Instance Regularization for Discriminative Language Model Pre-training

Zhuosheng Zhang[1,2*], Hai Zhao[1,2], Ming Zhou[3]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University
[3]Langboat Technology
zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, zhouming@chuangxin.com

Abstract

Discriminative pre-trained language models (PrLMs) can be generalized as denoising auto-encoders that work with two procedures, ennoising and denoising. First, an ennoising process corrupts texts with arbitrary noising functions to construct training instances. Then, a denoising language model is trained to restore the corrupted tokens. Existing studies have made progress by optimizing independent strategies of either ennoising or denosing. They treat training instances equally throughout the training process, with little attention on the individual contribution of those instances. To model explicit signals of instance contribution, this work proposes to estimate the complexity of restoring the original sentences from corrupted ones in language model pre-training. The estimations involve the corruption degree in the ennoising data construction process and the prediction confidence in the denoising counterpart. Experimental results on natural language understanding and reading comprehension benchmarks show that our approach improves pre-training efficiency, effectiveness, and robustness. Code is publicly available at https://github.com/cooelf/InstanceReg

1 Introduction

Leveraging self-supervised objectives to pre-train language models (PrLMs) on massive unlabeled data has shown success in natural language processing (NLP) (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Dong et al., 2019; Lan et al., 2020; Clark et al., 2020; Luo et al., 2021; Zhu et al., 2022). A wide landscape of pre-training objectives has been produced, such as autoregressive (Radford et al., 2018; Yang et al., 2019) and autoencoding (Devlin et al., 2019; Joshi et al., 2020) language modeling objectives, which serve as the principled mechanisms to teach language models general-purpose knowledge through the pre-training, and then those pre-trained PrLMs can be fine-tuned for downstream tasks. Based on these unsupervised functions, three classes of PrLMs have been proposed: autoregressive language models (e.g. GPT (Radford et al., 2018)), autoencoding models (e.g. BERT (Devlin et al., 2019)), and encoder-decoder models (e.g. BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020)). In this work, we focus on the research line of autoencoding models, also known as discriminative PrLMs that have achieved impressive performance on natural language understanding (NLU).

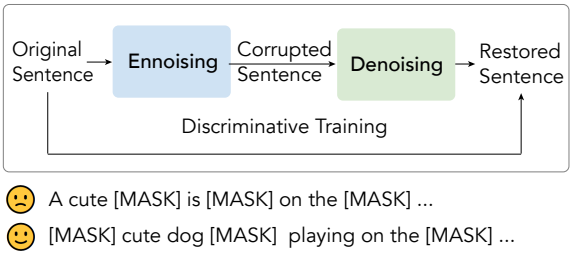

Figure 1: Overview of AutoDecoders. As the two examples show, the random sampling operation during ennoising would result in training instances of different degree of difficulty, e.g., the variety of valid alternatives.

Although the discriminative PrLMs may vary in language modeling functions or architectures as discussed above, they can be generalized as denoising auto-encoders, which contain two procedures, ennoising and denoising. The pre-training procedure is illustrated in Figure 1.

1) Ennoising corrupts texts with arbitrary noising functions to construct training instances. The corruption scheme includes edit operations like insertion, deletion, replacement, permutation, and retrieval (Devlin et al., 2019; Lewis et al., 2020b; Xu and Zhao, 2021; Wang et al., 2020; Guu et al.,

* Work done during internship at Lanboat. This work was partially supported by Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11255–11265
December 7-11, 2022

2020). For example, masked language modeling (MLM) (Devlin et al., 2019) replaces some input tokens in a sentence with a special symbol. BART uses token deletion, text infilling, and sentence permutation for corruption (Lewis et al., 2020a).

2) Denoising enables a language model to predict missing or otherwise corrupted tokens in the input sequences. Recent studies focus on designing improved language modeling functions to mitigate discrepancies between the pre-training phase and the fine-tuning phase. Yang et al. (2019) reformulates MLM in XLNet by restoring the permuted tokens in factorization order, such that the input sequence is autoregressively generated after permutation. In addition, using synonyms for the masking purpose (Cui et al., 2020) and simple pre-training objectives based on token-level classification tasks (Yamaguchi et al., 2021) have also proved effective as an MLM alternative.

Most of the existing studies of PrLMs fall into the scope of either investigating better ennoising operations or more effective denoising strategies. They treat training instances equally throughout the training process. Little attention is paid to the individual contribution of those instances. In standard MLM ennoising, randomly masking different tokens would lead to different degrees of corruption that may, therefore, cause different levels of difficulty in sentence restoration in denoising (as shown in Figure 1) and thus increase the uncertainty in restoring the original sentence structure during the denoising process. For example, if "not" is masked, the corrupted sentence tends to have a contrary meaning.

In this work, we are motivated to estimate the complexity of restoring the original sentences from corrupted ones in language model pre-training, to provide explicit regularization signals to encourage more effective and robust pre-training. Our approach includes two sides of penalty: 1) ennoising corruption penalty that measures the distribution disparity between the corrupted sentence and the original sentence, to measure the corruption degree in the ennoising process; 2) denoising prediction penalty that measures the distribution difference between the restored sequence and the original sentence to measure the sentence-level prediction confidence in the denoising counterpart. Experiments show that language models trained with our regularization terms can yield better performance and become more robust against adversarial attacks.

2 Related Work

Training powerful large-scale language models on a large unlabeled corpus with self-supervised objectives has attracted lots of attention, which commonly work in two procedures of ennoising and denoising. The most representative task for pre-training is MLM, which is introduced in Devlin et al. (2019) to pre-train a bidirectional BERT. A spectrum of ennoising extensions has been proposed to enhance MLM further and alleviate the potential drawbacks, which fall into two categories: 1) mask units and 2) noising scheme. Mask units correspond to the language modeling units that serve as knowledge carriers in different granularity. The variants focusing on mask units include the standard subword masking (Devlin et al., 2019), span masking (Joshi et al., 2020), and n-gram masking (Levine et al., 2021; Li and Zhao, 2021). For noising scheme, BART (Lewis et al., 2020a) corrupts text with arbitrary noising functions, including token deletion, text infilling, sentence permutation, in conjunction with MLM. UniLM (Dong et al., 2019) extends the mask prediction to generation tasks by adding the auto-regressive objectives. XLNet (Yang et al., 2019) proposes the permuted language modeling to learn the dependencies among the masked tokens. MacBERT (Cui et al., 2020) suggests using similar words for the masking purpose. Yamaguchi et al. (2021) also investigates simple pre-training objectives based on token-level classification tasks as replacements of MLM, which are often computationally cheaper and result in comparable performance to MLM. In addition, ELECTRA (Clark et al., 2020) proposes a novel training objective called replaced token detection, which is defined over all input tokens.

Although the above studies have an adequate investigation to reduce the mismatch between pre-training and fine-tuning tasks, an essential problem of the common denoising mechanism lacks attention. The construction of training examples based on ennoising operations would cause the break of sentence structure, either for replacement, addition, or deletion-based noising functions. In extreme cases, the destruction would lead to completely different sentences, making it difficult for the model to predict the corrupted tokens. Therefore, in this work, we propose to

enhance the pre-training quality by using instance regularization (IR) terms to estimate the restoration complexity from both sides of ennoising and denoising aspects.

The proposed approach is partially related to some prior studies of hardness measurement in training deep learning models (Lin et al., 2017; Kalantidis et al., 2020; Hao et al., 2021), whose focus is to guide the model to pay special attention to hard examples and prevent the vast number of easy negatives from overwhelming the training process. In contrast to optimizing the training process by heuristically finding the hard negatives, this work does not need to distinguish hard examples from ordinary ones, but measures the corruption degree between the masked sentence and the original sentence instead, and uses the degree as the explicit training signals.

3 Methodology

This section will start by formulating the ennoising and denoising processes for building PrLMs and then introduce our instance regularization approach to estimate the restoration complexity in both ennoising and denoising views.

3.1 Preliminary: Denoising Auto-Encoders

The training procedure for discriminative language models includes ennoising and denoising processes, as described below. For the sake of simplicity, we take the widely-used MLM as a typical example to describe the ennoising process.

Ennoising Given a sentence $W = \{w_1, w_2, \ldots, w_n\}$ with n tokens,[1] we randomly mask some percentage of the input tokens with a special mask symbol `[MASK]` and then predict those masked tokens. Suppose that there are m tokens replaced by the mask symbol. Let $\mathcal{D} = \{k_1, k_2, \ldots, k_m\}$ denote set of masked positions, we have W' as the masked sentence and $M = \{w_{k_1}, w_{k_2}, \ldots, w_{k_m}\}$ are the masked tokens. In the following part, we use w_k to denote each masked token for simplicity.

Denoising In the denoising process, a language model is trained to predict the masked token based on the context. W' is fed into the PrLM to obtain the contextual representations from the last Transformer layer, which is denoted as H.

[1]We assume that W has already been tokenized into a sequence of subwords.

Training The training objective is to maximize the following objective:

$$\mathcal{L}_{DAE} = -\frac{1}{m}\sum_{k\in\mathcal{D}}^{|D|} \log p(w_k \mid W'). \quad (1)$$

3.2 Instance Regularization

In this part, we will introduce our instance regularization approach, which involves two sides: corruption degree in the ennoising data construction process and the sequence-level prediction confidence in the denoising counterpart.

During denoising, the PrLM trained by MLM is required to predict the original masked tokens w_k given the hidden states H of the corrupted input W'. Let w'_k denote the predicted tokens, we replace the mask symbols by filling w'_k back to W'. As a result, we have the predicted sequence, denoted as $P = \{p_1, p_2, \ldots, p_n\}$, where the tokens in positions of $\mathcal{D}$ are predicted ones; otherwise, they are the same as the originals ones in W.

Obviously, the corruption would break the sentence structure and easily cause the semantic deviation of sentence representations. According to our observation, the hidden states would vary dramatically before and after the token corruption – similar findings were also observed in Wang et al. (2021) that small disturbance can inveigle PrLMs into making false predictions. In a more general perspective, replacing a modest percentage of tokens may result in a totally different sentence, let alone imperceptible disturbance as used for textual attacks.

Therefore, we propose two approaches called ennoising corruption penalty (ECP) and denoising prediction penalty (DPP) as the regularization terms in the training process to alleviate the issue. Figure 2 overviews the overall procedure. ECP measures the semantic shift from the original sentence to the corrupted one as an explicit signal to help the model distinguish easy and hard examples and learn with different weights, which can be seen as instance weighting compared with MLM. As the complement, DPP measures the sequence-level semantic distance between the predicted and original sentence to supplement the rough token-level matching of MLM, thus transforming the token prediction task to sequence matching to pay more attention to sentence-level semantics.

Both methods are used for estimating the difficulty of restoring the whole sequence from the

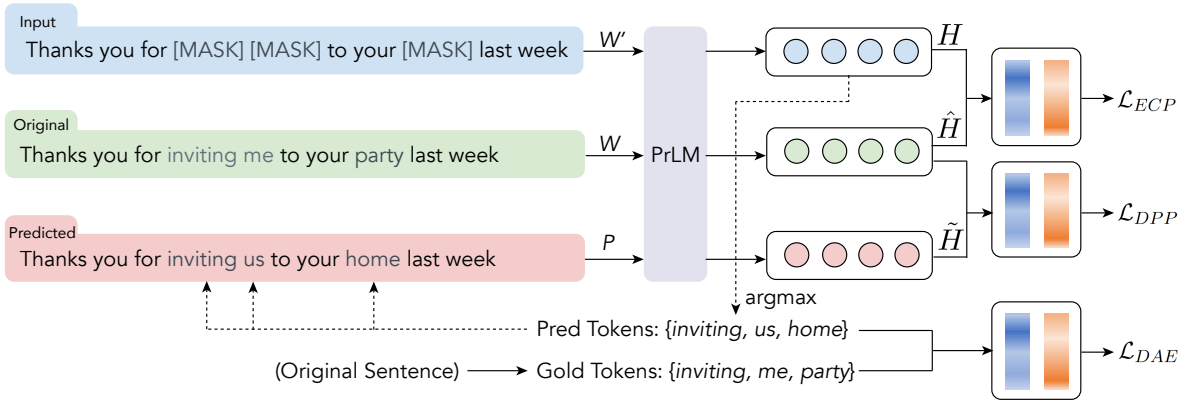

Figure 2: Overview of the procedure for the instance regularization approach, which estimates the corruption degree in the ennoising data construction process and the prediction confidence in the denoising counterpart.

corrupted ones, either in the role of the front-end ennoising or back-end denoising. Larger values of the estimation indicate larger semantic shifts.

Here we go back to the formulation in MLM. As shown in Figure 2, given the original sequence W, the masked sequence W', and the predicted sequence P, we obtain the contextualized representation from PrLM. Note that we already have the contextualized representation H for the input sequence W' in the vanilla MLM training. Similarly, we feed W and P to the PrLM, and the corresponding hidden states are written as $\hat{H}$ and $\tilde{H}$, respectively. Then, H, $\hat{H}$ and $\tilde{H}$ are leveraged as the elements for the corruption agreement and semantic agreement.

Ennoising Corruption Penalty After we get the distributions H and $\hat{H}$, we measure the extent of the corruption degree after ennoising by calculating the distribution difference between the masked and the original representations after normalization:

$$\mathcal{L}_{ECP} = D_{KL}(H, \hat{H}), \tag{2}$$

where KL refers to Kullback–Leibler (KL) divergence. Concretely, we apply softmax on the two matrices along the hidden dimensions to have two distributions. Then, we calculate KL divergence between the two distributions for each position in each sentence. Intuitively, higher $\mathcal{L}_{ECP}$ means the corruption is more severe, so is the gap between the ennoised instance and denoised prediction. Therefore, the model is supposed to update the gradient more significantly for those "harder" training instances.

Denoising Prediction Penalty In the denoising language modeling, the model would yield reasonable predictions but be discriminated as wrong predictions because such predictions do not match the single gold token for each training case using token-level cross-entropy. Therefore, we estimate the semantic agreement between the predicted sequence and the original gold sequence, by guiding the probability distribution of model predictions $\tilde{H}$ to match the expected probability distribution $\hat{H}$, we have:

$$\mathcal{L}_{DPP} = D_{KL}(\tilde{H}, \hat{H}), \tag{3}$$

where $\mathcal{L}_{DPP}$ is applied as the degree to reflect the sentence level semantic mismatch.

The semantic agreement method works as a means of soft regularization to capture the sequence-level similarity as a supplement to the standard hard token-level matching in cross-entropy.

In language model pre-training, we minimize $\mathcal{L}_H$ and $\mathcal{L}_S$. Thus, the loss function is written as

$$\mathcal{L} = \mathcal{L}_{DAE} + \mathcal{L}_{ECP} + \mathcal{L}_{DPP}. \tag{4}$$

4 Experiments

4.1 Setup

To verify the effectiveness of the proposed methods, we conduct pre-training experiments and fine-tune the pre-trained models on downstream tasks. All codes are implemented using PyTorch (Paszke et al., 2017).[2] The experiments are run on 8 NVIDIA GeForce RTX 3090 GPUs.

Pre-training We employ BERT and ELECTRA as the backbone PrLMs and implement our

[2]Our codes and models will be publicly available.

methods during the pre-training. For pre-training corpus, we use English Wikipedia corpus and BookCorpus (Zhu et al., 2015) following BERT (Devlin et al., 2019). As suggested in Liu et al. (2019), we do not use the next sentence prediction (NSP) objective as used in Devlin et al. (2019), but only use MLM as the baseline language modeling objective, with a masked ratio of 15%. After masking, 80% of the masked positions are replaced with `[MASK]`, 10% are replaced by randomly sampled words, and the remaining 10% are kept unchanged. We set the maximum length of the input sequence to 512, and the learning rates are 3e-5. We pre-train the base and large models for $100k$ steps using the pre-trained weights of the public BERT and ELECTRA models as initialization. The baselines are trained to the same steps for a fair comparison. To keep the simplicity like BERT training, following Li et al. (2020), we discard the generator in ELECTRA models and use the discriminator in the same way as BERT, with a classification layer to predict the corrupted tokens.

Fine-tuning We use an initial learning rate in {8e-6, 1e-5, 2e-5, 3e-5} with warm-up rate of 0.1 and L2 weight decay of 0.01. The batch size is selected in {16, 24, 32}. The maximum number of epochs is set in [2, 5] depending on tasks. Texts are tokenized with a maximum length of 384 for SQuAD and 512 for other tasks. Hyper-parameters were selected using the development set.

4.2 Tasks and Datasets

For evaluation, we fine-tune the pre-trained models on GLUE (General Language Understanding Evaluation) (Wang et al., 2019) and the popular Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) to evaluate the performance of the pre-trained models. The concerned tasks involve natural language inference, semantic similarity, text classification, and machine reading comprehension (MRC).

Natural Language Inference Natural Language Inference involves reading a pair of sentences and judging the relationship between their meanings, such as entailment, neutral and contradiction. We evaluate on three diverse datasets, including Multi-Genre Natural Language Inference (MNLI) (Nangia et al., 2017), Question Natural Language Inference (QNLI) (Rajpurkar et al., 2016) and Recognizing Textual Entailment (RTE) (Bentivogli et al., 2009).

Semantic Similarity Semantic similarity tasks aim to predict whether two sentences are semantically equivalent or not. The challenge lies in recognizing rephrasing of concepts, understanding negation, and handling syntactic ambiguity. Three datasets are used, including Microsoft Paraphrase corpus (MRPC) (Dolan and Brockett, 2005), Quora Question Pairs (QQP) dataset (Chen et al., 2018) and Semantic Textual Similarity benchmark (STS-B) (Cer et al., 2017).

Classification The Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019) is used to predict whether an English sentence is linguistically acceptable or not. The Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) provides a dataset for sentiment classification that needs to determine whether the sentiment of a sentence extracted from movie reviews is positive or negative.

Reading Comprehension As a widely used MRC benchmark dataset, SQuAD (Rajpurkar et al., 2016) is a reading comprehension dataset that requires the machine to extract the answer span given a document along with a question. We select the v1.1 version to keep the focus on the performance of pure span extraction performance. Two official metrics are used to evaluate the model performance: Exact Match (EM) and a softer metric F1 score, which measures the average overlap between the prediction and ground truth answer at the token level.

4.3 Main Results

Table 1 presents the results of our methods and baselines under the same pre-training settings on the GLUE development sets. We see that our method achieves consistent performance gains over both BERT and ELECTRA baselines under the base and large settings, i.e., with the increased average scores of +1.20%/0.57% on BERT (base/large) and +1.01%/0.83% on ELECTRA (base/large).[3] The results indicate that the instance regularization approach is effective for improving the general language understanding capacity of PrLMs.

We also show the comparisons with public

[3]We report the results on large models just to verify the consistent advance instead of pursuing absolute scores, due to the difficulty of training larger models on a single machine with 8 NVIDIA GTX 3090 GPUs (e.g., weak convergence with small batch sizes).

Model	CoLA *Mcc*	SST-2 *Acc*	MRPC *Acc*	STS-B *Spear*	QQP *Acc*	MNLI *Acc*	QNLI *Acc*	RTE *Acc*	Average -
Results on the development sets									
$\text{BERT}_{\text{base}}$	59.32	92.32	87.25	87.36	90.78	84.75	91.42	65.34	82.32
$\text{BERT-IR}_{\text{base}}$	61.39	93.46	87.50	89.05	90.90	85.28	91.84	68.95	83.52
$\text{BERT}_{\text{large}}$	62.45	93.58	88.24	90.48	91.45	87.20	92.37	74.01	84.97
$\text{BERT-IR}_{\text{large}}$	64.07	94.27	88.73	90.57	91.55	87.35	92.71	75.09	85.54
$\text{ELECTRA}_{\text{base}}$	65.53	94.95	88.97	89.96	91.24	88.45	92.53	77.62	86.16
$\text{ELECTRA-IR}_{\text{base}}$	68.95	95.30	90.44	90.52	91.40	88.66	93.04	79.06	87.17
$\text{ELECTRA}_{\text{large}}$	70.41	96.79	89.22	91.92	92.07	90.26	94.40	85.92	88.87
$\text{ELECTRA-IR}_{\text{large}}$	72.09	97.48	91.18	92.03	92.27	90.55	94.64	87.36	89.70
Results on the test sets									
$\text{BERT}_{\text{base}}$	52.1	93.5	84.8	85.8	89.2	84.6	90.5	66.4	80.9
$\text{BERT-IR}_{\text{base}}$	54.1	93.9	84.9	86.6	89.1	85.3	91.1	71.2	82.0
$\text{BERT}_{\text{large}}$	60.5	94.9	85.4	86.5	89.3	86.7	92.7	70.1	83.3
$\text{BERT-IR}_{\text{large}}$	61.7	94.2	85.7	87.1	89.4	86.5	92.9	72.1	83.7
$\text{ELECTRA}_{\text{base}}$	59.7	93.4	86.7	87.7	89.1	85.8	92.7	73.1	83.5
$\text{ELECTRA-IR}_{\text{base}}$	63.2	95.4	86.5	89.0	89.2	88.4	92.9	70.7	84.4
$\text{ELECTRA}_{\text{large}}$	68.1	96.7	89.2	91.7	90.4	90.7	95.5	86.1	88.6
$\text{ELECTRA-IR}_{\text{large}}$	70.1	97.0	89.8	91.6	90.2	90.9	95.8	86.8	89.0

Table 1: Comparisons between our proposed methods and the baseline models under on the GLUE development sets. STS-B is reported by Spearman correlation, CoLA is reported by Matthew's correlation, and the other tasks are reported by accuracy. Only one decimal place is reserved for the test results which are from the online GLUE server.

Model	EM Score	F1 Score
$\text{BERT}_{\text{base}}$	80.48	87.77
$\text{BERT-IR}_{\text{base}}$	81.28	88.38
$\text{BERT}_{\text{large}}$	83.54	90.26
$\text{BERT-IR}_{\text{large}}$	84.26	90.92
$\text{ELECTRA}_{\text{base}}$	83.82	90.59
$\text{ELECTRA-IR}_{\text{base}}$	84.49	91.18
$\text{ELECTRA}_{\text{large}}$	87.59	93.78
$\text{ELECTRA-IR}_{\text{large}}$	88.34	94.09

Table 2: Results on the SQuAD development set. The evaluation metrics are Exact-Match (EM) and F1 scores.

methods on the GLUE test sets. For a fair comparison, we only compare with the related single models fine-tuned for a single task, without model ensembling and task-specific tricks. According to the results, we observe that our models yield consistent advances on most of the tasks compared with public BERT and ELECTRA models under both base and large sizes.

We further evaluate the performance of our models on the challenging SQuAD MRC task. Table 2 shows the results, which indicate modest performance gains in the reading comprehension task. The results show that our method is not only effective for the sentence-level classification or regression tasks of NLU but also beneficial for passage-level reading comprehension.

5 Analysis

5.1 Ablation Study

To investigate the contribution of the internal components of the proposed IR objective, we conduct an ablation study under $\text{BERT}_{\text{base}}$ and $\text{ELECTRA}_{\text{base}}$ on the GLUE development set. Table 3 shows the performance when removing each one of the methods. We observe that removing either ECP or DPP objective will result in performance drop generally, which verifies the effectiveness of both methods.

5.2 Comparison with other distance measures

We apply KL divergence to measure the distance between distributions. We compare the performance for different distance measures by using mean-square error (MSE) loss. The average

Model	CoLA *Mcc*	SST-2 *Acc*	MRPC *Acc*	STS-B *Spear*	QQP *Acc*	MNLI *Acc*	QNLI *Acc*	RTE *Acc*	Average -
BERT-IR$_{\text{base}}$	61.39	93.46	87.50	89.05	90.90	85.28	91.84	68.95	83.52
- ECP	60.84	93.11	88.48	87.13	90.83	84.94	91.54	66.78	82.96
- DPP	59.90	93.23	87.01	87.43	90.89	84.70	91.43	67.87	82.81
ELECTRA-IR$_{\text{base}}$	68.95	95.30	90.44	90.52	91.40	88.66	93.04	79.06	87.17
- ECP	67.08	95.21	89.71	90.26	91.17	88.61	92.87	77.26	86.52
- DPP	67.75	95.18	89.21	90.35	91.28	88.50	92.75	76.89	86.49

Table 3: Ablation study of the proposed methods under BERT-base and ELECTRA-base on the GLUE development set. STS-B is reported by Spearman correlation, CoLA is reported by Matthew's correlation, and other tasks are reported by accuracy.

Model	CoLA *Mcc*	SST-2 *Acc*	MRPC *Acc*	STS-B *Spear*	QQP *Acc*	MNLI *Acc*	QNLI *Acc*	RTE *Acc*	Average -
BERT$_{\text{base}}$	30.88	87.84	74.75	76.06	87.77	75.70	84.24	56.68	71.74
BERT-IR$_{\text{base}}$	33.06	89.56	76.47	76.64	88.08	76.23	85.03	59.57	73.08

Table 4: Results of training BERT$_{\text{base}}$ and BERT-IR$_{\text{base}}$ from scratch.

Baseline	KL Divergence	MSE
82.32	83.52	83.27

Table 5: Comparison of using KL divergence and MSE to measure the distribution distances

GLUE scores (based on BERT-base) are shown in Table 5. We see that both IR methods contribute to better performance. The results further verify the general benefits of instance regularization for pre-training no matter what the distance function is.

5.3 Performance in Different Training Steps

To interpret the training effectiveness of our proposed method, we illustrate the performance of different training steps of BERT$_{\text{base}}$ and BERT-IR$_{\text{base}}$ on the development sets of the small-scale CoLA and the large-scale MNLI tasks, as shown in Figure 3. We see that the accuracy of the baselines boost slightly as the training steps increase. In contrast, our models can still yield obvious gains, which indicates our PrLM models could absorb extra beneficial signals via the newly proposed instance regularization approach.

5.4 Convergence Speed

Figure 4 shows the training curve of the BERT$_{\text{base}}$ and BERT-IR$_{\text{base}}$ models when training from scratch.[4] We observe that the absolute values of our approaches are relatively higher than the baselines at the very beginning. The reason is that our loss function is composed of three elements as formalized in Eq. 4. However, our model converges quickly. The loss of BERT-IR$_{\text{base}}$ falls below the baseline when the training goes on, and the slope of our curve is obviously larger than that of the baseline. In addition, we also evaluate the baseline and our model trained from scratch (Table 4), which achieve the average accuracy of 71.74% and 73.08% (+1.3%) on the GLUE datasets, respectively. The analysis above indicates that the PrLM model trained with our approach could absorb extra knowledge via the newly proposed instance regularization approach, and it would benefit the training of the vanilla masked language modeling as well.

5.5 Training Cost

Since the calculation of the regularization terms involves two forward passes of input sequences, we further investigate the influence of the training cost. Analysis shows that our model is efficient in training speed and parameter size. Taking the BERT-based model for example, the training

[4]For clear observation, we pre-train the baseline and our model from scratch instead of continuous training, because the checkpoints used for continuing training have already converged under a small loss, making it hard to interpret the convergence.

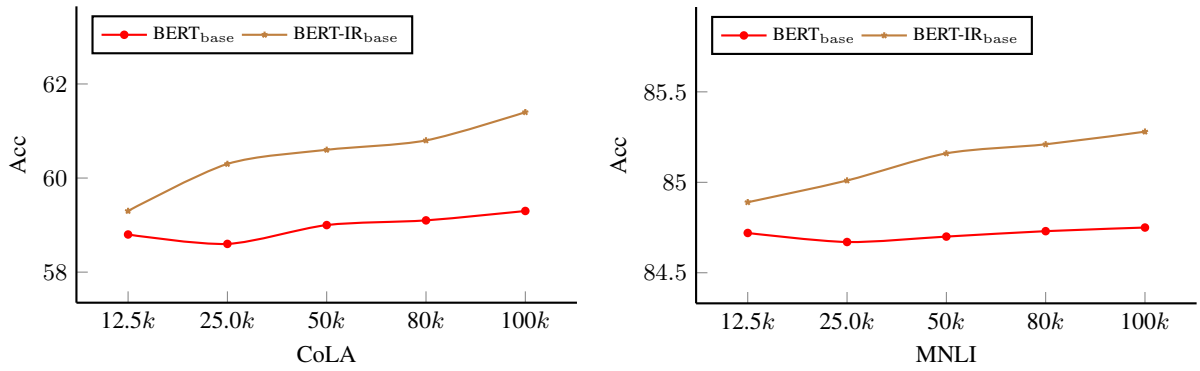

Figure 3: The performance (accuracy) of different training steps of BERT$_{base}$ and BERT-IR$_{base}$ on the CoLA and MNLI development sets.

Model	***Original Reference***		***SwapSynWordEmbedding***		***SwapSynWordNet***	
	EM Score	F1 Score	EM Score	F1 Score	EM Score	F1 Score
BERT$_{base}$	85.33	88.78	84.67 (↓0.67)	87.67 (↓1.11)	81.67 (↓3.67)	85.15 (↓3.63)
BERT-IR$_{base}$	84.33	87.70	84.67 (↑**0.33**)	87.82 (↑**0.12**)	82.33 (↓**2.00**)	85.42 (↓**2.28**)
ELECTRA$_{base}$	89.00	90.91	86.67 (↓2.33)	88.89 (↓2.02)	87.00 (↓2.00)	89.39 (↓1.53)
ELECTRA-IR$_{base}$	89.67	91.44	89.00 (↓**0.67**)	90.30 (↓**1.14**)	89.00 (↓**0.67**)	91.03 (↓**0.41**)

Table 6: Robustness evaluation on the SQuAD dataset. *Original* means the results of original dataset sampled from the SQuAD v1.1 development set by TextFlint (Wang et al., 2021), and *Swap.* indicates the transformed one. The assessed models are the base models from Table 2. In this analysis, the lower performance drop means the better.

time of BERT-IR$_{base}$ and BERT$_{base}$ baseline for 200K steps is 67h/74h (only 10% increase) with the same hyper-parameters on base models. The memory cost also keeps basically the same scale as the baseline since the regularization does not necessarily require extra gradient backpropagation.

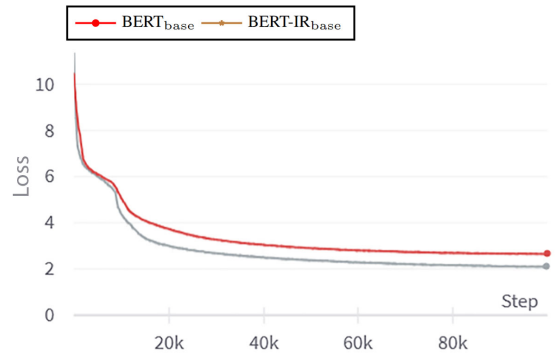

Figure 4: Training curve of the BERT-based models.

5.6 Robustness Against Synonym-Based Adversarial Attacks

The semantic agreement in IR measures the consistency between similar sentences, which may improve our model's robustness. To verify the hypothesis, we evaluate our models in Table 2 with synonym-based adversarial examples derived from the SQuAD v1.1 development set. The examples are generated by a robustness evaluation platform TextFlint (Wang et al., 2021), using *SwapSynEmbedding* and *SwapSynWordNet*, which transform an input by replacing its words with synonyms provided by GloVe embeddings (Pennington et al., 2014) or WordNet (Miller, 1998), respectively.

The results are shown in Table 6, from which we observe that the adversarial attacks can lead to an obvious performance drop of the baseline models, i.e., 3.67(EM)%/3.63(F1)% of BERT$_{base}$ on *SwapSynWordNet*. In contrast, our models perform less sensitively against the adversarial examples, and our BERT-IR$_{base}$ even yields an increase of scores in *SwapSynEmbedding* attack. The results indicate that the regularization helps the model to resist synonym-based adversarial attacks with less performance degradation.

6 Conclusion

In this paper, we study the instance-aware contribution estimation from the ennoising and

denoising processes in discriminative language model pre-training, motivated by the observation that the quality of denoising has to be subject to the complexity of the constructed training data from ennoising. The estimation is decomposed into ennoising corruption penalty and denoising prediction penalty, which are used as regularization terms for language model pre-training. Experiments show that language models trained with our regularization terms can yield improved performance on downstream tasks, with better robustness against adversarial attacks. In addition, the training efficiency can be improved as well, without severe costs of computation resources and training speed. We hope our work could facilitate related studies to improve training quality while keeping a lightweight model size.

7 Limitations

We acknowledge that the major limitation of the proposed method is additional computation compared with the vanilla language models because the calculation of the regularization terms involves two forward passes of input sequences. As discussed in Section 5.5, the training time of BERT-IR$_{\text{base}}$ and BERT$_{\text{base}}$ baseline for 200K steps is 67h/74h (about 10% increase) with the same hyper-parameters on base models. A more efficient instance regularization method without additional training passes could be future work.

References

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *ACL-PASCAL*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv preprint*, abs/2002.08909.

Yaru Hao, Li Dong, Hangbo Bao, Ke Xu, and Furu Wei. 2021. Learning to sample replacements for ELECTRA pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4495–4506.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. PMI-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. 2020. Task-specific objectives of pre-trained language models for dialogue adaptation. *ArXiv preprint*, abs/2009.04984.

Yian Li and Hai Zhao. 2021. Pre-training universal language representation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5122–5133.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2999–3007. IEEE Computer Society.

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. *ArXiv preprint*, abs/1907.11692.

Xueni Luo, Dawei Cheng, Haorui Ma, Junhao Wang, Mengzhen Fan, and Yifeng Luo. 2021. Leveraging domain information to classify financial documents via unsupervised graph momentum contrast. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3298–3302.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel Bowman. 2017. The RepEval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 1–10.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li,

Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Yi Xu and Hai Zhao. 2021. Dialogue-oriented pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2663–2673.

Atsuki Yamaguchi, George Chrysostomou, Katerina Margatina, and Nikolaos Aletras. 2021. Frustratingly simple pretraining alternatives to masked language modeling. *ArXiv preprint*, abs/2109.01819.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Peng Zhu, Dawei Cheng, Siqiang Luo, Ruyao Xu, Yuqi Liang, and Yifeng Luo. 2022. Leveraging enterprise knowledge graph to infer web events' influences via self-supervised learning. *Journal of Web Semantics*, page 100722.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society.