Balanced Adversarial Training: Balancing Tradeoffs Between Oversensitivity and Undersensitivity in NLP Models

Anonymous ACL submission

Abstract

Traditional (*oversensitive*) adversarial examples involve finding a small perturbation that does not change an input's true label but confuses the classifier into outputting a different prediction. *Undersensitive* adversarial examples are the opposite—the adversary's goal is to find a small perturbation that changes the true label of an input while preserving the classifier's prediction. Adversarial training and certified robust training have shown some effectiveness in improving the robustness of machine learnt models to oversensitive adversarial examples. However, recent work has shown that using these techniques to improve robustness for image classifiers may make a model more vulnerable to undersensitive adversarial examples. We demonstrate the same phenomenon applies to NLP models, showing that training methods that improve robustness to synonym-based attacks (oversensitive adversarial examples) tend to increase a model's vulnerability to antonym-based attacks (undersensitive adversarial examples) for both natural language inference and paraphrase identification tasks. To counter this phenomenon, we introduce *Balanced Adversarial Training* which incorporates contrastive learning to increase robustness against both over- and undersensitive adversarial examples.

1 Introduction

At the broadest level, an adversarial example is an input crafted intentionally to confuse a model. Most research on adversarial examples, however, focuses on a formal definition of an adversarial example as an inputs that is constructed by making minimal perturbations to a normal input which change the model's output, assuming that the small perturbations preserve the original true label (Goodfellow et al., 2015). This happens when the model is overly sensitive towards small changes in the input, so we refer to these as *oversensitive adversarial examples*. In NLP, synonym-based word substitution is a common method for constructing oversensitive adversarial examples (Alzantot et al., 2018; Jin et al., 2020).

Attackers can also target the opposite objective—to produce inputs with minimal but meaningful changes that flip the ground truth label but make the model retain its prediction (Jacobsen et al., 2019). This type of attack is known as an *undersensitive adversarial example*. It targets a model's weakness of being invariant to certain types of changes which make it insufficiently sensitive to change its prediction in response to changes in input. Attacks based on antonyms and negation have been proposed to create undersensitive adversarial examples for dialogue models (Niu and Bansal, 2018).

Recent work in the vision domain demonstrated that increasing adversarial robustness of image classification models by training with oversensitive adversarial examples may increase vulnerability to undersensitive adversarial examples (Tramer et al., 2020). Even in cases where the model certifiably guarantees that no adversarial examples can be found within the L_p-bounded distance, the norm-bounded perturbation does not align with the ground truth decision boundary. This *distance-oracle misalginment* makes it possible to have undersensitive adversarial examples located within the same perturbation distance, as depicted in Figure 1. Similarly, in text, oversensitive examples are usually generated with cosine similarity constraint to encourage the representations of the original and the perturbed sentence to be close in the embedding space. However, this similarity measurement may not preserve the actual semantics (Morris et al., 2020) and the model may learn poor representation during adversarial training.

Contributions. In this work, we study adversarial robustness tradeoffs in NLP models. While it is challenging to construct an automatic undersensitivity attack for image classifiers, we show that we are able to automate the process for NLP mod-

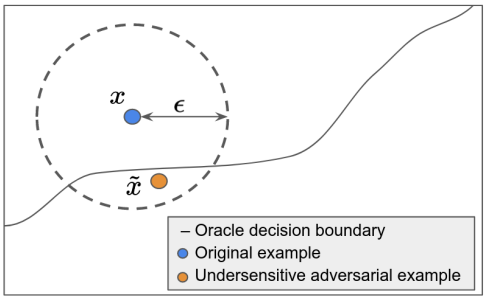


Figure 1: Distance-oracle misalignment. While the model is trained to be robust to ϵ-bounded perturbation, it becomes too invariant to small changes in the example (undersensitive example $\tilde{x}$) that actually lie on the other side of the oracle decision boundary.

els. We evaluate this robustness tradeoff on natural language inference and paraphrase identification tasks with BERT and RoBERTa models and show that while certified robust training increases robustness against oversensitive adversarial examples, it introduces vulnerability towards under-sensitivity attacks (Section 3). We use synonym-based attack for constructing oversensitive adversarial examples, and antonym-based attacks for constructing undersensitive adversarial examples (Figure 2 shows a few examples). We show that the antonym attack success rate increases as the model becomes more robust against synonym based attacks (Section 3.3). We also propose a modification to robust training, *Balanced Adversarial Training* (BAT), which utilizes a contrastive learning objective to help mitigate the distance misalignment problem by learning from both oversensitive and undersensitive examples (Section 4). We implement with two different contrastive learning objectives including pairwise and triplet loss and show the effectiveness in improving both oversensitivity and undersensitivity robustness (Section 4.2).

2 Constructing Adversarial Examples

We consider a classification task where the goal of the model f is to learn to map the textual input x, a sequence of words, $x_1, x_2, ..., x_L$, to its ground truth label $y \in \{1, ..., c\}$. We assume there is a labeling oracle $\mathcal{O}$ that corresponds to ground truth and outputs the true label of the given input. We focus on word-level perturbations where the attacker substitutes words in the original input x with words from a known perturbation set (which we show how we construct it in the following sections). The goal of the attacker is to find an adversarial example $\tilde{x}$ for input x such that the output of the model is different from what human would interpret, i.e. $f(\tilde{x}) \neq \mathcal{O}(\tilde{x})$.

2.1 Oversensitive Adversarial Examples

For a given input (x, y) correctly classified by model f and a set of allowed perturbed sentences $\mathcal{S}_x$, an *oversensitive adversarial example* is defined as an input $\tilde{x}_{over}$ such that:

1. $\tilde{x}_{over} \in \mathcal{S}_x$
2. $f(\tilde{x}_{over}) \neq f(x)$
3. $\mathcal{O}(\tilde{x}_{over}) = \mathcal{O}(x)$

There are many different methods for finding oversensitive adversarial examples. The most common way is to use synonym word substitutions where the target words are replaced with similar words found in the word embedding (Alzantot et al., 2018; Jin et al., 2020) or use known synonyms from WordNet (Ren et al., 2019). Recent work have also explored using masked language models to generate word replacements (Li et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2021).

We adopt the similar synonym word substitution method in Ye et al. (2020). For each word x_i in an input x, we create a synonym set S_{x_i} containing the synonym words of x_i including itself. $\mathcal{S}_x$ is then constructed by a set of sentences where each word in x can be replaced by a word in S_{x_i}. We consider the case where the attacker does not have a constraint on the number of words that can be perturbed for each input, meaning the attacker can perturb up to L words which is the length of x.

The underlying assumption for oversensitive examples to work is that the perturbed sentence $\tilde{x}_{over} \in S_x$ should have the same ground truth label as the original input x, i.e. $O(\tilde{x}_{over}) = O(x) = f(x)$. However, common practice for constructing oversensitive examples does not guarantee this is true. Swapping a word with its synonym may change the semantic meaning of the example since even subtle changes in words can have a big impact on meaning, and a word can have different meanings in different context. For instance, "the whole *race* of human kind" and "the whole *competition* of human kind" describe different thing. Nonetheless, previous human evaluation have shown that synonym-based adversarial examples still retain the same semantic meaning and label as the original texts most of the time (Jin et al., 2020; Li et al., 2020).

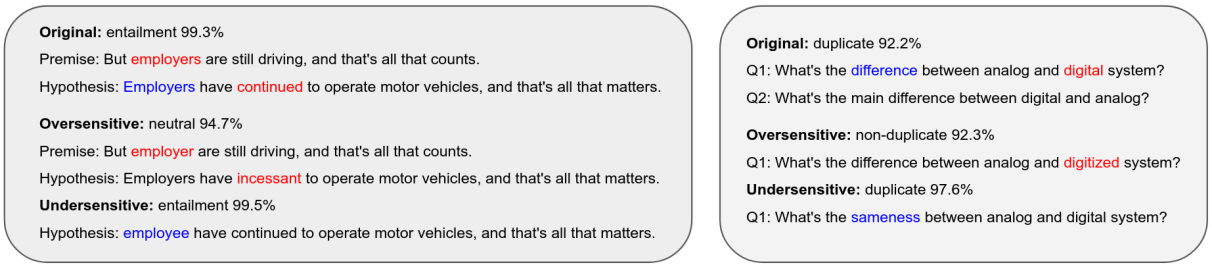


Figure 2: Oversensitive and undersensitive adversarial examples for BERT model fine-tuned on natural language inference (left) and paraphrase identification (right) tasks.

2.2 Undersensitive Adversarial Examples

For a given input (x, y) correctly classified by model f and a set of allowable perturbed sentences $\mathcal{A}_x$, an *undersensitive adversarial example* is defined as an input $\tilde{x}_{under}$ such that:

1. $\tilde{x}_{under} \in \mathcal{A}_x$
2. $f(\tilde{x}_{under}) = f(x)$
3. $O(\tilde{x}_{under}) \neq O(x)$

We use similar antonym word substitution strategy proposed by Niu and Bansal (2018) to construct undersensitive adversarial examples. Similar to synonym word substitutions, for each word x_i in an input x, we construct an antonym set A_{x_i} that consists of the antonyms of x_i. Since we would like to change the semantic meaning of the input in a way that is likely to flip its label for the task, the attacker is only allowed to perturb one word with its antonym for each sentence.

The way we construct undersensitive adversarial examples may not always satisfy the assumption where the ground truth label of the undersensitive example would be different from the original input. The substituted word may not affect the semantic meaning of the input depending on the task. For example, in natural language inference, changing "the weather is *great*, we should go out and have fun" to "the weather is *bad*, ..." does not effect the entailment relationship with "we should have some outdoor activities" since the main argument is in the second part of the sentence. However, we find that antonym substitutions are able to change the semantic meaning of the text most of the time and we choose two tasks that are most likely to change the label under antonym-based attack.

3 Sensitivity Tradeoffs

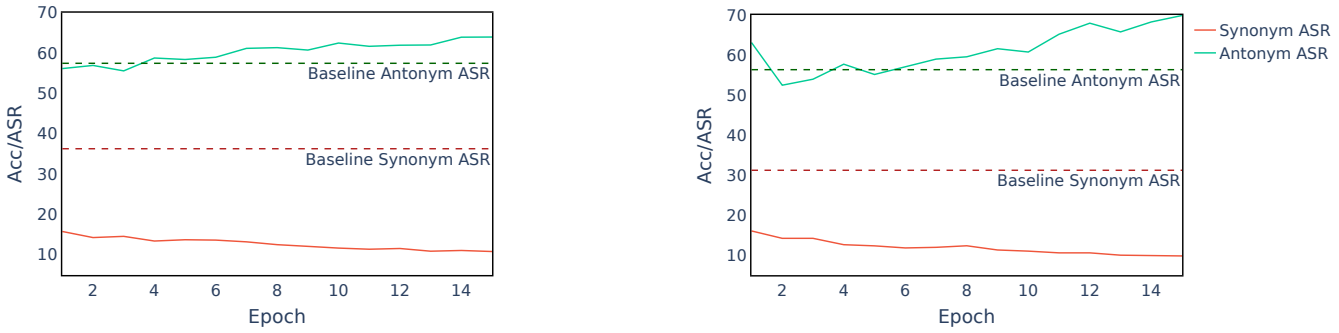
Methods for improving robustness aim to train models with decision boundaries that correctly classify inputs that would be oversensitive adversarial examples for non-robust models. Adversarial training is considered as the most effective defense strategy yet found against adversarial examples. It is usually done by augmenting the original training set with generated adversarial examples (Madry et al., 2018). It has also shown successful results in NLP domain (Yoo and Qi, 2021). Recent works have also studied certified robustness training which gives a stronger guarantee that the model is robust to all possible perturbations of a given input (Jia et al., 2019; Dong et al., 2021; Ye et al., 2020).

Normally, these defense methods only target oversensitive adversarial examples, so there is a risk that such methods increase vulnerability to undersensitive adversarial examples. According to the distance-oracle misalignment assumption (Tramer et al., 2020), the distance measure for finding adversarial examples and labeling oracle $\mathcal{O}$ is misaligned if we have $\mathcal{O}(\tilde{x}_{over}) = \mathcal{O}(x) = y$ and $\mathcal{O}(\tilde{x}_{under}) \neq \mathcal{O}(x)$, but $dist(x, \tilde{x}_{over}) > dist(x, \tilde{x}_{under})$ (Figure 1).

We explore this in the context of NLP models. Even though synonym word substitutions assume that the perturbed sentence should be semantically closer to the original sentence than any other sentence with a different semantic meaning in the embedding space, we may be able to find an undersensitive adversarial example that is closer to the original sentence.

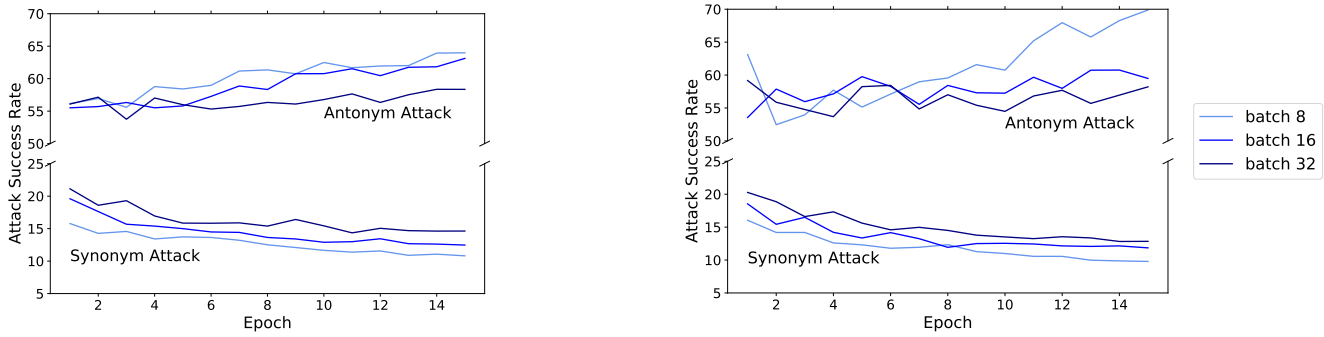
3.1 Setup

Our experiments are designed to test our hypothesis that optimizing adversarial robustness of NLP models using only oversensitive examples deteriorates the model's robustness on undersensitive adversarial examples. We use the SAFER certified robust training method proposed by Ye et al. (2020). The idea is to train a smoother model by randomly per-



(a) MNLI (BERT) (b) MNLI (RoBERTa)

Figure 3: Over-sensitivity and under-sensitivity tradeoff where under-sensitivity attack success rate increases as over-sensitivity attack success rate decreases. The figure shows the results on MNLI matched validation set. Dash lines show the synonym/antonym attack success rate on baseline model with normal training.



(a) MNLI (BERT) (b) MNLI (RoBERTa)

Figure 4: The synonym and antonym attack success rate at each SAFER training epoch with varying batch size. When the model is trained with smaller batch size, the synonym attack success rate is lower and the antonym success rate is higher.

turbing the sentences with words in the synonym substitution set at each training iteration.

We train BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models on two different tasks with SAFER training for 15 epochs. We then test the attack success rate for both oversensitivity and undersensitivity attacks at each training epoch. We use the same perturbation method as described in Section 2.1 for both the training and the attack. For each word, the synonym perturbation set is constructed by selecting the top K nearest neighbors with a cosine similarity constraint of 0.8 in GLOVE embeddings (Pennington et al., 2014), and the antonym perturbation set consists of antonym words found in WordNet (Miller, 1995). We follow the method of Jin et al. (2020) for finding oversensitive adversarial examples by using word importance ranking and Part-of-Speech (PoS) and sentence semantic similarity constraints as the search criteria. We use the same method for the undersensitivity attack, but exclude the semantic similarity constraint. For comparison, we set up baseline models with normal training on the original training sets.

3.2 Tasks

We choose two different tasks from the GLUE benchmark (Wang et al., 2018) that are good candidates for the antonym attack. Antonym-based attack works well on these tasks since both tasks consist of sentence pairs and changing a word to an opposite meaning is very likely to break the relationship between the pairs.

Natural Language Inference. We experiment with Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) which contains a premise-hypothesis pair for each example. The task is to identify the relation between the sentences in a premise-hypothesis pair and determine whether the hypothesis is true (entailment), false

(contradiction) or undetermined (neutral) given the premise. We consider the case where both premise and hypothesis can be perturbed, but only one word from either premise or hypothesis can be substituted for antonym attack. We do not consider examples with a neutral label when constructing undersensitive adversarial examples since antonym word substitutions may not change their label to a different class.

Paraphrase Identification. We use Quora Question Pairs (QQP) (Iyer et al., 2017) which consists of questions extracted from Quora. The goal of the task is to identify duplicate questions. Each question pair is labeled as duplicate or non-duplicate. For our antonym attack strategy, we only target the duplicate class since antonym word substitutions are unlikely to flip an initially non-duplicate pair into a duplicate.

We also conducted experiments using the Wiki Talk Comments (Wulczyn et al., 2017) dataset, a dataset for toxicity detection, by adding or removing toxic words for creating undersensitive examples. However, we found adding toxic words can reach almost 100% attack success rate, so there did not seem to be an interesting tradeoff to explore for available models for this task.

3.3 Results

We visualize the attack success rates for oversensitivity (synonym attack) and undersensitivity (antonym attack) attacks in Figure 3. The results confirm our hypothesis that optimizing adversarial robustness of NLP models using only oversensitive examples results in models that are more vulnerable to undersensitivity attacks. Robustness training for the BERT model on MNLI improves oversensitivity robustness, reducing the synonym attack success rate from 36% to 11% (a 69% decrease) after training for 15 epochs (Figure 3a), but antonym attack success rate increases from 56% to 64% (a 14% increase). The antonym attack success rate increases even more for the RoBERTa model (Figure 3b), increasing from 56% to 70% (a 25% increase) while the synonym attack success rate decreases from 31.2% to 13% (a 58% decrease). The RoBERTa model is pre-trained to be more robust than the BERT model, which perhaps explains the difference. We observe a sensitivity tradeoff for QQP dataset as well (see Appendix A.1).

Impact of Batch Size. We experiment with different batch sizes for over-sensitivity based robust training. We show the results on MNLI dataset in Figure 4. When the model is trained with a smaller batch size, the synonym attack success rate becomes lower, but the antonym success rate gets higher. This means that the model may overfit on the over-sensitive examples due to smaller training batch size, exacerbating the impact of the unbalanced adversarial training. We found similar evidence on the evaluation accuracy on the original validation set in Appendix A.2. While models with smaller batch sizes converge faster, they lead to lower performance and poorer generalization. This result suggests that SAFER with smaller batch size may create a larger robustness tradeoff. In the later section, we show that our proposed method would not be affected by the training batch size (Section 4.2).

4 Balanced Adversarial Training

In previous section, we argued that the oversensitivity/undersensitivity tradeoff can be attributed to distance-oracle misalignment. This section proposes and evaluates a modification to adversarial training that balances both kinds of adversarial examples.

4.1 Approach

To make the semantic distance in the representation space align better with human perception, the most intuitive way is to move the oversensitive example closer to the original input and push the original input apart from the undersensitive example in the representation space.

This goal matches the objective of contrastive learning, a type of self supervised learning that learns representations with positive (similar) examples close together and negative (dissimlar) examples far apart (Hadsell et al., 2006; Schroff et al., 2015). Positive examples are usually generated with data augmentation such as spatial transformation, and negative examples are sampled from other examples (Chen et al., 2020). We adapt contrastive learning to balance adversarial training by treating oversensitive adversarial examples as positive examples and undersensitive adversarial examples as negative examples.

We construct the positive pair as the original input with a corresponding oversensitive example, and the negative pair as the original input paired with an undersensitive example. We generate oversensitive and undersensitive examples by apply-

Model	Method	Eval Acc (%)	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	84.39/84.99	57.47/58.72	36.29/40.52
	A2T	84.44/85.00	56.51/57.86	21.67/24.84
	SAFER	84.20/84.66	58.36/58.45	14.62/16.61
	BAT-Pairwise	84.68/84.44	45.23/46.18	27.12/30.81
	BAT-Triplet	84.70/84.97	32.15/32.50	25.83/28.95
RoBERTa	Normal Training	87.85/87.42	56.34/58.85	31.20/34.60
	A2T	86.98/86.52	56.84/58.19	19.78/21.07
	SAFER	87.11/86.65	56.95/58.13	12.82/13.98
	BAT-Pairwise	87.57/87.52	39.71/40.12	27.56/30.79
	BAT-Triplet	87.61/86.99	32.74/33.57	26.90/28.91

Table 1: Balanced Adversarial Training evaluation results on MNLI matched/mismatched validation set.

ing synonym and antonym transformations respectively. The idea is to minimize the distance between the positive pairs and maximize the distance between the negative pairs.

We combine normal training with a contrastive learning objective and experiment with two different approaches for contrastive loss: pairwise and triplet loss. While recent contrastive learning incorporates multiple positive and negative examples for each input, we use these two methods as they consider the simplest case where only a positive and a negative example is needed for each input. Similarly to SAFER certified robust training, we use an augmented approach without querying the model to check if the attack succeeds. We choose this approach over traditional adversarial training since it is computationally less expensive.

Given an input (x, y), we generate an example $\tilde{x}_o$ by applying synonym perturbations and an example $\tilde{x}_u$ by applying antonym perturbations. Let $d(x_1, x_2)$ denote the distance measure between x_1 and x_2 in the representation space. For the pairwise approach, we optimize the distance for the over-sensitive pair $(x, \tilde{x}_o)$ and the under-sensitive pair $(x, \tilde{x}_u)$ independently:

$$\mathcal{L}^{BAT_{pair}} = \mathcal{L}_{ML} + \mathcal{L}_{pair}$$
$$\mathcal{L}_{ML} = \log f(y \mid x)$$
$$\mathcal{L}_{pair} = \alpha d(x, \tilde{x}_o) + \beta \max(0, m - d(x, \tilde{x}_u))$$

where the hyperparameters α and β control the weighting of the oversensitive and undersensitive pairs, and m is the margin. The $\mathcal{L}_{pair}$ loss term is designed to minimize the distance to the oversensitive adversarial example and maximize the distance to the undersensitive adversarial example. The margin m penalizes the model when the undersensitive example is less than m distance away from the original input ($d(x, \tilde{x}_u) < m$). We use cosine similarity for distance measure and set the margin m as 1. For the case where we are unable to find a valid oversensitive or undersensitive adversarial example, we set either $d(x, \tilde{x}_o)$ or $m - d(x, \tilde{x}_u)$ to 0.

For the triplet approach, the original input x acts as an anchor and a triplet $(x, \tilde{x}_o, \tilde{x}_u)$ is considered instead of pairs. The triplet loss aims to make the distance between the undersensitive pair larger than the distance between the oversensitive pair with at least a margin m: $d(x, \tilde{x}_u) > d(x, \tilde{x}_o) + m$. The training loss can be formalized as:

$$\mathcal{L}^{BAT_{triplet}} = \mathcal{L}_{ML} + \lambda \mathcal{L}_{triplet}$$
$$\mathcal{L}_{triplet} = \max(0, d(x, \tilde{x}_o) + (m - d(x, \tilde{x}_u)))$$

where the hyperparameter λ controls the weight of the contrastive loss term. Like the pairwise loss, if no oversensitive or undersensitive example is available, we mask out $d(x, \tilde{x}_o)$ or $m - d(x, \tilde{x}_u)$ in $\mathcal{L}_{triplet}$.

4.2 Results

Table 1 shows BAT training results on the MNLI validation sets. We use normal training as the non-robust baseline, and consider certified robust training, SAFER, and traditional adversarial training, A2T (Yoo and Qi, 2021), as the robust baselines. Balanced Adversarial Training increases the model's adversarial robustness against both antonym and synonym attacks, while preserving its performance on the original validation set. While both robust baselines that only consider oversensitive adversarial examples (SAFER and A2T) perform best when evaluated solely based on over-

Model	Method	Eval Acc (%)	F1	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	90.62	87.49	43.61	20.75
	SAFER	90.98	87.81	46.00	4.98
	BAT-Pairwise	89.99	86.67	21.24	15.81
	BAT-Triplet	90.85	87.81	14.26	15.78
RoBERTa	Normal Training	91.25	88.38	40.39	18.78
	SAFER	91.34	88.47	44.30	4.56
	BAT-Pairwise	89.99	86.62	18.29	17.07
	BAT-Triplet	91.04	88.21	13.02	16.89

Table 2: Balanced Adversarial Training evaluation results on QQP validation set.

sensitivity robustness, they are more vulnerable to undersensitive adversarial examples. We found that BAT-Triplet performs better than BAT-Pairwise in terms of improving robustness against antonym attacks. This may due to the fact that triplet loss forces the distance to undersensitive examples becomes larger than the distance to oversensitive examples.

With BAT-Triplet, the antonym attack success rate on BERT decreases from 57% to 32% (a 44% decrease) comparing to normal training, and the synonym atack success rate decreases from 36% to 26% (a 29% decrease). We also show the results on QQP dataset in Table 2. While the antonym attack success rates drop more than half (around 67% decrease) after BAT training, the synonym attack success rate has a 24% decrease on BERT and only 10% on RoBERTa, as the synonym attack success rate is already low on the model with normal training.

In Section 3.3, we observe that certified robust training with smaller batch size would result in larger gap in sensitivity tradeoff. We test BAT-Triplet with varying batch size when training BERT on MNLI task and we find that it gives consistent improvement on robustness regardless the batch size, as shown in Table 3.

4.3 Representation Analysis

We compare the learned representations of models trained with BAT to normal training and SAFER. We sample 500 examples from MNLI dataset (excluding the neutral class) and apply synonym and antonym perturbations for each input. We then project the model representations before the last classification layer to 2 dimensional space with t-SNE (van der Maaten and Hinton, 2008) and visualize the results in Figure 5.

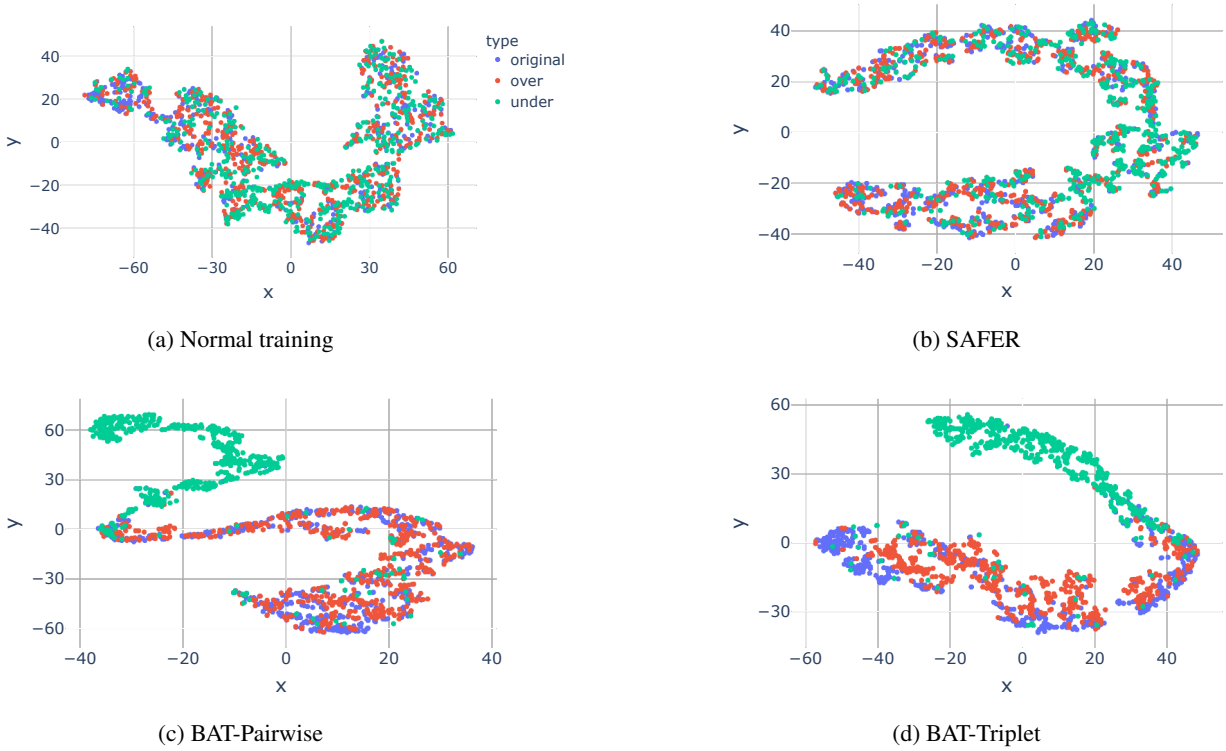
Batch Size	Accuracy (%)	Antonym ASR (%)	Synonym ASR (%)
8	84.45	34.59	25.66
16	84.08	31.05	25.89
32	84.70	32.15	25.83

Table 3: BAT-Triplet with BERT training, varying batch size, evaluated on MNLI matched validation set.

When training with normal training or SAFER, we can see that both oversensitive and undersensitive adversarial examples are fairly close to the original examples. However, with BAT-Pairwise or BAT-Triplet, undersensitive examples are pushed further away from both original and over-sensitive examples. This matches with BAT's training goal where the distance between undersensitive and original examples is maximized and the distance between oversensitive and original examples is minimized. This also shows how BAT is able to fix the distance-oracle misalignment, making the semantic distance in the representation space aligns better with human perception, and further improve robustness against both types of adversarial examples.

5 Related Work

Compared to oversensitive adversarial examples, undersensitivity has been less studied in NLP as well as other domains. Feng et al. (2018) delete words iteratively from the input to create examples that appear rubbish to human but retain the model's prediction with high confidence. Welbl et al. (2020) use Part-of-Speech and Name Entity based perturbations against reading comprehension models. Niu and Bansal (2018) study both types of attack strategies for dialogue models. They create undersensitive adversarial examples by substituting



(a) Normal training

(b) SAFER

(c) BAT-Pairwise

(d) BAT-Triplet

Figure 5: 2D projection of model representation for RoBERTa MNLI models trained with normal training, certified robust training with over-sensitive examples (SAFER), BAT-Pairwise, and BAT-Triplet.

words with antonyms or adding negation words to the input.

Our work is the first to study tradeoffs between oversensitive and undersensitive adversarial examples in NLP, but a few previous works have considered these tradeoffs in the vision domain. Jacobsen et al. (2019) show that adversary can not only target the model's excessive sensitivity but its excessive invariance to small changes in the input. They propose an alternative training objective based on information theory to make the model less invariant to semantically meaningful changes. Tramer et al. (2020) study the tradeoff between the two types of adversarial examples for image classifiers. They show that data augmentation can help increase robustness against undersensitivity attacks, but is not sufficient to impede both types of attacks. Our work differs in that we propose a new adversarial training method that improves model robustness against both types of adversarial examples. In addition, unlike images where human inspection is usually required to check whether the perturbed pixels would change the true label of the image, we are able to automate the process of generating undersensitive examples for text.

Recent work introduce contrastive learning for image classifiers in the adversarial learning setting where an oversensitive adversarial augmentation is used to generate positive examples and negative examples are sampled from other images. Kim et al. (2020) generate diverse positive examples by launching instance-wise attack on augmented images and show that it improves model's oversensitivity robustness. Ho and Nvasconcelos (2020) create challenging positive pairs by using the gradients of the contrastive loss to generate oversensitive adversarial examples and they show that it improves model performance.

6 Conclusion

We demonstrate the tradeoff between vulnerability to oversensitive and undersensitive adversarial examples for NLP models and show that increasing robustness against synonym based attack also increases vulnerability to antonym-based attacks. To manage this tension, we introduce a new adversarial training method, BAT, which targets the distance-oracle misalignment problem and can help balance the oversensitivity and undersensitivity in adversarial training.

References

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709.

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

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2.

Chih-Hui Ho and Nuno Nvasconcelos. 2020. Contrastive learning with adversarial examples. In *Advances in Neural Information Processing Systems*, volume 33.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. `https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs`.

Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. 2019. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05).

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. 2020. Adversarial self-supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33.

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.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11).

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2018. Adversarial over-sensitivity and over-stability strategies for dialogue models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 486–496, Brussels, Belgium. Association for Computational Linguistics.

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.

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.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Florian Tramer, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Joern-Henrik Jacobsen. 2020. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Johannes Welbl, Pasquale Minervini, Max Bartolo, Pontus Stenetorp, and Sebastian Riedel. 2020. Undersensitivity in neural reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1152–1165, Online. Association for Computational Linguistics.

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.

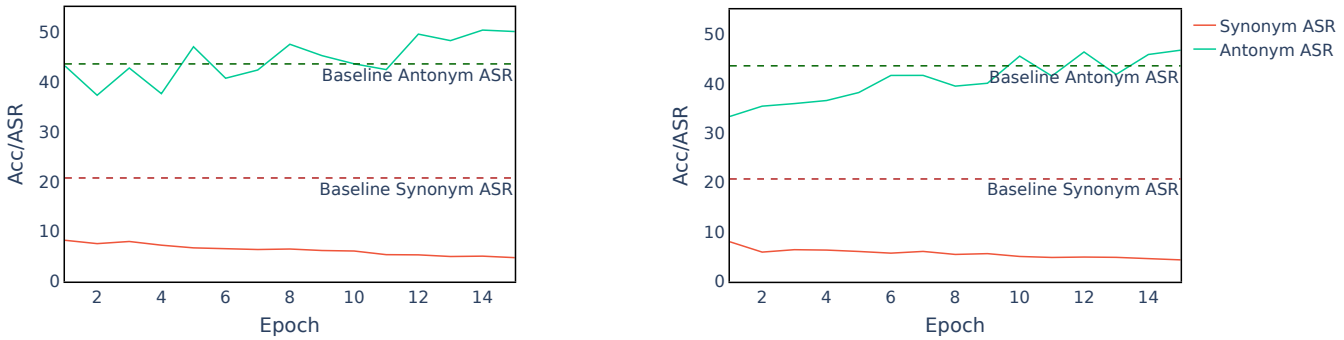
Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex Machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online. Association for Computational Linguistics.

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.

A Appendix

A.1 Over-sensitivity and Undersensitivity Tradeoff on QQP dataset

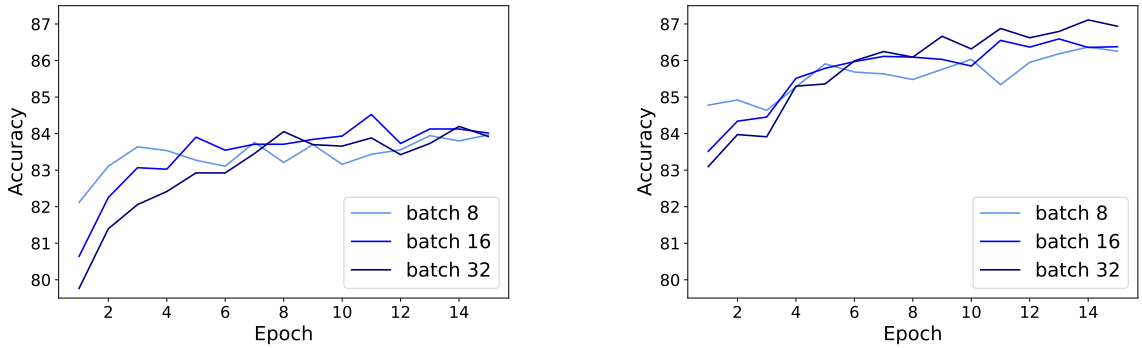


(a) QQP (BERT)

(b) QQP (RoBERTa)

Figure 6: Over-sensitivity and under-sensitivity tradeoff on QQP dataset.

A.2 Over-sensitivity Robust Training Evaluation Accuracy with Varying Batch Size



(a) MNLI (BERT)

(b) MNLI (RoBERTa)

Figure 7: The evaluation accuracy on original validation set at each SAFER training epoch with varying batch size.

A.3 Balanced Adversarial Training Details

We implement BAT similarly to the SAFER training method as described in Section 3.1 where we randomly perturb the inputs with words from the synonym/antonym substitution sets. We train BERT and RoBERTa models for 2 to 3 epochs with a learning rate of 2×10^{-5} or 3×10^{-5} and batch size of 32. For BAT-Triplet, we set the contrastive loss weight λ to 0.8 or 1.0. For BAT-Pairwise, we set the the weight of oversensitive pair and undersensitive pair (α, β) to (1.0, 1.0) or (1.0, 1.2).

A.4 Dataset Statistics

Dataset	Type	Train	Dev
MNLI	NLI	393K	20K
QQP	paraphrase	364K	391K

Table 4: Number of examples in each dataset split.