Improved Text Classification via Contrastive Adversarial Training

Lin Pan[1], **Chung-Wei Hang**[1], **Avi Sil**[2], **Saloni Potdar**[1]

IBM Watson[1]
IBM Research AI[2]
{panl, hangc, avi, potdars}@us.ibm.com

Abstract

We propose a simple and general method to regularize the fine-tuning of Transformer-based encoders for text classification tasks. Specifically, during fine-tuning we generate adversarial examples by perturbing the word embedding matrix of the model and perform contrastive learning on clean and adversarial examples in order to teach the model to learn noise-invariant representations. By training on both clean and adversarial examples along with the additional contrastive objective, we observe consistent improvement over standard fine-tuning on clean examples. On several GLUE benchmark tasks, our fine-tuned BERT$_{Large}$ model outperforms BERT$_{Large}$ baseline by 1.7% on average, and our fine-tuned RoBERTa$_{Large}$ improves over RoBERTa$_{Large}$ baseline by 1.3%. We additionally validate our method in different domains using three intent classification datasets, where our fine-tuned RoBERTa$_{Large}$ outperforms RoBERTa$_{Large}$ baseline by 1–2% on average. For the challenging low-resource scenario, we train our system using half of the training data (per intent) in each of the three intent classification datasets, and achieve similar performance compared to the baseline trained with full training data.

Introduction

Adversarial training (AT) introduced in Goodfellow, Shlens, and Szegedy (2015) provides an effective means of regularization and improving model robustness against adversarial examples (Szegedy et al. 2014) for computer vision (CV) tasks such as image classification. In AT of this form, a small, gradient-based perturbation is added to the original example, and a model is trained on both clean and perturbed examples. Due to the discrete nature of textual data, this method is not directly applicable to NLP tasks. Miyato, Dai, and Goodfellow (2017) extend this method to NLP and propose to apply perturbation to the word embeddings of an LSTM-based model (Hochreiter and Schmidhuber 1997) on text classification tasks. Since the word embeddings after perturbation do not map to new words in the vocabulary, the method is proposed exclusively as a means of regularization.

In this work, we present *CAT*, contrastive adversarial training for text classification. We build upon Miyato, Dai, and Goodfellow (2017) to regularize the fine-tuning of Transformer-based (Vaswani et al. 2017) encoders on text classification tasks. Instead of applying perturbation to word embeddings, we make a small change and apply it to the word embedding matrix of Transformer encoders and observe slightly better result in our experiments. Additionally, we encourage the model to learn noise-invariant representations by introducing a contrastive objective (van den Oord, Li, and Vinyals 2019) that pushes clean examples and their corresponding perturbed examples close to each other in the representation space, while pushing apart examples not from the same pair. We evaluate our method on a range of natural language understanding tasks including the standard GLUE (Wang et al. 2019) benchmark as well as three intent classification tasks for dialog systems.

On GLUE tasks, we compare our fine-tuning method against strong baselines of fine-tuning BERT$_{Large}$ (Devlin et al. 2019) and RoBERTa$_{Large}$ (Liu et al. 2019b) on clean examples with the cross-entropy loss. Our method outperforms BERT$_{Large}$ by 1.7% on average and RoBERTa$_{Large}$ by 1.3%. On intent classification tasks, our fine-tuned RoBERTa$_{Large}$ outperforms RoBERTa$_{Large}$ baseline by 1% on the full test sets and 2% on the difficult test sets. We further perform sample efficiency tests, where we use only half of the training data (per intent) and achieve near identical accuracy compared to the baseline trained using full training data.

Related Work

Adversarial Training

Adversarial Training (AT) has been explored in many supervised classification tasks that include object detection (Chen et al. 2018; Song et al. 2018; Xie et al. 2017), object segmentation (Arnab, Miksik, and Torr 2018; Xie et al. 2017) and image classification (Goodfellow, Shlens, and Szegedy 2015; Papernot et al. 2016; Su, Vargas, and Sakurai 2019).

AT can be defined as the process in which a system is trained to defend against malicious "attacks" and increase network robustness, by training the system with adversarial examples and optionally with clean examples. Typically these attacks are produced by perturbing the input (clean) examples, which makes the system predict the wrong class label (Chakraborty et al. 2018; Yuan et al. 2019). In this paper, we limit our discussion to white-box adversarial attacks, i.e., assuming access to model architecture and parameters.

Miyato, Dai, and Goodfellow (2017) extend the *Fast Gra-*

dient Sign Method (FGSM) proposed in Goodfellow, Shlens, and Szegedy (2015) to NLP tasks by perturbing word embeddings, and applies the method to both supervised and semi-supervised settings with Virtual Adversarial Training (VAT) (Miyato et al. 2016) for the latter. (Wu, Bamman, and Russell 2017) applies AT to relation extraction. Recent works (Kitada and Iyatomi 2020, 2021; Zhu et al. 2020) propose to apply perturbations to the attention mechanism in Transformer-based encoders. Compared to single-step FGSM, Madry et al. (2018) demonstrate the superior effectiveness of the multi-step approach to generate perturbed examples with projected gradient descent, which comes at a greater computational cost due to the inner loop that iteratively calculates the perturbations. Shafahi et al. (2019) propose "free" adversarial training. In the inner loop where perturbations are calculated, gradients with respect to model parameters are also calculated and updated. The number of training epochs are also reduced to achieve comparable complexity with natural training. Zhu et al. (2020) adopt the "free" AT algorithm and further add gradient accumulation to achieve a larger effective batch. Similar to Miyato, Dai, and Goodfellow (2017), perturbations are applied to word embeddings of LSTM and BERT-based models. In our work, we use the simpler one-step FGSM to generate perturbed examples and perform contrastive learning with clean examples.

Contrastive Learning

Recent advances in self-supervised contrastive learning, such as MoCo (He et al. 2020) and SimCLR (Chen et al. 2020) have bridged the gap in performance between self-supervised learning and fully-supervised methods on the ImageNet (Deng et al. 2009) dataset. Several works have successfully applied this representation learning paradigm to various NLP tasks.

A key component in contrastive learning is how to create positive pairs. (Fang et al. 2020) uses back-translation to generate another view of the original English data. Wu et al. (2020) apply word and span deletion, reordering, and substitution. Meng et al. (2021) use sequence cropping and masked sequence from an auxiliary Transformer. Giorgi et al. (2021) use nearby text spans in a document as positive pairs. Gunel et al. (2021) treat training examples of the same class as positive pairs and performs supervised contrastive learning (Khosla et al. 2020). Gao, Yao, and Chen (2021) use different dropout masks on the same batch data to generate positive pairs. As a supervised alternative, they leverage NLI datasets (Bowman et al. 2015; Williams, Nangia, and Bowman 2018) and treat premises and their corresponding hypotheses as positive pairs and contradictions as hard negatives.

In our work, we treat an original example, and its adversarial example as a positive pair, and a contrastive loss is used as an additional regularizer during fine-tuning. For multilingual NLP, Chi et al. (2020); Pan et al. (2021); Wei et al. (2020) leverage parallel data and perform contrastive learning on parallel sentences for cross-lingual representation learning.

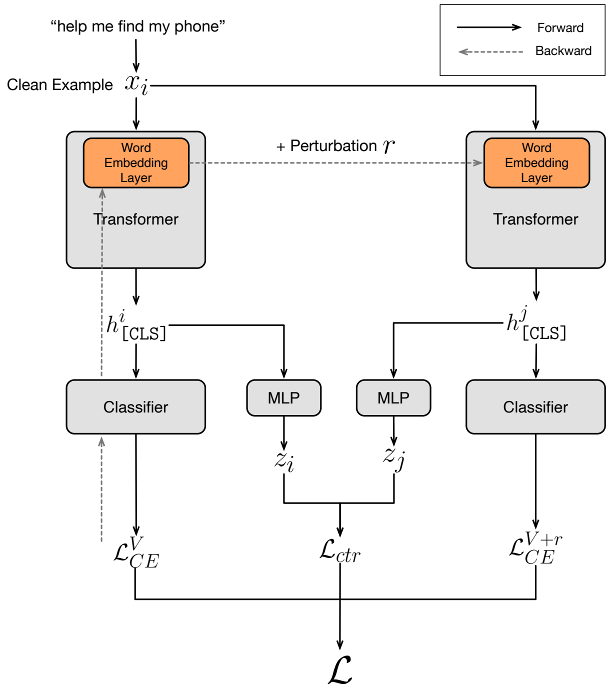

Figure 1: Model architecture for our proposed method to fine-tune Transformer-based encoders on text classification tasks. We use the Fast Gradient Sign Method to generate adversarial examples by perturbing the word embedding matrix V of the encoder. We then train on both clean and perturbed examples with the cross-entropy loss. Additionally, we introduce a third, contrastive loss that brings the representations of clean examples and their corresponding perturbed examples close to each other in order for the model to learn noise-invariant representations.

Improving BERT for Text Classification

As a general method to improve Transformer-based model performance on downstream tasks, Sun et al. (2020) and Gururangan et al. (2020) propose further language model pretraining in the target domain before the final fine-tuning. Du et al. (2021) propose to use self-training as another way to leverage unlabeled data, where a teacher model is first trained on labeled data, and is then used to label large amount of in-domain unlabeled data for the student model to learn from.

Recent developments in language model pretraining have also advanced state-of-the-art results on a wide range of NLP tasks. ELECTRA Clark et al. (2020) use a generator to generate noisy text via the mask language modeling objective and a discriminator is used to classify each input token as *original* or *replaced*. The model shows strong performance on downstream tasks by fine-tuning the discriminator. DeBERTa (He et al. 2021) proposes a disentangled attention mechanism and a new form of VAT, where perturbations are applied to the normalized word embeddings.

Method

In this section we first briefly describe the standard fine-tuning procedure for Transformer-based encoders on text classification tasks. We then introduce our method of generating adversarial examples and propose our method *CAT* that uses these examples to perform contrastive learning with clean examples. Figure 1 shows our overall model architecture.

Preliminaries

Our learning setup is based on a standard multi-class classification problem with input training examples $\{x_i, y_i\}_{i=1,\ldots,N}$. We assume access to a Transformer-based pre-trained language model (PLM), such as BERT, and RoBERTa. Given a token sequence $x_i = [CLS, t_1, t_2, \ldots, t_T, SEP]$ [1], the PLM outputs a sequence of contextualized token representations $H^L = [h^L_{[CLS]}, h^L_1, h^L_2, \ldots, h^L_T, h^L_{[SEP]}]$.

$$h^L_{[CLS]}, h^L_1, \ldots, h^L_T, h^L_{[SEP]} = PLM([CLS], t_1, \ldots, t_T, [SEP]),$$

where L denotes the number of model layers [2].

The standard practice for fine-tuning these large PLMs is to add a softmax classifier on top of the model's sentence-level representations, such as the final hidden state $h_{[CLS]}$ of the $[CLS]$ token in BERT:

$$p(y_c|h_{[CLS]}) = softmax(Wh_{[CLS]}) \quad c \in C, \tag{1}$$

where $W \in \mathbb{R}^{d_C \times d_h}$, and C denotes the number of classes. A model is trained by minimizing the cross entropy loss:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(p(y_{i,c}|h^i_{[CLS]})), \tag{2}$$

where N is batch size.

Adversarial Examples

Adversarial examples are imperceptibly perturbed input to a model that causes misclassification. Goodfellow, Shlens, and Szegedy (2015) propose the *Fast Gradient Sign Method* (FGSM) to generate such examples, and training on both clean and adversarial examples as an efficient way to improve model robustness against adversaries.

Formally, given a loss function $\mathcal{L}(f_\theta(x_i + r), y_i)$, where f_θ is a neural network parameterized by θ, and x_i the input example, we maximize the loss function subject to the max norm constraint on the perturbation r:

$$\max_r \mathcal{L}(f_\theta(x_i + r), y_i), s.t. \|r\|_\infty < \epsilon, \quad \text{where } \epsilon > 0 \tag{3}$$

Using first-order approximation, the loss function is approximately equivalent to the following:

$$\mathcal{L}(f_\theta(x_i + r), y_i) \approx \mathcal{L}(f_\theta(x_i), y_i) + \nabla_{x_i}\mathcal{L}(f_\theta(x_i), y_i)^T r \tag{4}$$

Solving for (3) and (4) yields perturbation in the following form:

$$r = -\epsilon \, sign(\nabla_{x_i}\mathcal{L}(f_\theta(x_i), y_i)) \tag{5}$$

Alternatively, using l_2-norm constraint on the perturbation r in (3) yields:

$$r = -\epsilon \frac{\nabla_{x_i}\mathcal{L}(f_\theta(x_i), y_i)}{\|\nabla_{x_i}\mathcal{L}(f_\theta(x_i), y_i)\|_2} \tag{6}$$

AT in (Goodfellow, Shlens, and Szegedy 2015) uses both the clean example x_i and perturbed example $x_i + r$ to train a model. For NLP problems where input is usually discrete, FGSM is not directly applicable. Miyato, Dai, and Goodfellow (2017) propose to apply perturbation to the word embedding v_i from the corresponding row in embedding matrix $V \in \mathbb{R}^{d_v \times d_h}$, where d_v is vocabulary size and d_h hidden size.

We follow this approach, but instead of perturbing the word embeddings, we directly perturb the *word embedding matrix* of Transformer-based encoders to generate our adversarial examples. Specifically, after each forward pass with clean examples, we calculate the gradient of the loss function in (2) with respect to the word embedding matrix V instead of word embedding in (5) to calculate the perturbation. Empirically, we find that perturbing the word embedding matrix performs better than word embeddings (see a comparison on GLUE in Table 9). For text classification tasks, we train on clean and adversarial examples with the cross entropy loss in (2). Additionally, we experiment with different forms of perturbation in (5) and (6), as well as randomly sampling between the two for each batch of data (10). We observe that using r with the max norm constraint consistently leads to the best result. In the Experiment section, we report results from using this form of perturbation.

Contrastive Learning

Intuitively, given a pair of clean and adversarial examples, we want their encoded sentence-level representation to be as similar to each other as possible so that our trained model will be more noise-invariant. At the same time, examples not from the same pair should be farther away in the representation space.

To model this relationship, we leverage contrastive learning as an additional regularizer during the fine-tuning process. Recent works on contrastive learning, such as MoCo (He et al. 2020), and SimCLR (Chen et al. 2020) use various forms of data augmentation, e.g., random cropping, and random color distortion, as the first step to create positive pairs. MoCo uses a queue structure to store negative examples, while SimCLR

[1] In the case of sequence pairs, another [SEP] token is added in between the sequences.

[2] We drop the layer superscript from here on for notation convenience.

Dataset	Task	Labels	Train	Metric	Train avg length	Dev avg length
MNLI	Textual entailment	3	393k	Accuracy	29	28
QQP	Question paraphrase	2	364k	Accuracy	21	21
QNLI	Question answering/Textual entailment	2	105k	Accuracy	35	37
MRPC	Paraphrase	2	3.k	F1	38	39
RTE	Textual entailment	2	2.5k	Accuracy	51	50
CoLA	Grammatical correctness	2	8.5k	MCC	8	8
SST-2	Sentiment analysis	2	67k	Accuracy	9	17

Table 1: GLUE sequence classification datasets statistics.

performs in-batch negative example sampling. A model is then trained by minimizing the InfoNCE loss. In our work, we employ the SimCLR formulation of positive and negative pairs, and its loss function to implement our contrastive objective.

Concretely, given the final hidden state $h^i_{[CLS]}$ of the $[CLS]$ token for a clean example, and $h^j_{[CLS]}$ for its corresponding adversarial example, we treat $(h^i_{[CLS]}, h^j_{[CLS]})$ as a pair of positive examples. Following Chen et al. (2020), we add a non-linear projection layer on top of them:

$$z_i = W_2 ReLU(W_1 h^i_{[CLS]}), \tag{7}$$

$$z_j = W_2 ReLU(W_1 h^j_{[CLS]}) \tag{8}$$

where $W_1 \in \mathbb{R}^{d_h \times d_h}$, $W_2 \in \mathbb{R}^{d_k \times d_h}$, and d_k is set to 300. With a batch of N clean examples and their corresponding adversarial examples, for each positive pair, there are $2(N-1)$ negative pairs, i.e., all the rest of the examples in the batch are negative examples. The contrastive objective is to identify the positive pair:

$$\mathcal{L}_{\text{ctr}} = -\log \frac{\exp(sim(z_i, z_j/\tau))}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(sim(z_i, z_k/\tau))}, \tag{9}$$

where $sim(u, v) = u^T v / \|u\|_2 \|v\|_2$ denotes cosine similarity between two vectors, and τ a temperature hyperparameter.

Finally, we perform fine-tuning in a multi-task manner and take a weighted average of the two classification losses and the contrastive loss [3]:

$$\mathcal{L} = \frac{(1-\lambda)}{2}(\mathcal{L}^{V}_{CE} + \mathcal{L}^{V+r}_{CE}) + \lambda \mathcal{L}_{ctr} \tag{10}$$

Experiment

Datasets

We conduct experiments on seven tasks of the GLUE benchmark, including textual entailment (MNLI, RTE), question answering/entailment (QNLI), question paraphrase (QQP), paraphrase (MRPC), grammatical correctness (CoLA), and sentiment analysis (SST-2). Table 1 summarizes the statistics of the GLUE tasks.

Dataset	Intents	Domains	Examples			Avg len
			Train	Test	Test (difficult)	
CLINC	150	10	17,999	4,500	750	8
BANKING	77	1	10,003	3,080	770	12
HWU	64	21	9,957	1,076	620	7

Table 2: Intent classification dataset statistics.

We additionally experiment on three commonly used intent classification datasets—CLINC (Larson et al. 2019), BANKING (Casanueva et al. 2020) and HWU (Liu et al. 2019a). Intent classification is the process of identifying the class (intent) of any utterance in a task-oriented dialog system. These three datasets largely represent a short-text classification task in real-world settings. Table 2 summarizes the statistics of the three datasets.

- **CLINC** covers 150 intents in 10 domains (e.g., banking, work, auto, travel). The dataset is designed to capture the breadth of topics that a production task-oriented chatbot handles. The dataset also comes with $1,200$ out-of-scope examples. In this work, we focus on in-scope examples only.
- **BANKING** is a single domain dataset created for fine-grained intent classification. The dataset consists of customer service queries in the banking domain, covering 77 intents across $10,003$ training examples and 3080 test examples.
- **HWU** covers 64 intents in 21 domains (e.g., alarm, email, game, news). The dataset is created in the context of a real-world use case of a home assistant bot. We use the one fold train-test split with $9,957$ training examples and $1,076$ test examples for our experiments.

On average, the sentence length of intent classification datasets is shorter than that of the GLUE tasks [4], since most of the GLUE tasks consist of two sentences. However, the number of classes is much greater (Table 1 and 2).

[3] In our experiments, we always assign equal weights to the two classification losses. It is possible that a different weight distribution yields better results.

[4] Sentence length is measured by the number of words instead of tokens since different BERT-like models use different tokenizers.

Model	MNLI	QQP	QNLI	MRPC	RTE	CoLA	SST-2	Avg
BERT$_{Large}$ (our impl)	86.6	91.4	92.0	90.0	69.3	61.8	93.6	83.5
BERT$_{Large}$ + AT + CTR (ours)	87.4†	92.2†	93.0†	91.6	71.5	65.8†	95.2†	85.2
RoBERTa$_{Large}$ (Liu et al. 2019b)	90.2	92.2	94.7	90.9	86.6	68.0	96.4	88.4
RoBERTa$_{Large}$ (our impl)	90.5	91.8	94.5	90.6	85.9	67.0	96.1	88.1
RoBERTa$_{Large}$ + AT + CTR (ours)	**91.1**†	92.5†	**95.1**†	**93.0**†	87.4	69.4	**97.0**	**89.4**
SCL (Gunel et al. 2021)	88.6	-	93.9	89.5	85.7	86.1*	96.3	-
PGD from (Zhu et al. 2020)	90.5	92.5	94.9	90.9	87.4	69.7	96.4	88.9
FreeAT from (Zhu et al. 2020)	90.0	92.5	94.7	90.7	86.7	68.8	96.1	88.5
FreeLB (Zhu et al. 2020)	90.6	**92.6**	95.0	91.4	**88.1**	**71.1**	96.8	89.4

Table 3: Results on the dev sets of GLUE benchmark. *AT* refers to the adversarial training component of our system and *CTR* the contrastive learning component. On average, our fine-tuned BERT$_{Large}$ model outperforms BERT$_{Large}$ baseline by 1.7%, and our fine-tuned RoBERTa$_{Large}$ improves over RoBERTa$_{Large}$ baseline by 1.3%. We fine-tune BERT$_{Large}$ and RoBERTa$_{Large}$ from their original checkpoints with no task-wise transfer learning involved. † indicates statistically significant improvement over the baseline. For CoLA and MRPC, we use Fisher Randomization test, and McNemar's test for all the other tasks. *It is unclear to us how the result for CoLA was derived in (Gunel et al. 2021) since their baseline RoBERTa$_{Large}$ is significantly higher than the ones reported in (Liu et al. 2019b) and other related works.

For the three intent classification datasets, in addition to the original evaluation data, we also evaluate on a *difficult* subset of each test set described in (Qi et al. 2021). The difficult subsets are constructed by comparing the TF-IDF vector of each test example to that of the training examples for a given intent. The test examples that are most dissimilar to the corresponding training examples are selected for inclusion to the difficult subset. We also experimented with generating our own difficult subsets in a similar manner using BERT-based sentence encoders [5], and compare each test example with the mean-pooling of the training examples for that intent. Result shows that the TF-IDF method yields a more challenging subset, thus we report results on the original subsets from Qi et al. (2021). The evaluation metric for all intent classification datasets is accuracy.

Training Details

We apply *CAT* to the fine-tuning of two backbone PLMs, BERT$_{Large}$ and RoBERTa$_{Large}$. For all experiments, we use AdamW optimizer with 0.01 weight decay and a linear learning rate scheduler. We set max sequence length to 128 and learning rate warmup for the first 10% of the total iterations.

On GLUE tasks, we largely follow the hyperparameter settings reported in Devlin et al. (2019) and Liu et al. (2019b) [6] to generate our BERT$_{Large}$ and RoBERTa$_{Large}$ baselines. For BERT$_{Large}$, we set batch size to 32 and fine-tune for 3 epochs. Grid search is performed over $lr \in \{0.00001, 0.00002, 0.00003\}$. For RoBERTa$_{Large}$, we sweep over the same learning rates as BERT$_{Large}$ and batch size $\in \{16, 32\}$.

Model	CLINC	BANKING	HWU	Avg
RoBERTa$_L$	97.4	93.9	92.4	94.6
RoBERTa$_L$+AT+CTR	**98.0**†	**95.0**†	**93.8**†	**95.6**

Table 4: Results on full test sets of intent classification datasets. *AT* refers to the adversarial training component of our system and *CTR* the contrastive learning component. On average, our fine-tuned RoBERTa$_{Large}$ improves over RoBERTa$_{Large}$ baseline by 1%. † indicates statistically significant improvement over the baseline using McNemar's test.

Model	CLINC	BANKING	HWU	Avg
RoBERTa$_L$	91.1	83.8	89.5	88.1
RoBERTa$_L$+AT+CTR	**92.1**†	**87.3**†	**90.8**†	**90.1**

Table 5: Results on *difficult* test sets of intent classification datasets. On average, our fine-tuned RoBERTa$_{Large}$ improves over RoBERTa$_{Large}$ baseline by 2%. The largest improvement is made on the BANKING dataset with 3.5%. † indicates statistically significant improvement over the baseline using McNemar's test.

On the three intent classification datasets, we follow the hyperparameter settings in Qi et al. (2021). We use a batch size of 32 and fine-tune for 5 epochs, and search over $lr \in \{0.00003, 0.00004, 0.00005\}$.

For fine-tuning with *CAT*, we use the exact same hyperparameter settings as the baseline, and further perform grid search over $\epsilon \in \{0.0001, 0.001, 0.005, 0.02\}$, $\tau \in \{0.05, 0.06, 0.07, 0.08, 0.09, 0.1\}$, and $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. All our experiments were run on a single 32 GB V100 GPU.

[5] The specific model used is "paraphrase-mpnet-base-v2" (Song et al. 2020), available from https://www.sbert.net/docs/pretrained_models.html.

[6] For RoBERTa$_{Large}$ baseline, we follow the hyperparameter settings specified at https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.glue.md

Model	Training data	CLINC	BANKING	HWU	Avg
RoBERTa$_{Large}$	**full**	**97.4**	93.9	**92.4**	**94.6**
RoBERTa$_{Large}$ + AT + CTR	**half**	97.1	**94.2**	92.3	94.5

Table 6: Sample efficiency test results on the full test sets of intent classification datasets. With our proposed fine-tuning method, we can use about half of the training data (per intent) and achieve near identical accuracy compared to baseline trained with full training data.

Model	MNLI	QQP	QNLI	MRPC	RTE	CoLA	SST-2	Avg
RoBERTa$_{Large}$	90.5	91.8	94.5	90.6	85.9	67.0	96.1	88.1
RoBERTa$_{Large}$ + AT	90.7	92.1	94.9	92.4	85.6	**69.9**	96.6	88.9
RoBERTa$_{Large}$ + AT + CTR	**91.1**	**92.5**	**95.1**	**93.0**	**87.4**	69.4	**97.0**	**89.4**

Table 7: Ablation results on the dev sets of GLUE benchmark. Adding our proposed AT that applies perturbation to the word embedding matrix leads to an improvement of 0.8% over the baseline, achieving the same performance as PGD in Table 3. Further adding the contrastive objective contributes to an additional 0.5% improvement.

GLUE Results

On GLUE tasks, we fine-tune BERT$_{Large}$ and RoBERTa$_{Large}$ using our method with two classification losses (on clean examples, and adversarial examples, respectively) and the contrastive loss. We compare them with the BERT$_{Large}$ and RoBERTa$_{Large}$ baseline, which are conventionally fine-tuned with classification loss on clean examples. For all experiments, we fine-tune BERT$_{Large}$ and RoBERTa$_{Large}$ from their original checkpoints with no task-wise transfer learning involved. We accompany each set of experiments with statistical significance test. For tasks evaluated with accuracy, we use McNemar's test. For CoLA, which is evaluated with Matthews correlation coefficient (MCC), and MRPC with F1, we use Fisher Randomization test.

Table 3 shows the dev set results. In summary, *CAT* fine-tuning approach consistently outperforms the standard fine-tuning approach for both BERT$_{Large}$ and RoBERTa$_{Large}$. *CAT* leads to a 1.7% improvement on average over conventionally fine-tuned BERT$_{Large}$. The largest improvement is observed on the CoLA task (i.e., 4.0%). For the stronger baseline of RoBERTa$_{Large}$, we observe an improvement of 1.3% on average with our method. On MRPC and CoLA, our result improves over RoBERTa$_{Large}$ baseline by 2.4%, showing the effectiveness our method on both single sequence, as well as sequence pair classification tasks. For statistical significance, we note that our improved results over the baseline are not significant on some smaller datasets, e.g., improved BERT$_{Large}$ on MRPC and RTE, and RoBERTa$_{Large}$ on MRPC, RTE and CoLA.

We also list results for three other AT methods: PGD (Madry et al. 2018), FreeAT (Shafahi et al. 2019) and FreeLB (Zhu et al. 2020), as well as Gunel et al. (2021) that use supervised contrastive learning for text classification. The three AT methods, which calculate perturbations iteratively, have been shown to produce stronger attacks compared to single-step methods (Athalye, Carlini, and Wagner 2018). Our single-step FGSM with perturbation applied to the word embedding matrix together with contrastive learning outperforms PGD and FreeAT while achieving the same overall performance as FreeLB. We note that baseline results from different papers are slightly different, which affects the results of proposed methods.

Model	CLINC	BANKING	HWU	Avg
RoBERTa$_L$	91.1	83.8	89.5	88.1
RoBERTa$_L$+AT	90.8	85.8	89.8	88.8
RoBERTa$_L$+AT+CTR	**92.1**	**87.3**	**90.8**	**90.1**

Table 8: Ablation results on *difficult* test sets of intent classification datasets. Adding AT improves over the baseline by 0.7%. Further adding the contrastive objective contributes to an additional 1.3% improvement.

Intent Classification Results

On the three intent classification datasets, CLINC, BANKING, and HWU, we experiment with the stronger RoBERTa$_{Large}$ baseline. Table 4 summarizes the results. On average, *CAT* outperforms the fine-tuned RoBERTa$_{Large}$ baseline by 1%, when evaluated on the full test sets of the three datasets. The largest improvement is made on the HWU dataset (1.4%).

Furthermore, we show that by training with adversarial examples and contrastive learning, *CAT* makes RoBERTa$_{Large}$ work better on the difficult test subsets of the three intent classification tasks. As shown in Table 5, our approach improves over standard RoBERTa$_{Large}$ fine-tuning by 2% on average. On BANKING, our method results in a large improvement of 3.5%. For statistical significance test on intent classification datasets, we use McNemar's test and observe significant improvement over baseline results for all datasets and evaluation settings.

Model	MNLI	QQP	QNLI	MRPC	RTE	CoLA	SST-2	Avg
BERT$_{Large}$ + AT^x + CTR	87.1	91.9	92.8	91.3	71.8	63.3	94.2	84.6
BERT$_{Large}$ + AT^V + CTR	87.4	92.2	93.0	91.6	71.5	65.8	95.2	85.2
RoBERTa$_{Large}$ AT^x + CTR	90.9	92.4	94.9	92.6	86.3	69.4	96.6	89.0
RoBERTa$_{Large}$ + AT^V + CTR	**91.1**	**92.5**	**95.1**	**93.0**	**87.4**	**69.4**	**97.0**	**89.4**

Table 9: Comparison between perturbation applied to word embeddings and word embedding matrix of BERT$_{Large}$ and RoBERTa$_{Large}$ during *CAT* fine-tuning. Results are on the dev sets of the GLUE benchmark. We use AT^x to denote adversarial training with perturbation applied to word embeddings and AT^V to the word embedding matrix. *CTR* refers to the contrastive learning component. On average, perturbing word embedding matrix outperforms perturbing word embeddings by 0.6% for BERT$_{Large}$ and 0.4% for RoBERTa$_{Large}$.

Model	MNLI	QNLI	MRPC	RTE	CoLA	SST-2	Avg
RoBERTa$_{Large}$ + AT + CTR (max norm/l_2-norm)	90.6	94.9	92.7	86.3	68.8	96.8	88.4
RoBERTa$_{Large}$ + AT + CTR (max norm)	**91.1**	**95.1**	**93.0**	**87.4**	**69.4**	**97.0**	**88.8**

Table 10: Comparison between consistently using the max norm constraint in generating adversarial examples, and randomly selecting between max norm and l_2-norm for each batch of data. Results are on the dev sets of GLUE benchmark. Consistently using the max norm constraint is 0.4% better than randomly selecting between max norm and l_2-norm.

Sample Efficiency

Next, we demonstrate that *CAT* has better sample efficiency compared to standard fine-tuning. We design the experiment on the three intent classification datasets. Specifically, we use about half of the training data (per intent) from each dataset to fine-tune RoBERTa$_{Large}$ with *CAT*. The training examples are randomly sampled from the full training data.

As Table 6 shows, with the usage of only half of the training data, our method achieves nearly the same results, compared to standard fine-tuning using all the training data. This result confirms the sample efficiency of our proposed *CAT*, and also indicates the advantage of our approach in the challenging low-resource scenarios.

Ablation

Finally, we perform ablation experiments on both GLUE benchmark and the three intent classification datasets. Results are presented in Table 7 and Table 8. On GLUE, by removing the contrastive loss, i.e., fine-tuning with clean and adversarial examples, we observe an accuracy drop of 0.5% on average. Compared to the baseline, this setup makes an average improvement of 0.8%. Our full system performs the best on all tasks except for CoLA, on which removing the contrastive loss yields the best performance. One interesting observation is that using our proposed perturbation to the word embedding matrix (instead of word embeddings), our *RoBERTa$_{Large}$ + AT* setting achieves the same average accuracy as PGD in Table 3 while using the simpler single-step FGSM.

On intent classification datasets, we use the difficult test sets for evaluation. Here, we observe a much larger effect from the additional contrastive loss, improving over *RoBERTa$_{Large}$ + AT* by 1.3% on average, while AT alone improves over the baseline by 0.7%. For all ablation experiments with the *RoBERTa$_{Large}$ + AT* setup, we perform a grid search over $\epsilon \in \{0.0001, 0.001, 0.005, 0.02\}$.

Conclusion

In this paper, we describe *CAT*, a simple and effective method for regularizing the fine-tuning of Transformer-based encoders. By leveraging adversarial training and contrastive learning, our system consistently outperforms the standard fine-tuning method for text classification. We use strong baseline models and evaluate our method on a range of GLUE benchmark tasks and three intent classification datasets in different settings. Sample efficiency and ablation tests show the positive effects of combining our adversarial and contrastive objectives for improved text classification. In the future, we plan to study additional word-level objectives to complement the sentence-level contrastive learning objective, in order to extend our method to other NLP tasks.

References

Arnab, A.; Miksik, O.; and Torr, P. H. 2018. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 888–897.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, 274–283. Stockholm.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 632–642. Lisbon: Association for Computational Linguistics.

Casanueva, I.; Temčinas, T.; Gerz, D.; Henderson, M.; and Vulić, I. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 38–45. Online.

Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; and Mukhopadhyay, D. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Chen, S.-T.; Cornelius, C.; Martin, J.; and Chau, D. H. 2018. ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector. CoRR abs/1804.05810 (2018). *arXiv preprint arXiv:1804.05810*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1597–1606. Virtual.

Chi, Z.; Dong, L.; Wei, F.; Yang, N.; Singhal, S.; Wang, W.; Song, X.; Mao, X.-L.; Huang, H.; and Zhou, M. 2020. InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training. *arXiv preprint arXiv:2007.07834*, 1–11.

Clark, K.; Luong, M.-T.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the 8th International Conference on Learning Representation (ICLR)*. Addis Ababa, Ethiopia.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *Processings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255. Miami.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 20th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186. Minneapolis: The Association for Computational Linguistics.

Du, J.; Grave, E.; Gunel, B.; Chaudhary, V.; Celebi, O.; Auli, M.; Stoyanov, V.; and Conneau, A. 2021. Self-training Improves Pre-training for Natural Language Understanding. *arXiv preprint arXiv:2010.02194*, 1–8.

Fang, H.; Wang, S.; Zhou, M.; Ding, J.; and Xie, P. 2020. CERT: Contrastive Self-supervised Learning for Language Understanding. *arXiv preprint arXiv:2005.12766*, 1–16.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821*, 1–16.

Giorgi, J.; Nitski, O.; Wang, B.; and Bader, G. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 879–895. Online.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *Proceedings of the 3rd International Conference on Learning Representation (ICLR)*. San Diego.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2021. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. In *Proceedings of the 9th International Conference on Learning Representation (ICLR)*. Virtual.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 8342–8360. Online.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735. Seattle.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of the 9th International Conference on Learning Representation (ICLR)*. Virtual.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, 18661–18673. Vancouver.

Kitada, S.; and Iyatomi, H. 2020. Attention Meets Perturbations: Robust and Interpretable Attention with Adversarial Training. *arXiv preprint arXiv:2009.12064*, 1–1.

Kitada, S.; and Iyatomi, H. 2021. Making Attention Mechanisms More Robust and Interpretable with Virtual Adversarial Training for Semi-Supervised Text Classification. *arXiv preprint arXiv:2104.08763*, 1–12.

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; Kummerfeld, J. K.; Leach, K.; Laurenzano, M. A.; Tang, L.; and Mars, J. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1311–1316. Hong Kong.

Liu, X.; Eshghi, A.; Swietojanski, P.; and Rieser, V. 2019a. Benchmarking Natural Language Understanding Services for building Conversational Agents. In *Proceedings of the International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, 165–183. Siracusa, Italy.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 1–13.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the 6th International Conference on Learning Representation (ICLR)*. Vancouver.

Meng, Y.; Xiong, C.; Bajaj, P.; Tiwary, S.; Bennett, P.; Han, J.; and Song, X. 2021. COCO-LM: Correcting and Contrasting Text Sequences for Language Model Pretraining. *arXiv preprint arXiv:2102.08473*, 1–13.

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. In *Proceedings of the 5th International Conference on Learning Representation (ICLR)*. Toulon, France.

Miyato, T.; ichi Maeda, S.; Koyama, M.; Nakae, K.; and Ishii, S. 2016. Distributional Smoothing with Virtual Adversarial Training. In *Proceedings of the 4th International Conference on Learning Representation (ICLR)*. San Juan, Puerto Rico.

Pan, L.; Hang, C.-W.; Qi, H.; Shah, A.; Potdar, S.; and Yu, M. 2021. Multilingual BERT Post-Pretraining Alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 210–219. Online.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, 372–387. IEEE.

Qi, H.; Pan, L.; Sood, A.; Shah, A.; Kunc, L.; Yu, M.; and Potdar, S. 2021. Benchmarking Commercial Intent Detection Services with Practice-Driven Evaluations. In *Proceedings of the 21st Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers (NAACL)*, 304–310. Online.

Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *arXiv preprint arXiv:1904.12843*.

Song, D.; Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Tramer, F.; Prakash, A.; and Kohno, T. 2018. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT)*.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver.

Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5): 828–841.

Sun, C.; Qiu, X.; Xu, Y.; and Huang, X. 2020. How to Fine-Tune BERT for Text Classification? *arXiv preprint arXiv:1905.05583*, 1–10.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representation (ICLR)*. Banff, AB, Canada.

van den Oord, A.; Li, Y.; and Vinyals, O. 2019. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 1–13.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NIPS)*, 5998–6008. Long Beach, CA.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 7th International Conference on Learning Representation (ICLR)*. New Orleans.

Wei, X.; Hu, Y.; Weng, R.; Xing, L.; Yu, H.; and Luo, W. 2020. On Learning Universal Representations Across Languages. *arXiv preprint arXiv:2007.15960*, 1–13.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 19th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, 1112–1122. New Orleans: The Association for Computational Linguistics.

Wu, Y.; Bamman, D.; and Russell, S. 2017. Adversarial Training for Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1778–1783. Copenhagen.

Wu, Z.; Wang, S.; Gu, J.; Khabsa, M.; Sun, F.; and Ma, H. 2020. CLEAR: Contrastive Learning for Sentence Representation. *arXiv preprint arXiv:2012.15466*, 1–10.

Xie, C.; Wang, J.; Zhang, Z.; Zhou, Y.; Xie, L.; and Yuille, A. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 1369–1378.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9): 2805–2824.

Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; and Liu, J. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *Proceedings of the 8th International Conference on Learning Representation (ICLR)*. Addis Ababa, Ethiopia.