A Robust Adversarial Training Approach to Machine Reading Comprehension

Kai Liu,[1]* Xin Liu,[2]*† An Yang,[3]† Jing Liu,[1] Jinsong Su,[2] Sujian Li,[3] Qiaoqiao She[1]
[1]Baidu Inc., Beijing, China
[2]Xiamen University, Xiamen, China
[3]Key Laboratory of Computational Linguistics, Peking University, MOE, China
{liukai20, liujing46, sheqiaoqiao}@baidu.com, liuxin@stu.xmu.edu.cn,
{yangan, lisujian}@pku.edu.cn, jssu@xmu.edu.cn

Abstract

Lacking robustness is a serious problem for Machine Reading Comprehension (MRC) models. To alleviate this problem, one of the most promising ways is to augment the training dataset with sophisticated designed adversarial examples. Generally, those examples are created by rules according to the observed patterns of successful adversarial attacks. Since the types of adversarial examples are innumerable, it is not adequate to manually design and enrich training data to defend against all types of adversarial attacks. In this paper, we propose a novel robust adversarial training approach to improve the robustness of MRC models in a more generic way. Given an MRC model well-trained on the original dataset, our approach dynamically generates adversarial examples based on the parameters of current model and further trains the model by using the generated examples in an iterative schedule. When applied to the state-of-the-art MRC models, including QANET, BERT and ERNIE2.0, our approach obtains significant and comprehensive improvements on 5 adversarial datasets constructed in different ways, without sacrificing the performance on the original SQuAD development set. Moreover, when coupled with other data augmentation strategy, our approach further boosts the overall performance on adversarial datasets and outperforms the state-of-the-art methods.

Introduction

Machine Reading Comprehension (MRC) has become a popular research topic in recent years. A lot of efforts have been devoted to create better MRC models (Seo et al. 2016; Wang et al. 2017; Yu et al. 2018; Devlin et al. 2018). Specifically, recent advances suggest that several MRC models can achieve human parity on several datasets. However, (Jia and Liang 2017) revealed that, these advanced models are vulnerable to specially designed adversarial attacks. The model performance significantly decreases on the adversarial examples which consist of the original answer passages and the generated misleading texts. Similar problems have been observed in (Goodfellow, Shlens, and Szegedy 2014; Zhang et al. 2017). As shown in Table 1, different types of adversarial examples are all able to distract the MRC model. Therefore, it is a critical problem to improve the robustness of the existing MRC models.

To deal with the robustness issue mentioned above, researchers have made several attempts. Currently, the most straightforward and effective approach is to augment the training dataset with adversarial examples. (Wang and Bansal 2018) augmented the training datasets by incorporating adversarial examples that fit a certain type of attacks, and trained an MRC model on the augmented dataset. This significantly improves the model robustness under the known certain types of attacks. However, such augmented datasets are more capable of simulating the known types of adversarial examples, while ignoring other unobserved types. According to our observation, the augmented training dataset of (Wang and Bansal 2018) helps defense adversarial attacks proposed in (Jia and Liang 2017) well, but still fails on other types of adversarial attacks (shown in the experiment section). Hence, we render that rule-based data augmentation approach is not adequate since the types of adversarial examples are innumerable.

To deal with the above challenge, we propose a model-driven approach to generate adversarial examples that can attack given MRC model. Then, we retrain and strengthen the MRC model by using the generated adversarial examples. The major benefit of our approach is that it does not require any specification of adversarial attack types, and we expect our model is more robust under general adversarial attacks.

Specifically, our approach can be divided into three steps: (1) We take an MRC model as a black-box and obtains a perturbation word embedding sequence for each instance sampled from the original dataset. The perturbation word embedding sequences are likely to cause the MRC model give wrong predictions. (2) From each perturbation embedding sequence, We sample a word sequence. Then, we take the sampled sequence as a misleading text, and insert it into the original instance to create an adversarial example. (3) We retrain the MRC model on the original datasets with the adversarial examples generated from step 2. Then, repeat step

*These authors contributed equally to this work.

†This work was done while the author was doing internship at Baidu Inc.

Question	What distinct quality of combustion was absent from philogiston theory?
Passage	Highly combustible materials that leave little residue, such as wood or coal, were thought to be made mostly of phlogiston; whereas non-combustible substances that corrode, such as iron, contained very little. **Air** did not play a role in phlogiston theory, nor were any initial quantitative experiments conducted to test the idea; instead, it was based on observations of what happens when something burns, that most common objects appear to become lighter and seem to lose something in the process. The fact that a substance like wood gains overall weight in burning was hidden by the buoyancy of the gaseous combustion products......

Adv Types	**Adversarial Examples with Misleading Texts**	**Predictions**
AddSent	{passage} + *The distinct quality of combustion of a engine was present from philogiston theory.*	engine
AddAny	{passage} + *theory ? absent ? week ran we absent absent chief*	buoyancy
AddAnsCtx	{passage} + *did not play a role in phlogiston theory, nor were any initial quantitative experiments conducted to test the idea;*	buoyancy

Table 1: An example of attacking BERT$_{base}$(Devlin et al. 2018) by appending various forms of misleading texts to the ending of passage. Without any misleading texts appended, the model predicts the correct answer ”Air” (in bold) as the result. However, the model is distracted and predicts wrong answers when different types of misleading texts are added. The misleading texts are created in different ways: *AddSent:*(Jia and Liang 2017) generates the misleading text by modifying the question according to certain rules and proofreads manually; *AddAny:*(Jia and Liang 2017) automatically searches the misleading texts word by word on various MRC models; *AddAnsCtx:* we generate the misleading text by removing the answer words in answer sentences.

1 with the retrained model until it converges. In this way, we expect the well trained model is able to tackle more general attacks rather than specific types of adversarial examples.

The experimental results show that our approach can significantly improve the robustness of the MRC models over five different types of adversarial examples. Based on ERNIE2.0 (Sun et al. 2019), one of the state-of-the-art MRC models, our approach gains a significant improvement of 8.4% F1 score averaged on all types of adversarial test sets. The overall improvement on F1 score suggests that our training approach strengthens the model robustness in a more general way. Moreover, coupled with manually designed training data (Wang and Bansal 2018), our approach can further boost the average F1 score and gains a 2.3% improvement over different MRC models. Our contributions are concluded as follows:

- We proposed a model-driven approach to improve the robustness of MRC models to defend against various types of adversarial examples. Instead of specifying types of adversarial examples, our approach does not hold any assumption and it can generate the adversarial examples that distract the MRC models. The experimental results show that our approach is capable of tackling with more general attacks rather than specific types of adversarial examples.
- Our approach is a good supplement to other data augmentation methods. The robustness of MRC models can be further improved by using our training approach.

Related Work

Researchers have devoted their efforts to robustness problems of MRC systems in many ways. Most of them can be boiled down to two categories:

Data enrichment: A direct and effective way to defend against adversarial examples is to generate corresponding examples and train on them. (Jia and Liang 2017) designed some types of adversarial examples and investigated them on various MRC models. In order to defend those types of adversarial examples, (Wang and Bansal 2018) automatically created additional training samples based on rules, and enriched training data with those samples. Based on the specific designed training data, MRC models are able to achieve state-of-the-art performance on AddSent task. Differ from work mentioned above, our work is not designed for any specific adversarial data set, and we attempt to seek a more general way to strengthen model robustness.

Model improvement: Many researchers tried to better design and train MRC models in order to improve model robustness to defend against adversarial examples. (Salant and Berant 2018) improved models' robustness by using pretrained language model embeddings as inputs in order to collect rich contextualized information. (Min et al. 2018) proposed a sentence selector to select minimal sentence sets for further prediction so as to avoid many distractions. (Liu et al. 2018) designed an output layer that averages multi-predictions to improvement the model robustness. (Hu et al. 2018b) trained robust single model based on ensemble ones through distillation training approach. Instead of adding heuristic design to MRC models, our training approach does not have to modify any model structures, it can be applied to all derivable models.

Besides efforts devoted into MRC systems, many efforts are also devoted into adversarial attacking methods on text. (Behjati et al. 2019) tried to distract a text classifier by training perturbation embeddings. (Iyyer et al. 2018) proposed a syntactically controlled paraphrase networks to generate grammatically adversarial examples. (Gong et al. 2018) and (Sato et al. 2018) generated misleading texts via training perturbation embeddings and searching the nearest tokens. And (Jia and Liang 2017; Alzantot et al. 2018) generated misleading text automatically by replacing tokens in text iteratively until get a success attack. Differ from attacking methods mentioned above, our work not only finds more attacking ways, but also tries to improve model robustness in an effective way.

Adversarial Training Method

Similar to Generative Adversarial Networks (GAN), our adversarial training method plays a min-max game between an adversarial example generator and a corresponding MRC model. Since the generation of discrete tokens is not a differentiable process, instead of adopting reinforcement learning methods (Kusner and Hernández-Lobato 2016; Yu et al. 2017), we select a sampling strategy to generate adversarial examples. Our training process follows a three steps algorithm: (1) Takes a well trained MRC model as the adversarial generator, and trains perturbation embedding sequences to minimize output probabilities of real answers under given questions and passages. (2) Greedily samples word sequences from perturbation embeddings as misleading texts to create and enrich our adversarial example set. (3) Trains the MRC model to maximize probabilities of real answers to defend against those adversarial examples. Then return to step 1 with retrained model as new generator until convergence.

In order to fully cover potential types of adversarial examples, our approach attempts to generate two kinds of misleading texts:

1. Misleading answer texts: misleading texts which try to convince MRC models that correct answers are located within the texts.
2. Misleading context texts: misleading texts that act as contexts and try to distract MRC models from correct answers and guide them to wrong ones (not necessary within the misleading texts).

where each type owns its own loss function in adversarial training. Meanwhile, in order to increase the diversity, our approach also tries to control misleading texts to be either similar or dissimilar to questions, while most known misleading texts are mainly question related (Jia and Liang 2017; Wang and Bansal 2018).

MRC Model Definition

We treat all derivable MRC models as black-boxes, leaving network internal details alone and we focus on model inputs and outputs. Given a pair of question q and passage p as inputs, most MRC models $f(q, p; \theta)$ attempt to search an answer located in a span s=[s^s, s^e] that maximizes the model probability:

$$\begin{aligned} f(q,p;\theta) &= \underset{s}{argmax}\, Pr(s\,|q,p;\theta) \\ &= \underset{s^s,s^e}{argmax}\, Pr(s^s\,|q,p;\theta)\, Pr(s^e\,|q,p;\theta) \end{aligned} \tag{1}$$

where θ denotes the parameters of the MRC model and s^s, s^e denote the start and end positions of s respectively. In detail, q and p denote token sequences of $t_{q_1}t_{q_2}...t_{q_m}$ and $t_{p_1}t_{p_2}...t_{p_n}$ respectively, where m and n denote the sequence lengths of question and passage respectively.

Since main stream MRC models usually adopt word embedding layers as the bottom input blocks, we assume inputs of MRC models are embeddings and simplify the model as:

$$f(e_q, e_p; \theta) = \underset{s}{argmax}\, Pr(s|e_q, e_p; \theta) \tag{2}$$

where e_q denotes an embedding sequence $e_{q_1}e_{q_2}...e_{q_m}$ of question q, where $e_{q_i} \in R^d$ denotes the i-th token embedding of q with dimension size d. Similarly, e_p means the passage embedding sequence $e_{p_1}e_{p_2}...e_{p_n}$, and $e_{p_i} \in R^d$ denotes the i-th token embedding of p. With a given vocabulary embedding table $V \in R^{|V|\times d}$, both e_{q_i} and e_{p_i} are lookup results of t_{q_i} and t_{p_i}.

Perturbation Embedding Training

Similar to (Behjati et al. 2019; Gong et al. 2018; Sato et al. 2018), our perturbation adversarial training method aims to train a perturbation embedding sequence for each instance under the supervision of target model so as to distract it. During the training, we treat the model as a generator and all model parameters are fixed. With given inputs e_q and e_p, the training method only tries to perturb each passage input e_p with an additional perturbation embedding sequence.

Based on the definition of MRC model, for each training instance, we firstly insert a continuous perturbation embedding sequence e' into the passage embedding sequence e_p. Therefore, we replace the model input e_p as in Formula (2):

$$e_{p'} = e_{p_1}e_{p_2}...e_{p_k} \oplus e'_1e'_2...e'_l \oplus e_{p_{k+1}}...e_{p_n} \tag{3}$$

where $\oplus$ is the concatenation operator, k is the insert position index, l is the length of the e', and $e'_i \in R^d$ denotes the i-th embedding vector of the e'.

To limit searching space of e' and for the convenience of further sampling, we define the i-th embedding of e', e'_i as the weighted sum of vocabulary embeddings:

$$e'_i = \sum_{j=0}^{|V|} w_{ij}v_j \tag{4}$$

where w_{ij} is the weight of j-th vocabulary token for i-th position, while $v_j \in R^d$ is the embedding of the j-th token in V. We define $w \in R^{l\times|V|}$ as the weight matrix of e', and $w_i \in R^{|V|}$ is the weight vector of e'_i. In order to have the weight vector w_i normalized in vocabulary space, we define w_{ij} as the softmax result of trainable parameter $\alpha \in R^{l\times|V|}$:

$$w_{ij} = \frac{exp(\alpha_{ij})}{\sum\limits_k exp(\alpha_{ik})} \tag{5}$$

where α_{ij} is a trainable parameter for w_i. In this way, for each instance, training a perturbation embedding sequence is to finding a proper distribution of weight matrix w. We train a w for later procedures individually for each instance.

Based on the settings above, we expect our training method to be able to generate misleading answer or misleading context texts. To generate misleading answer texts and distract the MRC model, we design a cross entropy loss function aims to cheat the model and make the model believe the answer is locating in perturbation embedding sequence:

$$L_a = -\frac{1}{2} \sum_{y \in \{s_d^s, s_d^e\}} \log Pr(y|e_q, e'_p; \theta) \tag{6}$$

where s_d=[s_d^s, s_d^e] is the distract answer span located in perturbation embedding sequence. To generate misleading context texts, we design a loss function aims to minimize the model estimation on ground truth span s_g=[s_g^s, s_g^e] in order to distract the MRC model:

$$L_c = \frac{1}{2} \sum_{y \in \{s_g^s, s_g^e\}} \log Pr(y|e_q, e'_p; \theta) \tag{7}$$

In this way, we define our training loss function as:

$$L = L_a + \lambda_c L_c + R_s \tag{8}$$

where λ_c is the weight of L_c. To increase the diversity of our approach, we add a regularization term R_s to our loss function, which is defined as a similarity regularizer to control the similarity between perturbation embeddings and questions & answers:

$$R_s = \lambda_q sim(e', e_q) + \lambda_a sim(e', e_a) \tag{9}$$

where e_a denotes embedding sequence of the answer sentence, which is a sub sequence of e_p. Weights of both similarity terms are denoted as λ_q and λ_a. And $sim(\cdot, \cdot)$ is defined as a bag-of-words cosine similarity function:

$$sim(e_1, e_2) = cos(\sum_i e_{1i}, \sum_i e_{2i}) \tag{10}$$

In this way, our training method trains a perturbation embedding e' defined by w that minimizes the loss L:

$$E' = \underset{e'}{argmin} L \tag{11}$$

where E' is our target perturbation embedding sequence.

The overview of perturbation embedding training process is simply illustrated in Figure 1. With given distract span s_d and ground truth s_g as supervised signals, gradients are back-propagated through MRC model from top layer to the bottom. α is turned to train weighted sum embeddings e' in order to distract model from ground truth. And we repeat the training process for each instance until the loss L is converged or lower than a certain $threshold$, then return the weight matrix w for further sampling.

Greedy Sampling

To generate discrete misleading texts, for each position of E', we greedily sample the most representative token t'_i who has the shortest Euclidean distance between embedding $v_{t'_i}$ and e'_i:

$$t'_i = \underset{t'_i}{argmin}\, EUC(e'_i, v_{t'_i}) \tag{12}$$

where t'_i means the sampled token of i-th embedding e'_i. Naturally, we can regard the weight w_{ij} in $e'_i = \sum w_{ij} v_j$ as the importance of v_j and simply sample the maximum

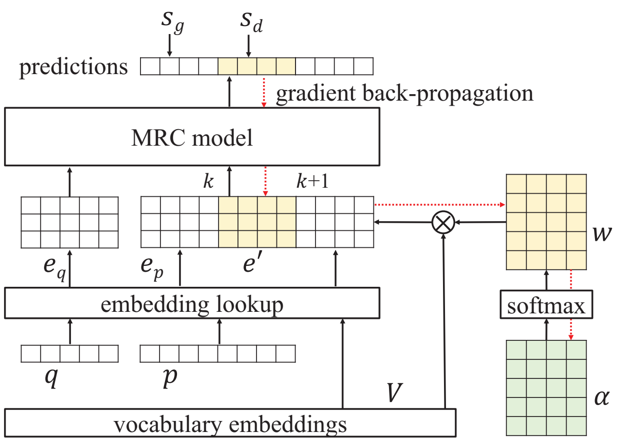

Figure 1: The overview of perturbation embedding training method. Grids in greens are trainable variables, grids in yellows are other variables determined by the trainable ones and the MRC model.

weighted v_j as the most representative embedding. As a result, our greedy sampling method can be simplified to sample the max weighted token: [§]

$$\begin{aligned} t'_i &= \underset{t'_i}{argmin}\, EUC(\sum w_{ij} v_j, v_{t'_i}) \\ &\approx \underset{t'_i}{argmax}\, w_{it'_i} \end{aligned} \tag{13}$$

where we also regard the t'_i of $w_{it'_i}$ as corresponding index of the vocabulary since a unique token has unique index in the vocabulary. Therefore, for each instance, generating a misleading text is sampling a max weighted token sequence $A = t'_1 t'_2 ... t'_l$ from the well trained w.

Retraining with Adversarial Examples

To train a more robust MRC model, we enrich training data with sampled adversarial examples and retrain our models on the enriched data. Given a misleading text and its corresponding triple data $\langle q, p, s_g \rangle$, we insert the misleading text A back into its position k of the passage. And we create an adversarial example $\langle q, p', s'_g \rangle$ with the modified passage p', where s'_g denotes the new ground truth span after the misleading text was inserted. Augmenting with these adversarial examples, we retrain the MRC model θ on the enriched training data and get the new model θ'.

Training Strategy

As shown in Algorithm 1, our adversarial training strategy can be divided into several steps. For each iteration, we randomly sample a sub training set $\{\langle q, p, s_g \rangle\}'$ from full training data set $\{\langle q, p, s_g \rangle\}$ for adversarial training through sampling function $SubSample(\cdot)$. With given sub training

[§]It is easy to prove that the equation can be established when $|v| = 1$ and $w_{it'_i} \geq 0.5$.

Algorithm 1: Adversarial Training Strategy

```
Input: The training set of a triple $\{\langle q,p,s_g\rangle\}$; The adversarial example length $l$; A well trained MRC model $\theta$; Max iteration time $T$;
Output: Adversarially trained model $\theta^{'}$;
1  while $trainloss < \epsilon$ and $t < T$ do
2      $\{\langle q,p,s_g\rangle\}' \leftarrow SubSample(\{\langle q,p,s_g\rangle\})$;
3      $\{w\}, trainloss \leftarrow PertTrain(\{\langle q,p,s_g\rangle\}';\theta)$;
4      $\{A\} \leftarrow Greedy(\{w\})$;
5      $\{\langle q,p',s_g'\rangle\}' \leftarrow Create(\{\langle q,p,s_g\rangle\}', \{A\})$;
6      $\{\langle q,p,s_g\rangle\}'' \leftarrow \{\langle q,p,s_g\rangle\} \cup \{\langle q,p',s_g'\rangle\}'$ ;
7      $\theta^{'} \leftarrow Train(\{\langle q,p,s_g\rangle\}'')$;
8      $\theta, \{\langle q,p,s_g\rangle\} \leftarrow \theta^{'}, \{\langle q,p,s_g\rangle\}''$
9  end
10 return $\theta^{'}$
```

set and model θ, firstly we train the weight matrix w and collect its average training loss ($trainloss$) through perturbation embedding training process $PertTrain(\cdot;\cdot)$. Secondly, we greedily sample misleading texts $\{A\}$ from $\{w\}$ ($Greedy(\cdot)$) and have them correspondingly inserted back to the passages to create adversarial examples ($Create(\cdot,\cdot)$). Thirdly, with given enriched training data set $\{\langle q,p,s_g\rangle\}''$, we retrain model θ by $Train(\cdot)$. Then we replace the model and training data with the enhanced ones for later iteration. The algorithm starts again until $trainloss$ is greater than a threshold ϵ or the maximum iteration time T is reached.

Experiments

In this section, we evaluate the performances of our training approach on different training and test sets based on four different MRC models. Then we discuss why our approach might be a more generic one by investigating the distributions of various adversarial example types. Besides, we further investigate performance impacts of various model settings.

Dataset and Systems

In the experiment, we train MRC models on SQuAD (Rajpurkar et al. 2016) training set and its enhanced version AddSentDiverse (Wang and Bansal 2018) respectively. We test the models on six different test sets, i.e. standard SQuAD development set and five different types of adversarial test sets. All adversarial test sets are appended with misleading texts as parts of their passages based on standard SQuAD development set,

Training Datasets:

- SQuAD (Rajpurkar et al. 2016): One of the most popular MRC datasets. The dataset consists of 87.5K question-answer training pairs which documents are dumped from Wikipedia and question-answer are annotated by crowdsourcing. And we select the SQuAD v1.0 as the training dataset
- AddSentDiverse (ASD) (Wang and Bansal 2018): Based on the observation of AddSent (Jia and Liang 2017), they enriched SQuAD training data with correspondingly designed adversarial examples. And the size of dataset reaches 109.4K.

Test Sets:

- SQuAD (DEV) (Rajpurkar et al. 2016): The development set of SQuAD v1.0 in which contains 10K triple $\langle q,p,s_g\rangle$ instances for evaluation.
- AddSent (AS) (Jia and Liang 2017): Grammatical adversarial test set in which misleading texts are converted from questions through rules and crowdsourcing. The dataset contains 1k question instances.
- AddAny (AA) (Jia and Liang 2017): Ungrammatical adversarial test set which misleading texts are automatically generated according to question words and common words. The dataset contains 1k question instances.
- AddAnyExtend (AAE): This is our implementation of $AddAny$ with extended vocabulary which contains not only question words but also high frequency words, passage words and random common words. The dataset contains 2.6k question instances.
- AddAnsCtx (AAC): A test set that uses answer context as misleading texts which includes 10k instances. The misleading texts are answer sentences with answer tokens removed.
- AddNegAns (ANA): A test set that uses negative expressions of fake answers as misleading texts, which includes 5k instances. We create misleading texts by changing answer sentence from "A is $answer$" to "A is not $fakeanswer$" (e.g., "I like this *book*" → "I *do not* like this *movie*").

Based on datasets mentioned above, we test four different MRC baseline systems: QANet (Yu et al. 2018), BERT (Base & Large) (Devlin et al. 2018) and ERNIE2.0 (Sun et al. 2019). We have them trained by our adversarial training approach respectively. We adopt F1 score as the main evaluation metric.

Experiment Settings

In perturbation embedding training phase, we randomly insert perturbation embedding between sentences, and have embeddings randomly initialized. During the embedding training, we set the batch size of QANet to be 32, BERT$_{base}$ 12, BERT$_{large}$/ERNIE 2.0 to be 4. We limit the perturbation sequence length l to be 10. For each batch, we randomly set λ_q, λ_p to be -10 or 10, and set λ_c to be 0.5. And we set s_d with random length in the middle of each perturbation embedding. To determine the convergence of the embedding training process, we set the $threshold$ as 1.5 and we set the maximum training step as 200 because the most training losses tend to be stable (differences are lower than 1e-3) around 200 steps.

In training iteration, we set maximum training time T to be 5, $trainloss$'s stopping threshold ϵ to be 12.0. In order to

Model	DEV	AS	AA	AAE	AAC	ANA	average
Baseline systems							
QANet	83.3	48.0	54.8	63.7	**78.1**	**79.8**	64.9
BERT$_{base}$	88.4	49.9	47.8	52.8	74.4	69.9	59.0
BERT$_{large}$	90.6	60.2	60.4	64.1	81.1	80.1	69.2
ERNIE2.0	92.5	64.8	68.2	68.7	86.8	89.1	75.5
Related works							
DCN+MINIMAL(Min et al. 2018)	80.6	59.7	-	-	-	-	-
R.M-Reader(Hu et al. 2018a)	86.3	58.5	-	-	-	-	-
RMR+A2D(Hu et al. 2018b)	87.9	61.3	-	-	-	-	-
QANet+AddSentDiverse	80.7	**73.4**	54.3	61.2	74.5	75.7	67.8 (+2.9)
BERT$_{base}$+AddSentDiverse	88.3	80.4	84.9	61.2	76.9	73.5	75.4 (+16.4)
BERT$_{large}$+AddSentDiverse	90.4	86.3	87.3	70.5	82.2	82.4	81.7 (+12.5)
ERNIE2.0+AddSentDiverse	92.2	89.1	90.0	74.7	88.3	88.1	86.0 (+10.5)
Our approach							
QANet+Ours	82.1	49.2	**66.9**	**67.3**	77.2	79.3	68.0 (+3.1)
QANet+AddSentDiverse+Ours	80.8	72.2	61.9	63.2	75.3	76.0	**69.7 (+4.8)**
BERT$_{base}$+Ours	87.9	57.2	83.5	**70.7**	76.9	76.9	73.0 (+14.0)
BERT$_{base}$+AddSentDiverse+Ours	87.8	**81.7**	**85.2**	70.0	**79.2**	**79.2**	**79.1 (+20.1)**
BERT$_{large}$+Ours	90.4	63.1	88.3	**76.6**	81.6	81.7	78.3 (+9.1)
BERT$_{large}$+AddSentDiverse+Ours	90.1	**86.7**	**88.5**	75.7	**84.3**	**84.5**	**84.0 (+14.8)**
ERNIE2.0+Ours	92.5	70.8	**92.0**	**79.6**	88.2	**89.1**	83.9(+8.4)
ERNIE2.0+AddSentDiverse+Ours	91.9	**89.6**	91.2	79.4	**88.5**	88.2	**87.4(+11.9)**

Table 2: Experiment results on SQuAD develop set and adversarial test sets. All scores are F1 scores in percentage.

seek a balance between effectiveness and efficiency, we randomly sample 5% training data for adversarial training and larger ratios will not provide satisfied performance within a single iteration according to our early experiments. In sampling phase, we save all successful embeddings for greedy sampling. After sampling, we retrain MRC models follow the early stopping strategy (Bassler et al. 2010).

In order to generate misleading texts effectively, for each training instance, we utilize a local vocabulary V, in which tokens are mainly related to questions and passages. We collect the local vocabulary for each batch in different ways: high-frequency words; top-10 similar words who have shortest distance to tokens of the q & a in embedding space (cosine similarity); synonyms and antonyms regrading to questions and passages, gathered from WordNet (Miller 1995); hypernyms and hyponyms are similarly gathered from WordNet. To make the model easier to converge, the vocabulary size is limited to 200.

Results and Discussions

Experiment Results All baseline results are shown in Table 2. The results indicate that baseline systems are all vulnerable when facing grammatical or ungrammatical adversarial test sets. Although transformer based systems gain superior performances on all test sets, great reductions have been observed on all adversarial test sets. Comparing to BERT$_{base}$, QANet has a weaker reading performance but better robustness performances due to its simpler model design which usually leads to robuster performances.

Based on the same systems, our training approach has shown its effectiveness on all adversarial test sets. It gains an F1 improvement of 23.8% at most and an average of 8.4% on ERNIE2.0. Moreover, it gains even better average improvements of 14.0%/9.1% on BERT$_{base}$/BERT$_{large}$. It is not so consistent that QANet gains limited improvements due to its simpler model design which might limit its model ability. However, our approach still promotes its average performance by 3.1% in F1 score. The improvement on all adversarial test sets and systems indicate that it is able to enhance the robustness of MRC models.

The experimental results also suggest that the enhanced training data ASD (Wang and Bansal 2018) gains great improvement on some of adversarial test sets. Since the training data is targeting on question related test sets (AS, AA), it works very well on these test sets as expected. However, its improvement on other test sets (AAE, AAC, ANA), which are not in the form of questions , are not as great as it is on question related test sets. For all baseline systems, it only gains an average improvement of 1.7% in F1 on AAE, AAC, ANA. Such difference suggests the rule-based method might not be capable to handle all possible types of the adversarial examples. By contrast, our approach shows better robustness on unobserved types of adversarial examples and it gains an average improvement of 4.7% in F1 score on AAE, AAC, ANA test sets. Our approach does not set any strong assumption on test sets, so we believe that our approach has a stronger ability to strengthen model robustness in a more general way.

Moreover, the performances shown in Table 2 suggest that our approach, coupled with ASD dataset, can further improve the performance with an average F1 improvement of 3.7%/2.3%/1.4% on BERT$_{base}$/BERT$_{large}$/ERNIE2.0 re-

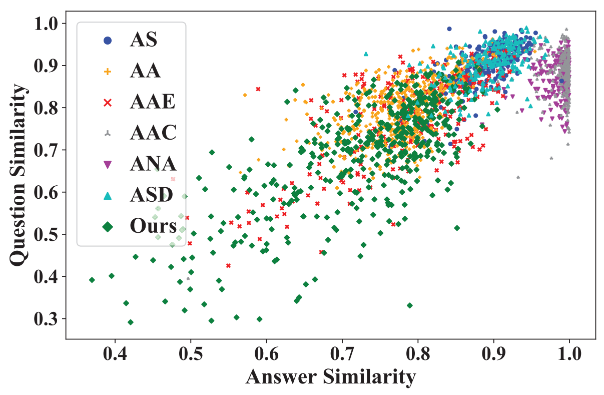

Figure 2: Misleading text distributions of different test sets and training samples.

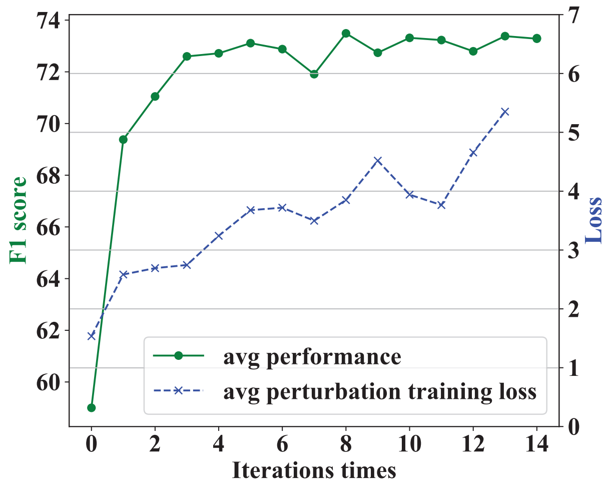

Figure 3: Performance on varying iterations.The solid line denotes the average performance of adversarial test sets. The dotted line denotes the average training loss during perturbation embedding training phase.

spectively. Considering that the ceiling performance on development set is about 92%, our training approach can boost the average performance from 86.0% to 87.4% bringing an over 20% decrease in error rate. These observations indicate our general approach can be a good supplement to (Wang and Bansal 2018) which is rule-based designed.

Adversarial Example Distribution Analysis To have a deeper insight of different types of adversarial examples, we investigate the misleading text distributions in a two-dimensional coordinate system, where X-axis is the answer sentence similarity and Y-axis is the question similarity. Both similarities are bag-of-words embedding cosine similarities between misleading texts and answers & questions. The results are shown in Figure 2, and as expected, ASD dataset has more overlaps with AS and AA. As a result, it gains better improvement on these test sets. By contrast, our data has a more extensive distribution in the space. Its extensiveness enable itself to cover more types of adversarial examples, which can partly explain the general improvement in Table 2 on various test sets.

Results of Different Iterations In order to investigate how the number of training iteration influences the performance, we test BERT$_{base}$ model using iteration time in a range of [1, 14], and it is the baseline performance when the time is equal to 0. As results shown in Figure 3, the growing performance curve suggest the effectiveness of our iterative training algorithm, and a good performance can be achieved at iteration around 4 or 5. But limited improvements can be obtained in later iterations as the curve suggesting. There is a probable explanation is that the perturbation embeddings are harder to cheat the model in later iterations due to the growing perturbation embedding training loss. Since our misleading texts are not “real” enough to confuse humans, we believe our approach can be further improved by better searching of perturbation embeddings and better sampling of meaningful misleading texts.

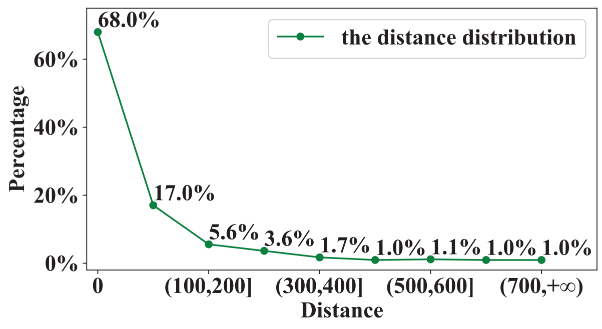

Figure 4: Distance distribution (in character) between wrong predictions and misleading texts. The distance is 0 if parts of the wrong prediction are located in the misleading text.

Distance Distribution of Wrong Predictions In order to investigate distance correlations between the wrong prediction and the misleading text, we examine the distances between boundary of predictions and misleading texts based on effective adversarial examples. As shown in Figure 4, more than two thirds of wrong predictions locate in misleading texts. And act as misleading contexts, misleading texts are more likely to guide the model to wrong answers nearby. This observation suggests that it is much easier to generate misleading answers than misleading contexts, and the localization suggests that misleading texts are able to draw the attention of mode to nearby deceptive “answers”.

Ablation Study To investigate the effectiveness of different parts of our model, we test our approach on BERT$_{base}$ with corresponding parts removed, and results are shown in Table 4. According to results, between two loss functions, L_a plays a much more important role in perturbation embedding training. It indicts that “answer” contained misleading texts might be more effective in our approach. And similar observations in the experiment above can also support

Models	Generated Misleading Texts	Predictions
QANet	*exactly player maybe are burnt tend think exactly off best*	burnt
BERT$_{base}$	*and distinct more combustion ##m no could theory common away*	combustion ##m no
BERT$_{large}$	*burns of ##on , since absent theory quantitative ##t of*	quantitative
ERNIE2.0	*something on not absent theory absent nor idea lighter quality*	lighter quality

Table 3: Generated misleading texts of different models. Corresponding question and passage are shown in Table 1. Examples of transformer models are in subword units (Sennrich, Haddow, and Birch 2015).

Model	average	Δ
BERT$_{base}$	73.0	
$-$answer loss L_a	64.1	-8.9
$-$context loss L_c	70.9	-2.1
$-$similarity regularizer R_s	69.7	-3.3

Table 4: Ablation experiment results.

this claim. For similarity regularizer R_s, the reduction suggests that the extensiveness in token distribution enhances the model robustness.

Misleading Text Analysis To investigate the quality of misleading texts, we sample misleading text results of different models and have them shown in Table 3. We can noticed that all texts are ungrammatical and meaningless sequences but they successfully distract baseline systems. It suggests current models such like BERT$_{large}$ and ERNIE2.0 though have excellent language modeling ability is still vulnerable to unpredictable noises. As a result, there is still much room for further improvement of our generation method so as to generate much misleading texts.

Conclusion

Current state-of-the-art MRC models are vulnerable facing different types of adversarial examples and the lack of robustness becomes a serious problem. Since it is not practical to enumerate and investigate all possible types of adversarial examples, improving robustness MRC models by investigating their weaknesses and strengthening them is a more effective way. As a result, we proposed a simple but effective adversarial training approach to enhance MRC model robustness via training these models on generated adversarial examples. With a three step algorithm, our approach train a robust MRC model by playing a min-max game between an adversarial example generator and the model trainer, where the generator is the model itself. Based on strong baselines, experiments on various test sets show that our novel approach can substantially boost MRC model robustness performance in a more general way. Further more, coupled with augmented training data, which is rule-based designed, our approach is able to further improve model robustnesses and outperform start-of-the-art performances.

Acknowledgments

This work is supported by the National Key Research and Development Project of China (No. 2018AAA0101900), the Natural Science Foundation of China (No.61533018 and No.61672440) and Baidu-Peking University Joint Project. We also thank Sheng Lu and the anonymous reviewers for their constructive criticism of the manuscript.

References

Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.-J.; Srivastava, M.; and Chang, K.-W. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

Bassler, D.; Briel, M.; Montori, V. M.; Lane, M.; Glasziou, P.; Zhou, Q.; Heelsansdell, D.; Walter, S. D.; Guyatt, G. H.; and Group, A. S. 2010. Stopping randomized trials early for benefit and estimation of treatment effects: Systematic review and meta-regression analysis. *Jama the Journal of the American Medical Association* 303(12):1180.

Behjati, M.; Moosavi Dezfooli, S. M.; Soleymani Baghshah, M.; and Frossard, P. 2019. Universal adversarial attacks on text classifiers.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Gong, Z.; Wang, W.; Li, B.; Song, D.; and Ku, W.-S. 2018. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Hu, M.; Peng, Y.; Huang, Z.; Qiu, X.; Wei, F.; and Zhou, M. 2018a. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 4099–4106.

Hu, M.; Peng, Y.; Wei, F.; Huang, Z.; Li, D.; Yang, N.; and Zhou, M. 2018b. Attention-guided answer distillation for machine reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2077–2086. Brussels, Belgium: Association for Computational Linguistics.

Iyyer, M.; Wieting, J.; Gimpel, K.; and Zettlemoyer, L. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Jia, R., and Liang, P. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2021–2031.

Kusner, M. J., and Hernández-Lobato, J. M. 2016. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

Liu, X.; Shen, Y.; Duh, K.; and Gao, J. 2018. Stochastic answer networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1694–1704. Melbourne, Australia: Association for Computational Linguistics.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Min, S.; Zhong, V.; Socher, R.; and Xiong, C. 2018. Efficient and robust question answering from minimal context over documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 1725–1735.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2383–2392.

Salant, S., and Berant, J. 2018. Contextualized word representations for reading comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 554–559. New Orleans, Louisiana: Association for Computational Linguistics.

Sato, M.; Suzuki, J.; Shindo, H.; and Matsumoto, Y. 2018. Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917*.

Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603.

Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Hua, W.; and Wang, H. 2019. Ernie 2.0: A continual pre-training framework for language understanding. *ArXiv* abs/1907.12412.

Wang, Y., and Bansal, M. 2018. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 575–581.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 189–198.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Yu, A. W.; Dohan, D.; Luong, M.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Zhang, G.; Yan, C.; Ji, X.; Zhang, T.; Zhang, T.; and Xu, W. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 103–117. ACM.