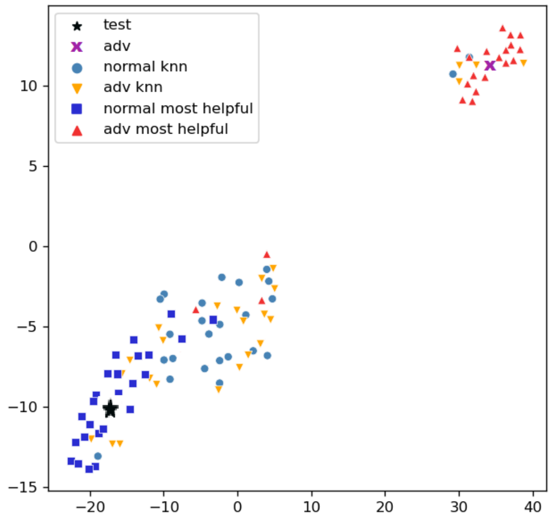


Figure 2: The correspondence between the helpful training records based on IFs in the embedding space of a DNN trained on the IMDB dataset. We present (using t-SNE) the embedding space of a DNN for an actual example (black star) with its adversarial version (purple cross) along with their 25 nearest neighbors (blue) and most helpful samples based on the IF (red).

8% better than the second) and 90% on WORDATT (more than 1% better than the second, RSV, which is tailored to word-level attacks). For MULTINLI WORDATT, it is around 4% better than the second best. The only one where it is not best, CHARATT, is only very slightly below the best performer DISP. (We note that for DISP we report the accuracy values from Liu et al. (2022). This means that the DISP detector used more data in its training set, and so has an advantage in this respect.) MAHAL also performs quite strongly, either better or similar to the baseline detectors, although not as strongly as NNIF; this mirrors the findings in image processing. MDRE results are lower than in Liu et al. (2022) as a consequence of using less data for training all detection classifiers, as discussed in §4.4.

In terms of aggregate task performance, in all our experiments, the detection accuracy on the natural language inference task is lower than the sentiment analysis task in general. As the MULTINLI dataset is a three-class problem and additionally uses mismatched test sentences, the detection is innately harder.

5.2 Analyses

Regions around adversarial examples The assumption underpinning the Cohen et al. (2020) method is that influential training samples and nearest neighbors should overlap for normal examples, but less so for adversarial examples: having two views on 'nearby' points is key, illustrated in Fig 1.

Dataset	Attack	Penultimate layer	Feature Ensemble
IMDB	CHARATT	0.5967	0.9167
	WORDATT	0.536	0.8147
MULTINLI	CHARATT	0.5172	0.6972
	WORDATT	0.4983	0.6212

Table 3: Detection accuracy of Mahalanobis detector in two settings: penultimate layer (no calibration) and feature ensemble.

We produce an analogous figure in Fig 2 for a randomly selected IMDB test point and its adversarial counterpart generated by WORDATT. We plot 25 nearest neighbors and 25 most helpful IF points using t-SNE (van der Maaten and Hinton, 2008). Ideally, normal neighbors and influence points (blue) should be more tightly grouped and closer to the test point (star); Cohen et al. (2020) expect that for the adversarial point (cross), the neighbors (orange down triangle) should often be separated from the influence points (red up triangle). We see this to some extent in Fig 2 with many adversarial neighbors near the normal point but adversarial influence points near the adversarial point.

This is more difficult to see than in the idealised schematic of Fig 1, so for one view of differences in this pair of points we separate IFs and NNs in Fig 3 with recalculated t-SNE for each. It is apparent that the IFs by themselves do a good job of separating normal from adversarial examples here, while the NNs are more mixed. We give representative examples for the other datasets and attacks in App C. The same pattern is true for the IMDB example on WORDATT. For both MULTINLI, however, the IFs are less clearly separating the points, so the NNIF method relies on combining the two (NN, IF) views in the detector.

To verify whether this is more generally true than just visually for Fig 3, we aim to measure how separable the samples of these plots are. As a measure of separability, we train 2000 SVC binary classifiers, one for each of our 1000 sampled test and adversarial point pairs, for both IFs and NNs. Each classifier is trained using GridsearchCV on the top 100 points in t-SNE space (either IFs or NNs), so each classifier corresponds to a plot like those in Fig 3 (App D). Accuracies averaged across the 1000 classifiers are in Table 4, with p-values for a one-tailed test of proportions (positing alternative hypothesis H_1 that the IF classifier is more accurate). Table 4 indicates that the IF points are

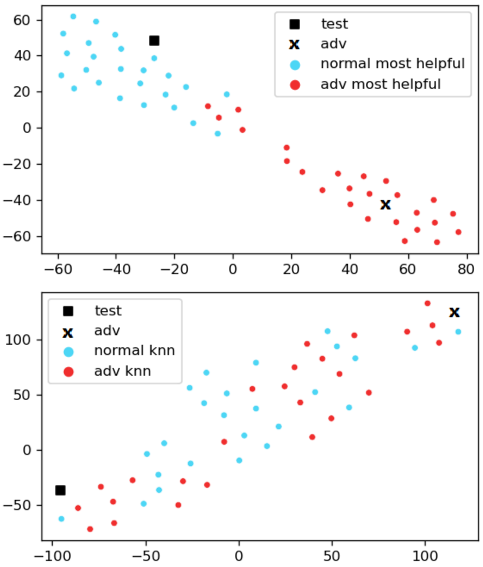


Figure 3: Normal and adversarial train subspace observed on the IMDB record used in Fig 2 under WordAtt by influence function (top) and DkNN (bottom)

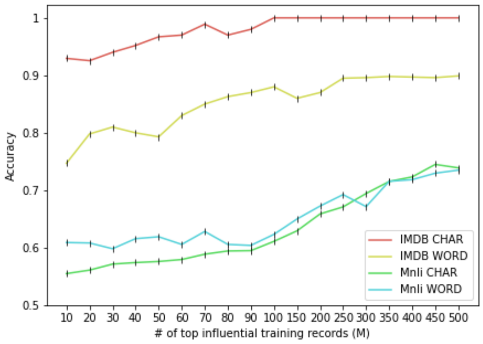


Figure 4: Accuracy of NNIF for different values of M.

generally much more clearly separable and so IF points contribute especially strongly to the method, except for MultiNLI against WordAtt, where they are essentially the same and the method relies on the two-view aspect of NNIF. This observation about the relative importance of the IF contribution was not made by Cohen et al. (2020), and so may be specific to NLP tasks, although this would require more investigation to verify. We also note that our results align with observations of Han et al. (2020), that in the harder task of MultiNLI (§5.1, Table 4), IFs provide a different perspective to characterising the datapoint of interest. We give some text examples in App E.

To look further into the more challenging combination of MultiNLI and CharAtt (as the one case in Table 2 where NNIF was not the highest scoring, albeit by a small margin), we consider a successful and an unsuccessful detection case by NNIF, with the actual examples given in the appendices in Tables 11 and 12, and the corresponding t-SNE plots of IF and NNs in Figs 5 and 6, respectively. The IFs in Fig 5 (the correct example) are somewhat more clustered, with the red (adversarial) points mostly in the top right, than the IFs in Fig 6 (the incorrect example); this lines up with the results of Table 4 in that separability of IF does seem to matter for MultiNLI +CharAtt.

Attack	Avg Acc NNIF	Avg Acc kNN	p-value
IMDB CharAtt	0.6875	0.5626	$<$ .00001
IMDB WordAtt	0.7812	0.5644	$<$ .00001
MultiNLI$_{\text{CharAtt}}$	0.6399	0.5625	$<$ .00001
MultiNLI$_{\text{WordAtt}}$	0.5603	0.5632	0.448

Table 4: SVC accuracy of linearly separating the 2D-t-SNE embedding subspace of neighboring train samples of 1000 test records and their adversarial versions

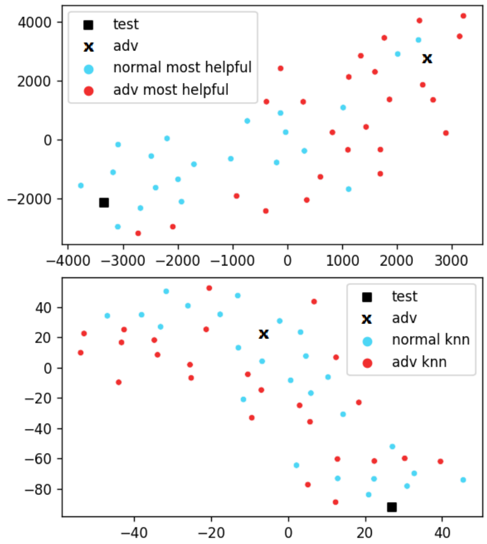


Figure 5: Normal and adversarial subspace of the MultiNLI CharAtt text in Table 10 by IF (top) and DkNN (bottom)

Varying M in NNIF Fig 4 plots the accuracies of the NNIF method for both tasks and attacks, for a range of values of M. The accuracy broadly monotonically increases until plateaus for the IMDB results, although the MultiNLI results look to be still increasing. This is a contrast with the image processing results of Cohen et al. (2020), where much smaller values of M (e.g. 30) produced better results. It is unclear what characteristics of our tasks (fewer classes, more long-distance dependencies, . . .) lead to this difference.

Ablation for Mahal Table 3 shows the accuracies of Mahal using only the final layer or the feature ensemble. As with Lee et al. (2018), the feature ensemble produces much better results. The im-

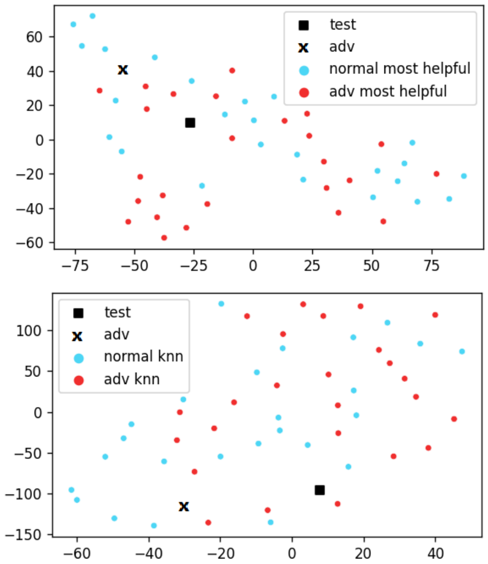


Figure 6: Normal and adversarial subspace by IF (top) and DKNN (bottom) on the unsuccessful detection by NNIF of the MULTINLI CHARATT text in Table 12

provement is larger for IMDB, but still important for MULTINLI, as without the ensemble, detection is essentially at the chance. Noting that the target model of Lee et al. (2018) had many more hidden layers in the ensemble, it is an open question as to whether introducing additional dense layers into our LLM-based model might improve detection while still preserving target model performance.

6 Conclusion and Future Work

We have adapted from image processing two methods, NNIF (Cohen et al., 2020) and MAHAL (Lee et al., 2018), that detect adversarial examples using learned representations. Both perform strongly, with NNIF the best on three of four task/attack combinations, and a close second on the fourth, against several strong baselines.

Our analysis shows that influence function points make a particularly important contribution to the NNIF method. The MULTINLI task is more challenging for all methods; here it is the complementary nature of information from influence functions and nearest neighbors, supporting observations by Han et al. (2020) about the different perspective of influence functions in this more complex NLP task.

The NNIF method is computationally expensive, so future work will look at ways to make it more efficient. Additionally, to gain a fuller understanding of what information influence functions can provide in NLP tasks, future work will look at a wider range of tasks and attacks.

7 Limitations

The major limitation is the computationally expensive calculation of influence functions in our NNIF method. For this, following Cohen et al. (2020) we restrict the data size to 10k (5k test, 5k adversarial) for NNIF and follow a similar approach for other methods for comparability. This helps faster explanation generation in SHAP as well. We use a small architecture as recommended in Han et al. (2020) for the BERT$_{\text{BASE}}$ model for NNIF and other detectors. As noted in the paper, we recognise that there is the FASTIF method of Guo et al. (2021) for speeding up influence function calculation, but because of the restriction of influence function points to nearest neighbors, it is not suitable for our application.

We use only two datasets/tasks and two attack methods, partly because of the computational expense of NNIF. While they are commonly used in the adversarial example literature as well as the analysis of influence functions in NLP by Han et al. (2020) and represent different levels of task complexity and attack type, a wider range of datasets/tasks and attack methods is needed for a full characterisation of influence functions and the nature of adversarial subspaces.

For all experiments, we restrict the maximum sequence length following Liu et al. (2022), which may influence the detectors' performance, especially for the NLI task, that requires the model to learn from a hypothesis and premise text pairs.

For the detector baselines, we used the most available methods. There are two recent contemporaneous methods by Wang et al. (2022) and (Bao et al., 2021) that explore the idea that adversarial perturbations are typically rare-frequency words, and create augmented training sets by replacing those words in each sentence with synonyms. For the detection, Wang et al. (2022) matches the voted prediction with the obtained prediction and (Bao et al., 2021) trains the model on a separate auxiliary learning objective. Between these two works, we choose the RSV from Wang et al. (2022) in our work. For RSV, we follow the similar setting from Wang et al. (2022) in choosing the vote number, word substitution rate and stop word selection for both IMDB and MULTINLI. A different setting for MULTINLI may improve the result.

References

Naman Agarwal, Brian Bullins, and Elad Hazan. 2017. Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*, 18:116:1–116:40.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitutions without performance sacrifice. *CoRR*, abs/2105.14553.

Nicholas Carlini and David A. Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639. ISCA.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2020. Detecting adversarial samples using influence functions and nearest neighbors. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14441–14450. Computer Vision Foundation / IEEE.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. 2017. Detecting adversarial samples from artifacts. *CoRR*, abs/1703.00410.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Alex Barber, editor, *ELL*, pages 2–665. Elsevier.

Gil Fidel, Ron Bitton, and Asaf Shabtai. 2020. When explainability meets adversarial learning: Detecting adversarial examples using SHAP signatures. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–8. IEEE.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Divya Gopinath, Guy Katz, Corina S. Pasareanu, and Clark W. Barrett. 2018. Deepsafe: A data-driven approach for assessing robustness of neural networks. In *Automated Technology for Verification and Analysis - 16th International Symposium, ATVA 2018, Los Angeles, CA, USA, October 7-10, 2018, Proceedings*, volume 11138 of *Lecture Notes in Computer Science*, pages 3–19. Springer.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Han Guo, Nazneen Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. 2021. FastIF: Scalable influence functions for efficient model interpretation and debugging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10333–10350, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. 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 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Compu-*

tational Linguistics, pages 2752–2765, Online. Association for Computational Linguistics.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. 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.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2016. Learning robust representations of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1979–1985, Austin, Texas. Association for Computational Linguistics.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 21–27, Valencia, Spain. Association for Computational Linguistics.

Na Liu, Mark Dras, and Wei Emma Zhang. 2022. Detecting textual adversarial examples based on distributional characteristics of data representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 78–90, Dublin, Ireland. Association for Computational Linguistics.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582. IEEE Computer Society.

Edoardo Mosca, Lukas Huber, Marc Alexander Kühn, and Georg Groh. 2022. Detecting word-level adversarial text attacks via shapley additive explanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 156–166. Association for Computational Linguistics.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, et al. 2016a. Technical report on the cleverhans v2. 1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*.

Nicolas Papernot and Patrick D. McDaniel. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *CoRR*, abs/1803.04765.

Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016b. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. 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, Sameer Singh, and Carlos Guestrin. 2018. 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)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Thomas Tanay and Lewis D. Griffin. 2016. A boundary tilting persepective on the phenomenon of adversarial examples. *CoRR*, abs/1608.07690.

Florian Tramèr, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. 2017. The space of transferable adversarial examples. *CoRR*, abs/1704.03453.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Xiaosen Wang, Yifeng Xiong, and Kun He. 2022. Detecting textual adversarial examples through randomized substitution and vote. In *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pages 2056–2065. PMLR.

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

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. 2019. Feature denoising for improving adversarial robustness. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 501–509. Computer Vision Foundation / IEEE.

Jin Yong Yoo and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. 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 4904–4913, Hong Kong, China. Association for Computational Linguistics.

A Experimental Setup Details

The size of the datasets and the number of adversarial samples generated by each of the attack methods are given in Tab. 5. Obtained accuracies of the $\text{BERT}_{\text{BASE}}$ model are in Tab. 6 and the other models used in MDRE are in Tab. 7

A.1 Attack Methods

CHARATT. We implement CHARATT as proposed by Pruthi et al. (2019). It tweaks the original texts by randomly swapping, dropping and adding characters or adding a keyboard mistake. *Swapping* refers to exchanging places of two adjacent internal characters. *Dropping* removes a character and *Adding* inserts a new character at a randomly selected position. *Keyboard mistakes* is for substituting a character with one of its adjacent characters in keyboards.

In our experiments, we allow a maximum of half the words from the original text to be perturbed, so the maximum number of possible attacks on the IMDB and MULTINLI datasets is 256 and 128 per sentence, respectively.

WORDATT. Alzantot et al. (2018) proposed an effective and widely used adversarial attack that we incorporate in our work as WORDATT.

This method allows the attacker to alter practically every word from the sentence if required with the context-preserving synonymous words. The synonym search is done over a large search space that includes the GloVe word vectors (Pennington et al., 2014), counter-fitting word vectors (Mrkšić et al., 2016), and the Google 1 billion words language model (Chelba et al., 2014). Then, following the natural selection methods, crossover and mutation techniques from the population-based genetic algorithm are applied to generate the next set of adversarial sentences. On each iteration, several adversarial texts that are unsuccessful in changing the model's prediction are removed from the pool.

However, Jia et al. (2019) found that the algorithm is computationally expensive and recommended using a faster language model and stopping the semantic drift of the algorithm that refers to applying the language model on the synonyms picked from previous iterations as well to choose words from their neighboring word-space.

We incorporate the above recommendations by utilising a faster Transformer-XL architecture (Dai et al., 2019) that is pretrained on the WikiText-103 dataset (Merity et al., 2017) and prohibiting the semantic drift by finding all test examples words' neighbors only before attacks. We also restrict the minimum number of perturbations to one-fifth of the maximum sequence length which is 102 and 51 for the IMDB and MULTINLI, respectively.

A.2 Baseline Detection Methods

The first four are from Liu et al. (2022) (we omit the language model, as it operates essentially at the chance), and we use the implementations from there.[4]

Learning to Discriminate Perturbations (DISP) (Zhou et al., 2019). DISP is one of the commonly used baselines for adversarial text detection that identifies a set of character-level of word-level perturbed tokens and then applies an embedding estimator that predicts embeddings for each perturbed token and maps them to the actual word to repair the perturbations.

If the model's prediction on an adversarial text restored by DISP remains the same class as the prediction on its original version, we consider it a successful detection of an adversarial example.

Frequency-guided word substitutions (FGWS) (Mozes et al., 2021). Mozes et al. (2021) verifies that in the case of word-level attacks, the synonym replacements normally occur in low frequency. They use this concept in a model-agnostic rule-based adversarial text detection algorithm Frequency-Guided Word Substitutions (FGWS).

Firstly, the algorithm sets a word frequency threshold to identify infrequent words that have frequencies lower than this value. Then the algorithm replaces those words with their high-frequency synonyms and selects the replaced sentences as adversarial samples if the model's prediction confidence scores for the replacements change over a threshold. They use WordNet (Fellbaum, 2005) and GloVe vectors (Pennington et al., 2014) to find the synonyms. They experiment by taking {*0*-th, *10*-th, $\cdots$, *100*-th} percentile of word frequencies in the training set as the word-frequency threshold. Finally, on these selected alternative sentences, if the prediction confidence differs from their corresponding original sentence's prediction confidence by more than a certain amount, the original sentences are determined as adversarial examples.

Local Intrinsic Dimensionality (LID) (Liu et al., 2022). From the image processing detection methods, Liu et al. (2022) adapt the Local Intrinsic

[4] https://github.com/NaLiuAnna/MDRE

Dataset	Training.	Validation.	Testing.	Correctly Predicted Test Examples	Adversarial/Original Examples	
					character-level	word-level
IMDB	20,000	5,000	25,000	23,226	12,299	9,627
MULTINLI	314,162	78,540	9,832	8,062	7,028	3,240

Table 5: The number of examples used in experiments

Dataset	Clean Accuracy	CHAR ATTACK	WORD ATTACK
IMDB	0.9290	0.3656	0.6999
MULTINLI	0.8201	0.4848	0.6864

Table 6: BERT$_{\text{BASE}}$ classifier accuracy on the clean and adversarial examples

Dimensionality (LID) approach of Ma et al. (2018). This technique creates a local distance distribution for a test record to its neighbors from the training set. They apply this to transformer models by taking the outputs of each layer from the target model to represent the training records.

Following Liu et al. (2022), we use the BERT$_{\text{BASE}}$ model and implement a logistic regression classifier as the detector, and tune the size of the neighbors k through a grid search over 100, 1000, and the range [10, 42) with a step size 2.

MultiDistance Representation Ensemble Method (MDRE) (Liu et al., 2022). Motivated by the notion that adversarial examples are out-of-distribution samples as recognized in Lee et al. (2018) and Feinman et al. (2017), Liu et al. (2022) assume that texts with the same prediction label lie on similar data submanifold and adversarial perturbation on these texts put them to another data submanifold, thus altering the model's prediction on them.

They measure the Euclidean distance between each reference datapoint and the nearest neighbors from the training datapoints with similar predicted labels and establish that this distance will be greater for the adversarial reference point than the normal one. They further use ensemble learning to combine distances between representations learned from multiple DNNs and build a binary logistic regression model to detect adversarial examples.

Following (Liu et al., 2022), we also use four learning models: [BERT$_{\text{BASE}}$, RoBERTa$_{\text{BASE}}$, XLNet$_{\text{BASE}}$, BART$_{\text{BASE}}$] in our experiments. Table 7 reports the clean accuracies of the other target classifiers used in feature ensembling in MDRE.

Randomized Substitution and Vote (RSV) (Wang et al., 2022). A word-level attacker's target is to find an optimal synonym substitution that mutually influences other words in the sentence. Taking this optimization target of the adversary, Wang et al. (2022) resort to randomly substituting words from the text with their synonyms and argue that this random word substitution destroys the mutual interaction between words and eliminates adversarial perturbation.

At first, they generate a set of perturbed samples by randomly replacing some words from a text with their arbitrary synonyms. Then the model's output logits for the processed samples are accumulated and voted to determine a prediction label for the text samples. If the original text's prediction doesn't match the voted prediction label it is considered as an adversarial example.

We use their code.[5]

SHapley Additive exPlanations (SHAP) (Mosca et al., 2022). In this work, Mosca et al. (2022) adopt an adversarial image detection method for word-level attacks on text. They train an adversarial detector with the SHapley Additive exPlanations (SHAP) values of the training data for each of the test data using the SHAP explainer proposed and implemented by Fidel et al. (2020).

They experiment on multiple classifiers as the detectors such as logistic regression, random-forest classifier, support vector classifier and a neural network. They also show that the detector doesn't require a large number of training samples for it to be successful. In our work, we follow the same and report the best accuracy obtained among the four detectors.

We use their code.[6] Accuracies of all the detectors are in Table 8.

B Computing Influence Function

For a datapoint $z_i = (x_i, y_i)$ from the training set $\{(x_1, y_1), \ldots, (x_i, y_i) \in (X, Y)\}$ and model parameters $\theta \in \Theta$, the loss of the model be $L(z, \theta)$

[5] https://github.com/JHL-HUST/RSV

[6] https://github.com/huberl/adversarial_shap_detect_Repl4NLP

Dataset	Attack Method	BERT$_{\text{BASE}}$	RoBERTa$_{\text{BASE}}$	XLNet$_{\text{BASE}}$	BART$_{\text{BASE}}$
IMDB	Clean	0.9290	0.9532	0.9336	0.9429
	CharAtt	0.3656	0.8613	0.5770	0.8286
	WordAtt	0.6999	0.8714	0.7918	0.8425
MultiNLI	Clean	0.8201	0.8671	0.8630	0.8455
	CharAtt	0.4848	0.7104	0.6670	0.6457
	WordAtt	0.6864	0.7068	0.6870	0.6296

Table 7: Different classifier accuracies on both clean and adversarial dataset for MDRE.

Dataset	Attack	Logistic Regression	Random Forest	SVC	DNN
IMDB	CharAtt	0.740	0.804	0.803	0.812
	WordAtt	0.605	0.764	0.684	0.75
MultiNLI	CharAtt	0.588	0.614	0.613	0.61
	WordAtt	0.528	0.697	0.633	0.621

Table 8: Detection accuracy obtained from four detector classifiers used in SHAP.

and the optimized parameters are:

$$\theta' = \underset{\theta \in \Theta}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$$

The influence score is then calculated by observing the impact of a modification in the weight of a train datapoint on the decision of the prediction for the test datapoint. Assume we upweigh the training datapoint z by a small ϵ amount, which produces below θ':

$$\theta'_{\epsilon,z} = \underset{\theta \in \Theta}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

Then, according to Koh and Liang (2017), the influence of the boosted z on the parameters θ' can be defined by:

$$\frac{d\theta'_{\epsilon,z}}{d\epsilon}|_{\epsilon=0} = -H_{\theta'}^{-1} \nabla_\theta L(z, \theta') \qquad (1)$$

where $H'_\theta = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \theta')$ is the Hessian of the model.

Applying the chain rule to the Eq. 1 can be derived to the below form that measures the influence $I_{up,loss}$ of z on the loss of a test point z_{test}:

$$I_{up,loss}(z, z_{test}) = - \nabla_\theta L(z_{test}, \theta') H_\theta^{-1} \nabla_\theta L(z, \theta') \qquad (2)$$

The NNIF method uses the $I_{up,loss}$ score.

Parameters	Values
C	[1, 10, 1000, 10000]
gamma	[1, .1, .01, 0.001, 'auto']
kernel	['linear', 'rbf', 'poly', 'sigmoid']

Table 9: Gridsearch parameters for building SVC.

C Illustrations of Regions Around Test and Adversarial Points

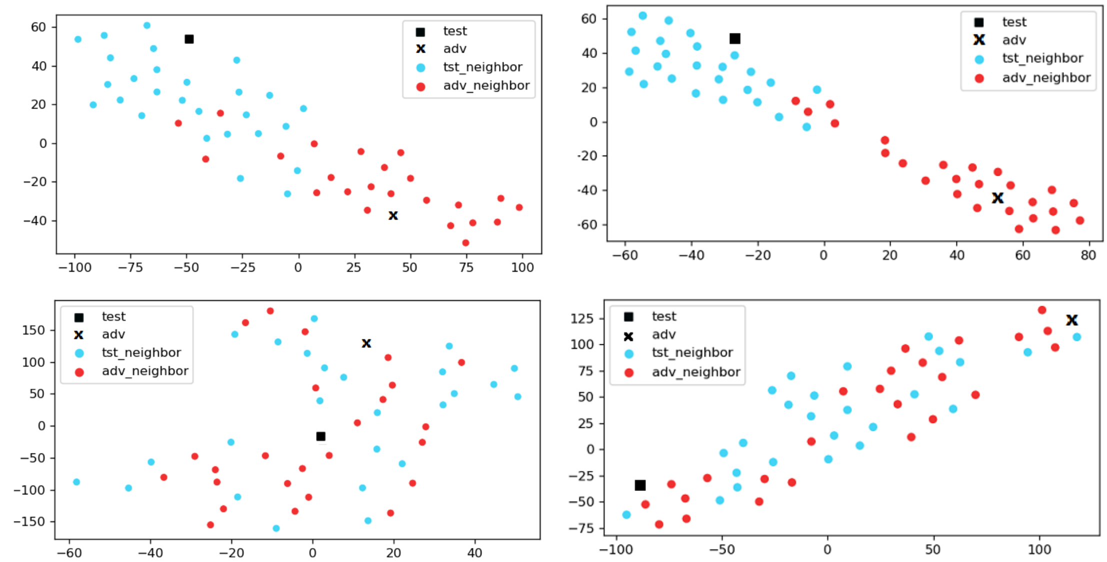
Looking at the training samples that influence the prediction of a test datapoint, gives us an illustration of the decision subspace of the DNN on it. To illustrate the subspace, we measure the top 25 influential (IF) and nearest neighbor (NN) training embeddings for a test record and its adversarial counterpart for each attack and plot them along with the test and adversarial points. All embeddings are reduced to two dimensions by using t-SNE. Figures 7 and 8 show an example each for the IMDB and MultiNLI datasets, respectively. On each figure, the top row depicts the IF-based training points and the bottom row shows the NN-based training points.

D Separability of Points: IF vs NN

We build SVC classifiers on the neighboring train embeddings to evaluate how well the influence function is describing the learned subspace of the DNN than the DkNN. The best SVC classifiers over NNs and IF points for each of the 1000 test and adversarial example pairs are estimated through GridSearch over the parameters as depicted in Table 9.

E Experimental Results Examples

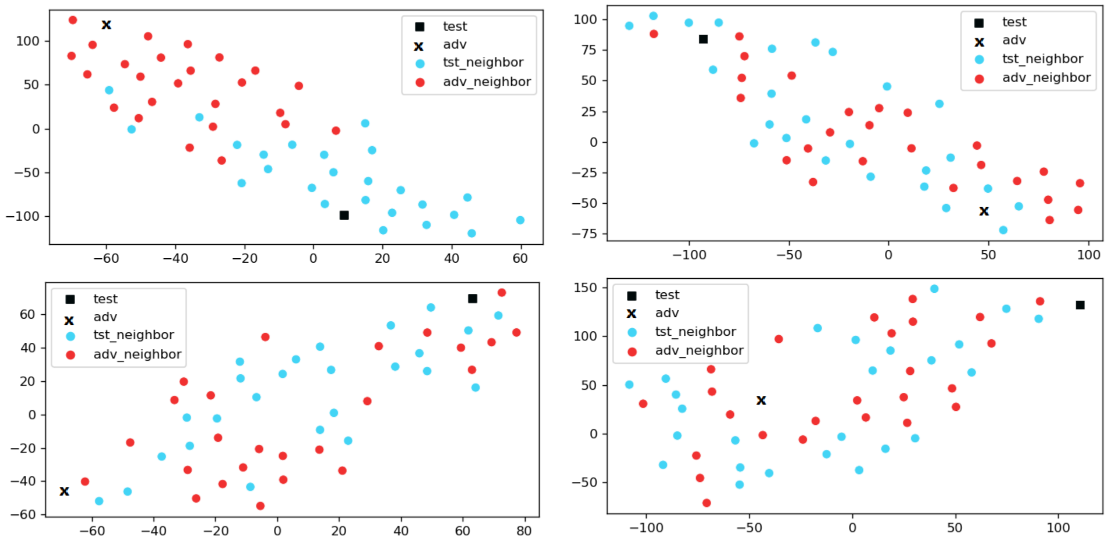
NNIF combines the DkNN ranking on top of the influence scores to select the best training instances for a test datapoint. In Tables 10, 11 and 12 we illustrate examples for WordAtt and CharAtt respectively, showing the top three helpful and harm-



(a) CHARATT

(b) WORDATT

Figure 7: Embedding subspace (applied t-SNE) of a test sample from the IMDB dataset (black-square) and its adversarial version (purple-cross) generated by three types of attacks. The top set of images shows the 25 most influential training samples and the bottom set shows the top 25 nearest neighbors (KNN).



(a) CHARATT

(b) WORDATT

Figure 8: Embedding subspace (applied t-SNE) of a test sample from the MULTINLI dataset (black-square) and its adversarial version (purple-cross) generated by three types of attacks. The top set of images shows the 25 most influential training samples and the bottom set shows the top 25 nearest neighbors (KNN).

ful training instances for the detection of the adversarial attack. We also show the DKNN rankings of the top training instances filtered by the IF scores in the table.

As DISP performs better in one of the experimental settings in Liu et al. (2022), we further pick one example sentence from the paper that DISP detects correctly and observe NNIF's performance on it. NNIF is also able to detect the sentence correctly. In Table 13 we show the influential instances for this prediction as well.

Original text - label Entailment - prediction Entailment		
Premise: Address your remarks to the chair illustrates metonymy a figure of speech in which something is called by the name of something else associated with it. **Hypothesis:** Using one word to refer to something that is associated with it is a figure of speech.		
WordAtt- prediction Contradiction		
Premise: Address your remarks to the chair draws metonymy a digit of speech in which something is called by the name of something else associated with it . **Hypothesis:** Using one word to refer to something that is associated with it is a digit of speech		
Top Helpful		**NNIF Rank**
1	**Premise:** my adult women friends are anguishing over over some of these choices. **Hypothesis:** I don't have any woman friends.	2185
2	**Premise:** Vrenna, now! Jon kicked the barrel and it broke open. **Hypothesis:** Jon told Vrenna what to do.	4956
3	**Premise:** I brought my Gauntlet to bear; electricity leaping out. **Hypothesis:** My gauntlet was magical and electricity jumped out of it.	5473
Top Harmful		**NNIF Rank**
1	**Premise:** Examining the elements of the definition also may help make this distinction clear. **Hypothesis:** Ignoring the elements of the definition also may help make this distinction clear.	4308
2	**Premise:** the uniformed services, recognize that promotional material received by a uniformed service member traveling on official business at government expense belongs to the government and must be relinquished in accordance with service regulations. **Hypothesis:** The material belongs to the government even after the hand out.	5819
3	**Premise:** The emergency department at Harborview probably sees 50 times as many patients with alcohol problems as the psychiatry or family medicine departments. **Hypothesis:** The family medicine departments do not treat alcohol problems.	2221

Table 10: Top three helpful and harmful train instances based on IF score and further ranking of them by DkNN for a correctly predicted adversarial text by NNIF for MultiNLI WordAtt

Original text - label Entailment - prediction Entailment		
Premise: Although a seemingly mundane, tactical aspect of business, a firm's inventory strategy reflects its approach to managing risk. **Hypothesis:** It is possible to determine a firm's risk management philosophy by examining their inventory strategy.		
CharAtt- prediction Neutral		
Premise: Although a seemingly mundane , tactcial aspect of business , a firm 's itnventory strategxy reflects its approach to managing risk. **Hypothesis:** It is possible to determine a firm 's risk management philkosophy by examining their inventory strategxy.		
Top Helpful		**NNIF Rank**
1	**Premise:** Online investment guru Tokyo Joe was sued by the SEC in a civil fraud case. **Hypothesis:** Tokyo Joe has been sued before.	5091
2	**Premise:** Wesray's purchase of Avis was trendy in three ways. **Hypothesis:** There are three reasons why Wesray's purchase of Avis is trendy.	266
3	**Premise:** yeah i don't mind that um my husband never cared for fast food so we didn't go that often but you know i have no problem with uh going to a McDonald's or a Wendy's. **Hypothesis:**My husband and I did not eat that much fast food.	5630
Top Harmful		**NNIF Rank**
1	**Premise:** In accordance with the prescribed statutory process, on August 17, 2001, we reported to the Congress, the President, the Vice President, and other officials that the NEPDG had not provided the requested records. **Hypothesis:** We told Congress that the NEPDG had failed to give us the records.	1346
2	**Premise:** The case for not acting until you have to was put most vividly by Senate Assistant Majority Leader Don Nickles of Oklahoma, in a remark that also captures the hard-nosed attitude regarding humanitarian concerns. **Hypothesis:** The statement summarized their sentiment.	1044
3	**Premise:** Predicting that he would get a lot of heat for treating the minister with respect, Novak said that Farrakhan was more measured and a lot less confrontational and provocative than a lot of the politicians we talk to regularly on this program. **Hypothesis:** Predicting he would get a lot of hear for respecting the minister, Novak said Farrakhan was measured and less confrontational.	5623

Table 11: Top three helpful and harmful train instances based on IF score and further ranking of them by DkNN for a correctly predicted adversarial text by NNIF for MultiNLI CharAtt

Original text - label Neutral - prediction Neutral		
Premise: And, instead of providing an open-ended guarantee on prices to its distributors, the company would guarantee the price for only two weeks after purchase by the distributor, refusing to take back computers unless they malfunctioned. **Hypothesis:** The distributor could potentially lose out due to this method.		
CharAtt- prediction Contradiction		
Premise: And , instead of providing an openedned guaantee on pices to its disrtibutors , the comapny wuld guaantee the price for only two weeks after purchase by the distributor, refusing to take badk computers ulness they malfunctioned . **Hypothesis:** Tehe distributor could poteIntially lose out de to this mehgod .		
Top Helpful		**NNIF Rank**
1	**Premise:** No it was gas because you washed your legs all over because you did it in shorts. **Hypothesis:** Your legs were washed all over due to having done it in shorts.	2373
2	**Premise:** Leaving the British official who twice searched his luggage none the wiser, he managed by meticulous observation to memorize the principal features of the power loom well enough to produce his own version of it on his return to Boston. **Hypothesis:** He failed at retaining the information in his head but managed to build a rough prototype of the power loom anyway.	4706
3	**Premise:** Bin Ladin shares Qutb's stark view, permitting him and his followers to rationalize even unprovoked mass murder as righteous defense of an embattled faith. **Hypothesis:** Bin Ladin views his actions as a defense of his faith.	925
Top Harmful		**NNIF Rank**
1	**Premise:** As graduates of the class of 1990, we would like to leave behind something tangible, in appreciation for the support and encouragement we have received from other students in the School of Engineering and Technology. **Hypothesis:** We want leave a concrete symbol of our appreciation to the school.	1336
2	**Premise:** Fortunately, not all reports are as disturbing as Hochschild's. **Hypothesis:** Thankfully, not all reports are as terrifying as Hochschild's.	4817
3	**Premise:** Of the two, the W geographical listings seem more W lists Aylesbury, which, through some grievous, egregious fault, is not in the geographical section of the L but does appear in the A-Z section (because of the ducks). **Hypothesis:** For some reason, the ducks put the topic in the A-Z section.	268

Table 12: MultiNLI CharAtt adversarial text that the NNIF fails to detect; showing top three helpful and harmful train instances based on IF score and further ranking of them by DkNN

Original text - label Entailment prediction Entailment		
Premise: Finally, it might be worth mentioning that the program has the capacity to store in a temporary memory buffer about 100 words (proper names, for instance) that it has identified as not stored in its dictionary. **Hypothesis**: It's possible to store words in a temporary dictionary, if they don't appear in a regular dictionary.		
WORDATT- prediction Neutral		
Premise: Finally, it might be worth mentioning that the program has the capacity to store in a temporary memory buffer about 100 words (proper names, for instance) that it has identified as not stored in its dictionary. **Hypothesis**: It's possible to shopping words in a temporary dictionary, if they don't appear in a regular dictionary.		
Repaired text by DISP		
Premise: Finally, it might be worth that that the program has the capacity to store in a temporary memory buffer about 100 words (proper names, for instance) that it has identified as not stored in its dictionary. **Hypothesis**: It's possible to **do** words in a temporary dictionary, if they don't appear in a regular dictionary.		
Repaired text by FGWS		
Premise: Finally, it might be worth **name** that the program has the capacity to store in a temporary memory pilot about 100 words (proper names, for instance) that it has identified as not stored in its dictionary. **Hypothesis**: It's possible to shopping words in a temporary dictionary, if they don't appear in a regular dictionary.		
Top Helpful		**NNIF Rank**
1	**Premise:** yeah i mean they're they're throwing more money at it now than ever before and things are getting worse. **Hypothesis:** The money is going to the wrong things, so it's not fixing the problem.	1295
2	**Premise:**I put $75 on the New England Patriots as a 2.5-point underdog and $50 on a Boston Red Sox playoff game against the Cleveland Indians. **Hypothesis:** I bet a total of $125 dollars on the New England Patriots and the Boston Red Sox.	3870
3	**Premise:** Graffiti written by Russian soldiers can be seen in the caves of Antiparos. **Hypothesis:** Russian soldiers drew graffiti on the walls of The Louvre.	4122
Top Harmful		**NNIF Rank**
1	**Premise:** But I am most serious. **Hypothesis:** I'm not joking at all.	808
2	**Premise:** the uniformed services, recognize that promotional material received by a uniformed service member traveling on official business at government expense belongs to the government and must be relinquished in accordance with service regulations **Hypothesis:** TThe material belongs to the government even after the hand out.	2835
3	**Premise:** According to NIST, accreditation is the formal authorization by the management official for system operation and an explicit acceptance of risk. **Hypothesis:** Accreditation is the formal authorization by the management official for system operation and an explicit acceptance of risk, according to NIST.	2908

Table 13: Example sentence from Liu et al. (2022) that is predicted correctly by NNIF, DISP, LID, MDRE and incorrectly by FGWS