The Thirty-Fourth AAAI Conference on Artiﬁcial Intelligence (AAAI-20)

**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

1Baidu Inc., Beijing, China 2Xiamen University, Xiamen, China

3Key Laboratory of Computational Linguistics, Peking University, MOE, China

{liukai20, liujing46, sheqiaoqiao}@baidu.com, [liuxin@stu.xmu.edu.cn,](mailto:liuxin@stu.xmu.edu.cn)

{yangan, lisujian}@pku.edu.cn, [jssu@xmu.edu.cn](mailto:jssu@xmu.edu.cn)

### Abstract

Lacking robustness is a serious problem for Machine Read- ing Comprehension (MRC) models. To alleviate this prob- lem, one of the most promising ways is to augment the train- ing 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 de- fend against all types of adversarial attacks. In this paper, we propose a novel robust adversarial training approach to im- prove 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 sched- ule. When applied to the state-of-the-art MRC models, in- cluding 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. More- over, when coupled with other data augmentation strategy, our approach further boosts the overall performance on ad- versarial datasets and outperforms the state-of-the-art meth- ods.

# 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). Specif- ically, 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 adversar- ial 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, re- searchers 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 incorporat- ing adversarial examples that ﬁt a certain type of attacks, and trained an MRC model on the augmented dataset. This sig- niﬁcantly improves the model robustness under the known certain types of attacks. However, such augmented datasets are more capable of simulating the known types of adver- sarial examples, while ignoring other unobserved types. Ac- cording 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 sec- tion). Hence, we render that rule-based data augmentation approach is not adequate since the types of adversarial ex- amples 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 exam- ples. 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 per- turbation word embedding sequence for each instance sam- pled from the original dataset. The perturbation word em- bedding sequences are likely to cause the MRC model give wrong predictions. (2) From each perturbation embedding

Copyright *§*c 2020, Association for the Advancement of Artiﬁcial

sequence, We sample a word sequence. Then, we take the

Intelligence (www.aaai.org). All rights reserved.

∗These authors contributed equally to this work.

†This work was done while the author was doing internship at

Baidu Inc.

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 ad- versarial examples generated from step 2. Then, repeat step

**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* engine

{ }

*theory.*

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 experi-* buoyancy

{ }

*ments conducted to test the idea;*

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 train- ing approach strengthens the model robustness in a more general way. Moreover, coupled with manually designed training data (Wang and Bansal 2018), our approach can fur- ther boost the average F1 score and gains a 2.3% improve- ment over different MRC models. Our contributions are con- cluded as follows:

We proposed a model-driven approach to improve the ro- bustness of MRC models to defend against various types of adversarial examples. Instead of specifying types of adversarial examples, our approach does not hold any as- sumption 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 augmen- tation methods. The robustness of MRC models can be further improved by using our training approach.

•

# Related Work

Researchers have devoted their efforts to robustness prob- lems 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 ad- versarial examples, (Wang and Bansal 2018) automatically created additional training samples based on rules, and en- riched training data with those samples. Based on the spe- ciﬁc 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 ensem- ble ones through distillation training approach. Instead of adding heuristic design to MRC models, our training ap- proach 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 train- ing 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 per- turbation embeddings and searching the nearest tokens. And (Jia and Liang 2017; Alzantot et al. 2018) generated mis- leading text automatically by replacing tokens in text itera- tively until get a success attack. Differ from attacking meth-

ods mentioned above, our work not only ﬁnds more attack- ing ways, but also tries to improve model robustness in an effective way.

# Adversarial Training Method

Similar to Generative Adversarial Networks (GAN), our ad-

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

*f* (*eq, ep*; *θ*)= *argmax Pr*(*s e q,e p*; *θ*) (2)

|

*s*

where *eq* denotes an embedding sequence *eq eq ...eq* of

versarial training method plays a min-max game between

question

1 2 *m*

*d*

an adversarial example generator and a corresponding MRC

ding of

1

2

*n*

*i*

*q*, where *eqi* ∈ *R* denotes the *i*-th token embed-

model. Since the generation of discrete tokens is not a dif- ferentiable process, instead of adopting reinforcement learn-

*q* with dimension size *d*. Similarly, *ep* means the passage embedding sequence *ep ep ...ep* , and *ep* ∈ *Rd*

ing methods (Kusner and Herna´ndez-Lobato 2016; Yu et

denotes the *i*-th token embedding of *p*. With a given vocab-

ulary embedding table *V* ∈ *R*|*V* |×*d*, both *eq* and *ep* are

al. 2017), we select a sampling strategy to generate adver- sarial examples. Our training process follows a three steps

lookup results of *tqi*

*i i*

and *tpi* .

algorithm: (1) Takes a well trained MRC model as the ad- versarial generator, and trains perturbation embedding se- quences to minimize output probabilities of real answers under given questions and passages. (2) Greedily samples word sequences from perturbation embeddings as mislead- ing 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 ex- amples, our approach attempts to generate two kinds of mis- leading 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 con- texts 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 adversar- ial training. Meanwhile, in order to increase the diversity, our approach also tries to control misleading texts to be ei- ther 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 in- puts, most MRC models *f* (*q, p*; *θ*) attempt to search an an-

## 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 *eq* and *ep*, the training method only tries to perturb each passage input *ep* with an additional perturbation embedding sequence.

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

*ept* = *ep*1 *ep*2 *...epk* ⊕ *e*×1*e*×2*...e*×*l* ⊕ *epk*+1 *...epn* (3)

where is the concatenation operator, *k* is the insert posi- tion index, *l* is the length of the *e*×, and *e*×*i Rd* 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:

|*V* |

Σ

*e*×*i* = *wijvj* (4)

*j*=0

where *wij* is the weight of *j*-th vocabulary token for *i*-th position, while *vj* ∈ *Rd* is the embedding of the *j*-th token in *V* . We deﬁne *w* ∈ *Rl*×|*V* | as the weight matrix of *e*×, and *wi R*|*V* | is the weight vector of *e*×*i*. In order to have the weight vector *wi* normalized in vocabulary space, we deﬁne *wij* as the softmax result of trainable parameter *α* ∈ *Rl*×|*V* |:

∈

swer located in a span *s*=[*ss*, *se*] that maximizes the model

*exp*(*α* )

probability:

*f* (*q, p*; *θ*)= *argmax Pr*(*s* |*q, p*; *θ*)

*w* = *ij ij exp*(*α* )

*k*

*ik*

(5)

*s*

= *argmax Pr*(*ss* |*q, p*; *θ*) *Pr*(*se* |*q, p*; *θ*)

Σ

(1)

where *αij* is a trainable parameter for *wi*. In this way, for each instance, training a perturbation embedding sequence

*ss,se*

is to ﬁnding a proper distribution of weight matrix *w*. We

where *θ* denotes the parameters of the MRC model and *ss*, *se* denote the start and end positions of *s* respectively. In detail, *q* and *p* denote token sequences of *tq*1 *tq*2 *...tqm* and *tp*1 *tp*2 *...tpn* respectively, where *m* and *n* denote the se- quence lengths of question and passage respectively.

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 mis- leading 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:



predictions

sg sd

gradient back-propagation

MRC model

e e

k

k+1

×

q

q

embedp ding lookup

vocabpulary embeddings

e'

V

softmax

1 Σ

*L* = − log *Pr*(*y*|*e , e*× ; *θ*) (6)

*a*

2

*q*

*p*

*d*

*d*

*y*∈{*ss,se* }

where *sd*=[*ss*, *se*] is the distract answer span located in per- w

*d*

*d*

turbation embedding sequence. To generate misleading con- text texts, we design a loss function aims to minimize the model estimation on ground truth span *sg*=[*ss*, *se*] in order

*g g*

to distract the MRC model:

1

Σ

*L* =

*c*

2

*y*∈{*ss ,se* }

*g*

*g*

log *Pr*(*y*|*e , e*× ; *θ*) (7) α

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

*q*

*p*

*L* = *La* + *λcLc* + *Rs* (8)

where *λc* is the weight of *Lc*. To increase the diversity of our approach, we add a regularization term *Rs* to our loss func- tion, which is deﬁned as a similarity regularizer to control the similarity between perturbation embeddings and ques- tions & answers:

*Rs* = *λqsim*(*e*×*, eq*)+ *λasim*(*e*×*, ea*) (9)

where *ea* denotes embedding sequence of the answer sen- tence, which is a sub sequence of *ep*. Weights of both sim-

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

weighted *vj* as the most representative embedding. As a re- sult, our greedy sampling method can be simpliﬁed to sam- ple the max weighted token: §

ilarity terms are denoted as *λq* and *λa*. And *sim*( *,* ) is de- ﬁned as a bag-of-words cosine similarity function:

· ·

*i*

*sim*(*e*1*, e*2)= *cos*(Σ *e*1*i,* Σ *e*2*i*) (10)

*i*

*i*

*t*×*i* = *argmin EUC*(

*tti*

Σ

≈ *argmax witti*

*tti*

*wijvj, vtt* )

(13)

In this way, our training method trains a perturbation em- bedding *e*× deﬁned by *w* that minimizes the loss *L*:

*E*× = *argminL* (11)

*et*

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 *sd* and ground truth *sg* 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 con-

where we also regard the *t*×*i* of *witt* 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

*i*

*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 corre- sponding triple data *q, p, sg* , we insert the misleading text *A* back into its position *k* of the passage. And we create an

( )

adversarial example (*q, p*×*, s*×*g* ) with the modiﬁed passage *p*×,

verged 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 *vtt* and *e*×*i*:

*i*

*t*×*i* = *argmin EUC*(*e*×*i, vtt* )

*t i* (12)*t*

*i*

where *s*×*g* denotes the new ground truth span after the mis- leading text was inserted. Augmenting with these adversar- ial 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 {(*q, p, sg*)}× from full training data set {(*q, p, sg*)} for adversarial training through

where *t*× means the sampled token of *i*-th embedding *e*× .

Naturally, we can regard the weight *wij* in *ei* =

*wijvj*

It is easy to prove that the equation can be established when

sampling function *SubSample*(·). With given sub training

*i* × Σ *i* §

as the importance of *vj* and simply sample the maximum

*|v|* =1 and *witti ≥* 0*.*5.

**Algorithm 1:** Adversarial Training Strategy

**Input:** The training set of a triple *q, p, sg* ; The adversarial example length *l*; A well trained MRC model *θ*; Max iteration time *T* ;

{( )}

**Output:** Adversarially trained model *θt* ;

**1 while** *trainloss < s and t < T* **do**

**2** {(*q, p, sg*)}× ←− *SubSample*({(*q, p, sg*)});

×

**3** {*w*}*, trainloss*←− *PertTrain*({(*q, p, sg*)} ; *θ*);

**4** {*A*} ←− *ttreedy*({*w*});

**5** {(*q, p*×*, s*×*g* )}× ←− *Create*({(*q, p, sg*)}×*,* {*A*});

**6** {(*q, p, sg*)}×× ←− {(*q, p, sg*)} ∪ {(*q, p*×*, s*×*g* )}× ;

**7** *θt* ←− *Train*({(*q, p, sg*)}××);

**8** *θ,* {(*q, p, s* )} ←− *θt ,* {(*q, p, s* )}××

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 *q, p, sg* instances for evaluation.

( )

•

AddSent (AS) (Jia and Liang 2017): Grammatical adver- sarial test set in which misleading texts are converted from questions through rules and crowdsourcing. The dataset

•

*g g*

**9 end**

**10 return** *θt*

contains 1k question instances.

AddAny (AA) (Jia and Liang 2017): Ungrammatical ad- versarial test set which misleading texts are automati- cally generated according to question words and common words. The dataset contains 1k question instances.

•

set and model *θ*, ﬁrstly we train the weight matrix *w* and collect its average training loss (*trainloss*) through per- turbation embedding training process *PertTrain*( ; ). Sec- ondly, we greedily sample misleading texts *A* from *w* (*ttreedy*( )) and have them correspondingly inserted back to the passages to create adversarial examples (*Create*( *,* )). Thirdly, with given enriched training data set *q, p, sg* ××,

·

{( )}

· ·

·

{ } { }

· ·

we retrain model *θ* by *Train*( ). 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 *s* 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 distribu- tions of various adversarial example types. Besides, we fur- ther investigate performance impacts of various model set- tings.

## Dataset and Systems

In the experiment, we train MRC models on SQuAD (Ra- jpurkar et al. 2016) training set and its enhanced ver- sion 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 adver- sarial 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 crowd- sourcing. And we select the SQuAD v1.0 as the training dataset

•

AddAnyExtend (AAE): This is our implementation of *AddAny* with extended vocabulary which contains not only question words but also high frequency words, pas- sage words and random common words. The dataset con- tains 2.6k question instances.

AddAnsCtx (AAC): A test set that uses answer context as misleading texts which includes 10k instances. The mis- leading texts are answer sentences with answer tokens re- moved.

•

•

AddNegAns (ANA): A test set that uses negative ex- pressions of fake answers as misleading texts, which in- cludes 5k instances. We create misleading texts by chang- ing 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 differ- ent 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 evalu- ation metric.

## Experiment Settings

In perturbation embedding training phase, we randomly in- sert 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 *λq*, *λp* to be -10 or 10, and set *λc* to be 0.5. And we set *sd* with random length in the middle of each perturbation em- bedding. 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 train- ing 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 *s* 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 ran- domly 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 sam- pling 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 col- lect the local vocabulary for each batch in different ways: high-frequency words; top-10 similar words who have short- est 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 Ta- ble 2. The results indicate that baseline systems are all vul- nerable when facing grammatical or ungrammatical adver- sarial test sets. Although transformer based systems gain su- perior 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 de- sign 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 improve- ments 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 abil- ity. However, our approach still promotes its average perfor- mance by 3.1% in F1 score. The improvement on all adver- sarial 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 im- provement on some of adversarial test sets. Since the train- ing 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 as- sumption 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-

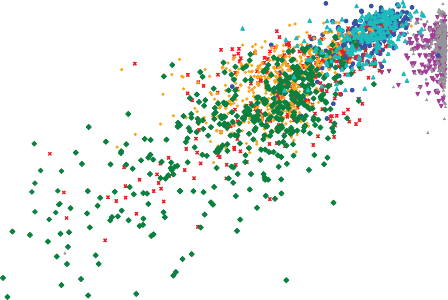


Figure 2: Misleading text distributions of different test sets and training samples.



tensiveness 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 perfor- mance, 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 embed- dings 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.

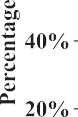


Figure 3: Performance on varying iterations.The solid line denotes the average performance of adversarial test sets. The dotted line denotes the average training loss during perturba- tion embedding training phase.

spectively. Considering that the ceiling performance on de- velopment 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 sim- ilarities 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 ex-

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 predic- tion and the misleading text, we examine the distances be- tween 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 mislead- ing 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 local- ization suggests that misleading texts are able to draw the attention of mode to nearby deceptive “answers”.

**Ablation Study** To investigate the effectiveness of differ- ent 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, *La* plays a much more important role in perturbation embed- ding training. It indicts that “answer” contained misleading texts might be more effective in our approach. And simi- lar observations in the experiment above can also support

|  |  |  |
| --- | --- | --- |
| **Models** QANet BERT*base* | **Generated Misleading Texts**  *exactly player maybe are burnt tend think exactly off best*  *and distinct more combustion ##m no could theory common away* | **Predictions**  burnt 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 *La* 64.1 -8.9

−context loss *Lc* 70.9 -2.1

−similarity regularizer *Rs* 69.7 -3.3

Table 4: Ablation experiment results.

this claim. For similarity regularizer *Rs*, the reduction sug- gests 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 differ- ent 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 ro- bustness becomes a serious problem. Since it is not practical to enumerate and investigate all possible types of adversar- ial examples, improving robustness MRC models by inves- tigating 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 ro- bustness 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 ap- proach can substantially boost MRC model robustness per- formance 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.; Sri- vastava, 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. Explain- ing 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 Inter- national 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 distilla- tion for machine reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2077–2086. Brussels, Belgium: As- sociation for Computational Linguistics.

Iyyer, M.; Wieting, J.; Gimpel, K.; and Zettlemoyer,

L. 2018. Adversarial example generation with syn- tactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

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

Kusner, M. J., and Herna´ndez-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 an- swer networks for machine reading comprehension. In *Pro- ceedings 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. Efﬁ- cient 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 Pa- pers*, 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 repre- sentations for reading comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the As- sociation for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 554–559. New Or- leans, Louisiana: Association for Computational Linguis- tics.

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 ma- chine 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 comprehen- sion. *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 frame- work for language understanding. *ArXiv* abs/1907.12412.

Wang, Y., and Bansal, M. 2018. Robust machine com- prehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan- guage 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 lo- cal convolution with global self-attention for reading com- prehension. 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 Com- puter and Communications Security*, 103–117. ACM.