Enhance Robustness of Language Models Against Variation Attack through Graph Integration

Zi Xiong[1,2,†], Lizhi Qing[1], Yangyang Kang[1,*], Jiawei Liu[2,*], Hongsong Li[1], Changlong Sun[1], Xiaozhong Liu[3], Wei Lu[2]

zi.d.xiong@gmail.com, {yekai.qlz, yangyang.kangyy, hongsong.lhs}@alibaba-inc.com,
{laujames2017, weilu} @whu.edu.cn, changlong.scl@taobao.com, Xliu14@wpi.edu

[1]Institute for Intelligent Computing, Alibaba Group
[2]School of Infromation Management, Wuhan University
[3]Worcester Polytechnic Institute

Abstract

The widespread use of pre-trained language models (PLMs) in natural language processing (NLP) has greatly improved performance outcomes. However, these models' vulnerability to adversarial attacks (e.g., camouflaged hints from drug dealers), particularly in the Chinese language with its rich character diversity/variation and complex structures, hatches vital apprehension. In this study, we propose a novel method, CHinese vAriatioN Graph Enhancement (CHANGE), to increase the robustness of PLMs against character variation attacks in Chinese content. CHANGE presents a novel approach for incorporating a Chinese character variation graph into the PLMs. Through designing different supplementary tasks utilizing the graph structure, CHANGE essentially enhances PLMs' interpretation of adversarially manipulated text. Experiments conducted in a multitude of NLP tasks show that CHANGE outperforms current language models in combating against adversarial attacks and serves as a valuable contribution to robust language model research. These findings contribute to the groundwork on robust language models and highlight the substantial potential of graph-guided pre-training strategies for real-world applications.

Keywords: PLMs, Chinese adversarial attacks, variation graph

1. Introduction

The field of natural language processing (NLP) has seen remarkable advancements in recent years, with pre-trained language models (PLMs) like BERT (Devlin et al., 2018) being one of the most widely adopted tools for various applications, such as language generation (Radford et al., 2018), text classification (Sun et al., 2019), and entity recognition (Andrew and Gao, 2007). PLMs are trained on vast amounts of text data, enabling them to capture patterns and relationships between words and phrases. Yet there is a major limitation of PLMs that their vulnerability to adversarial attacks can lead to the dissemination of false or misleading information (Jiang et al., 2019; Liu et al., 2022; Chen et al., 2024). Adversarial attacks refer to malicious modifications made to the input text, which can cause the language model to make incorrect predictions or misbehave (Ebrahimi et al., 2018). This has become a growing concern, as the use of these models in real-world applications continues to increase. In particular, the Chinese language presents unique challenges in terms of adversarial attacks due to its rich variety of characters (Jiang et al., 2019).

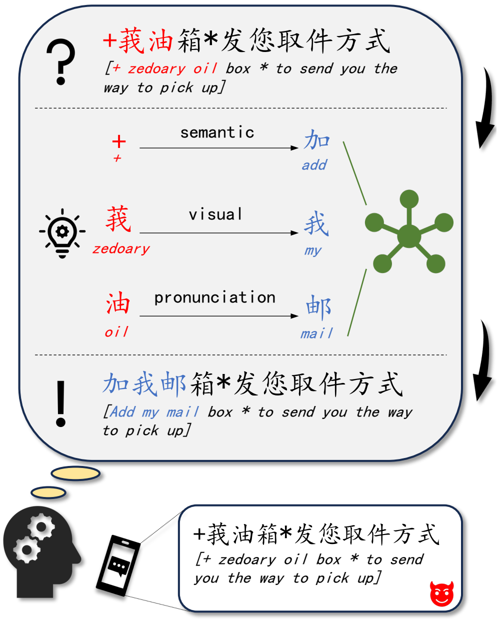
?
+莪油箱*发您取件方式
[+ zedoary oil box * to send you the way to pick up]
semantic
+
+
加
add
visual
莪
zedoary
我
my
pronunciation
油
oil
邮
mail
!
加我邮箱*发您取件方式
[Add my mail box * to send you the way to pick up]
+莪油箱*发您取件方式
[+ zedoary oil box * to send you the way to pick up]

Figure 1: Character Variations via semantic, visual, and pronunciation in Chinese Spam Texts.

Existing methods for mitigating the vulnerability of language models to adversarial attacks primarily focus on fine-tuning models with augmented data (Wang et al., 2020), pre-training models on adversarial examples (Su et al., 2022), employing adversarial training techniques (Si et al., 2021a) or incorporating regularization methods (Liang et al., 2021). The data augmentation strategy heavily depends on the coverage of the augmented data,

[*]Corresponding authors

[†]This work was done when Zi Xiong worked as an intern at Alibaba

which may require an exhaustive exploration of the adversarial space to generate a comprehensive training dataset. This process can be both computationally expensive and time-consuming. Moreover, incorporating adversarial samples may negatively impact the model's performance on clean datasets, as adversarial examples often significantly differ from regular samples.

In this paper, we introduce a novel approach to enhance the robustness of PLMs for the Chinese language. Our proposed method combines multiple adversarial pre-training tasks and incorporates a Chinese Character Variation Knowledge Graph to improve the model's ability to comprehend adversarially attacked natural language text. The multi-task framework facilitates the integration of the knowledge graph into the language model, enabling the model to better capture the linguistic and contextual nuances of the attacked text, thereby enriching the model's textual representations. Our proposed framework, the **CH**hinese v**A**riatio**N** **G**raph **E**nhancing method(**CHANGE**), is illustrated in Figure 2. This PLM-independent method bolsters the model's robustness against poisoned text content and consists of two main components:

(1) The **C**hinese **V**ariation **G**raph **I**ntegration (CVGI) method which employs a variation graph to enhance PLM robustness during the fine-tuning procedure. As depicted in Figure 2, we reconstruct the input sentence following the variation graph.

(2) A Variation Graph Instructed Pre-training method which further trains the PLMs under the guidance of the Variation Graph by appending additional pre-training tasks with the graph to transformer-based PLMs, maximizing the potential of the variation graph.

In comparison to existing techniques, our proposed method offers a more lightweight and cost-efficient solution. By leveraging the graph information as a supplement, the approach maintains convenience while minimally impacting the model's performance on clean datasets. It is primarily attributed to the model's reduced reliance on adversarial data and diminished training on perturbed distributions. Meanwhile, the model places greater emphasis on integrating the graph information, leading to a more streamlined and effective learning process.

Our experimental results demonstrate the superiority of the proposed approach compared to existing PLMs. The results show improved performance in several NLP tasks, as well as increased robustness against adversarial attacks. These findings highlight the potential of multi-task and knowledge graph-augmented language models for practical applications and provide valuable insights for the development of robust language models. In conclusion, our contribution to the field of NLP research is a novel approach for enhancing the robustness of language models against adversarial attacks in the Chinese language, which can be applied to other languages as well.

2. Related Work

Robust Chinese Language Models: The use of PLMs has revolutionized the field of NLP, allowing for significant improvements in a wide range of downstream tasks without the need for training a new model from scratch. One of the first and most influential of these models is BERT (Devlin et al., 2019), which employs a masked language model objective and next sentence prediction task to learn universal language representations. This approach has been further refined by subsequent models such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019). In the realm of Chinese NLP, ChinseBERT (Cui et al., 2021) has made significant strides by incorporating both glyph and pinyin features of Chinese characters into its pre-training process, achieving state-of-the-art results on many Chinese NLP tasks. Despite these successes, the focus of these models has largely been on improving performance on standard texts, with relatively little attention paid to enhancing their robustness.

In the face of real-world adversarial scenarios, many black-box attacks have been developed under the assumption that the adversary only has query access to the target models without any knowledge of the models themselves (Li et al., 2018; Ren et al., 2019; Garg and Ramakrishnan, 2020). To defend against these attacks, countermeasures such as adversarial training and adversarial detection have been proposed to reduce the inherent vulnerability of the model. Adversarial training typically involves retraining the target model by incorporating adversarial texts into the original training dataset, which can be seen as a form of data augmentation (Si et al., 2021b; Ng et al., 2020). Adversarial detection involves checking for spelling errors or adversarial perturbations in the input and restoring it to its benign counterpart (Zhang et al., 2020a; Bao et al., 2021). While these methods have proven effective in the English NLP domain, they are difficult to directly apply to the Chinese domain due to language differences. As a result, many studies have attempted to design specific defenses that take into account the unique properties of Chinese. For example, Wang et al. (2018) and Cheng et al. (2020) improved Chinese-specific spelling check using phonetic and glyph information. Li et al. (2021) proposed AdvGraph, which uses an undirected graph to model the phonetic and glyph adversarial relationships among Chinese characters and improves the robustness of several traditional models. Su et al. (2022) proposed

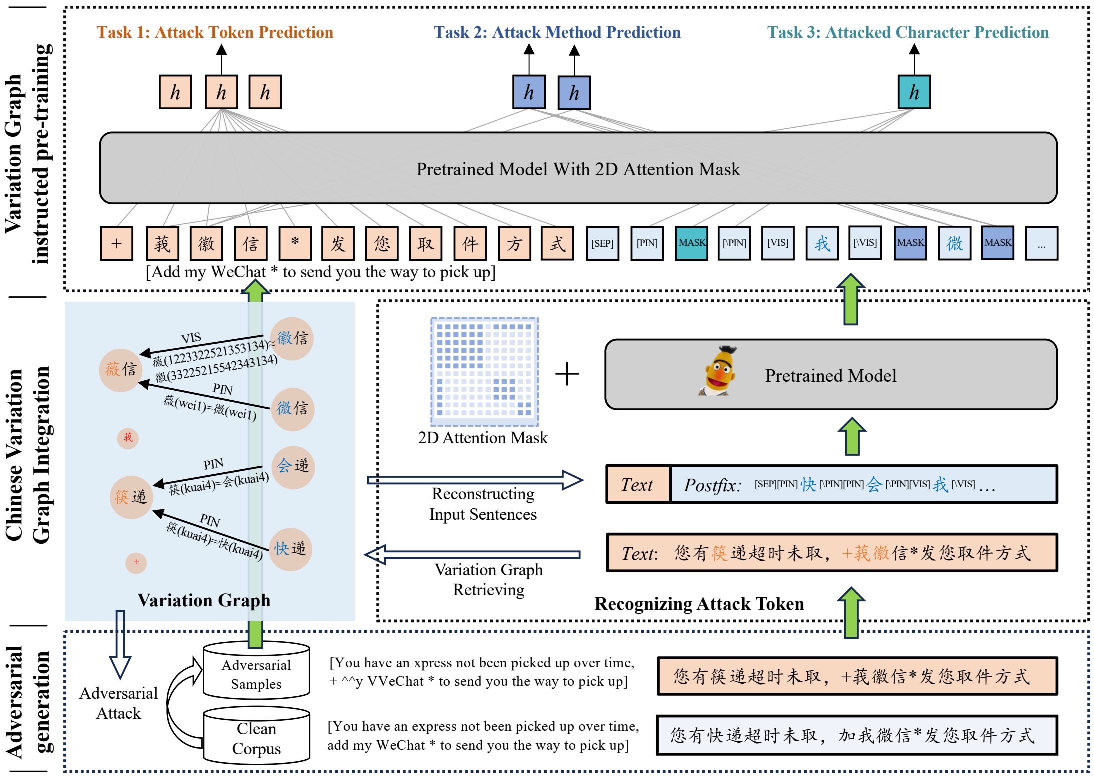
Variation Graph instructed pre-training
Task 1: Attack Token Prediction
Task 2: Attack Method Prediction
Task 3: Attacked Character Prediction
h
h
h
h
h
h
Pretrained Model With 2D Attention Mask
+
莪
徽
信
*
发
您
取
件
方
式
[SEP]
[PIN]
MASK
[\PIN]
[VIS]
我
[\VIS]
MASK
微
MASK
...
[Add my WeChat * to send you the way to pick up]
Chinese Variation Graph Integration
VIS
薇(1223322521353134)≈
徽(3322521554234314)
徽信
薇信
PIN
薇(wei1)=微(wei1)
微信
莪
PIN
筷(kuai4)=会(kuai4)
会递
筷递
PIN
筷(kuai4)=快(kuai4)
快递
+
Variation Graph
2D Attention Mask
+
Pretrained Model
Reconstructing
Input Sentences
Text
Postfix: [SEP][PIN] 快 [\PIN][PIN] 会 [\PIN][VIS] 我 [\VIS] ...
Variation Graph
Retrieving
Text: 您有筷递超时未取，+莪徽信*发您取件方式
Recognizing Attack Token
Adversarial generation
Adversarial
Attack
Adversarial
Samples
Clean
Corpus
[You have an xpress not been picked up over time,
+ ^^y VVeChat * to send you the way to pick up]
您有筷递超时未取，+莪徽信*发您取件方式
[You have an express not been picked up over time,
add my WeChat * to send you the way to pick up]
您有快递超时未取，加我微信*发您取件方式

Figure 2: The overview architecture of the CHANGE method. For the attacked content, the Chinese Variation Graph Integration recognizes the possible variation and reconstruct a postfix attached to the raw input.

RoCBERT to enhance robustness by pre-training the model from scratch with adversarial texts covering combinations of various Chinese-specific attacks, which may not be maintained in downstream tasks.

Knowledge Graph Enhanced PLMs: The enhancement of language understanding in PLMs can be achieved by integrating structured knowledge and linguistic semantics. Recent advancements in Knowledge-Enhanced PLMs (KEPLMs) have uncovered two primary approaches: The first category pertains to the direct incorporation of structured knowledge in PLMs. This category encompasses methods that leverage linguistic semantics inherent in pre-training sentences, and those that utilize entity embeddings derived from structured knowledge. The former technique is exemplified by Lattice-BERT (Lai et al., 2021), which pre-trains a Chinese PLM on a word lattice (Buckman and Neubig, 2018), exploiting multi-granularity inputs to imbue the model with richer semantic understanding. In contrast, the latter technique employs entity embeddings from knowledge encoders, woven into contextual representations to enhance semantic understanding, as exemplified by ERNIE-THU (Zhang et al., 2019). The second category comprises methods that involve reformatting and incorporating knowledge information within the PLM framework, either by encoding textual descriptions of knowledge or transforming knowledge relation triplets into text. This category emphasizes the integration of knowledge descriptions and relation triplets into the textual modality. For instance, KEPLER (Wang et al., 2021) represents the former by jointly encoding pre-training corpora and entity descriptions within a shared semantic space in the same PLM. The latter technique is effectively demonstrated by K-BERT (Liu et al., 2020) and CoLAKE (Sun et al., 2020), which transform relation triplets into text and add the transformed text to training samples, circumventing the need for pre-trained embeddings. This paper posits that amalgamating various forms of knowledge information can substantially bolster the context-aware representation abilities of PLMs.

3. Chinese Variation Graph Integration

We introduce a variation graph that encompasses a comprehensive collection of Chinese character variations utilized in adversarial attacks on Chinese text. This knowledge graph captures variations in phonetics, character shape, and other aspects of Chinese characters, representing the most prevalent attack methods employed in malicious texts such as those found in black-gray industries and fraudulent activities (Jiang et al., 2019; Su et al., 2022; Yang et al., 2021). By incorporating this graph into a language model via a carefully redesigned transformer encoder, our approach enhances the model's resilience to adversarial attacks and preserves its intended functionality. Furthermore, our proposed method is both flexible and lightweight, making it suitable for integration with most transformer-based PLMs.

3.1. Chinese Character Variation Graph

The Chinese Character Variation Graph, includes most of the Chinese character variation approaches. The graph is annotated as $\mathcal{G} = (c_0, r_0, c_1), (c_2, r_1, c_3), ..., (c_i, r_m, c_j)$, in which c_n means attacked character or attack character, r_m means the attack method. The attack methods include the following variation forms:

Pinyin: In the Chinese language, multiple characters may share the same pinyin code representing their pronunciation, making them homophones. Our pinyin variation replaces a Chinese character with one of its homophones. Since a single character may have multiple pronunciations, this can result in homophone variations with different pinyins. We constructed our pinyin variation relation using the pypinyin library[*].

Visual: Chinese characters are logograms, and their visual appearance conveys a significant amount of information. Visual similarity can sometimes be used to confuse readers about the intended meaning of a text. Our visual transformation is based on the SimilarCharacter dataset[†], which calculates Chinese character similarity using the cv2 library[‡].

Character to pinyin: In Chinese text attack scenarios, replacing a character with its pinyin code is a common tactic used to evade word filtering checks. Our character to pinyin variation is also based on the pypinyin library.

These methods can be used to attack Chinese text, making it difficult for text censorship systems to accurately detect and remove illegal or malicious content while human readers can comprehend the content by association, experience, or metaphor. In the attacking scenario, an attack incident may correspond to a triplet in the variation graph. For example, in Figure 2, a node for the character "微" (person) could be connected to a node "薇", with an edge "[PIN]" to indicate the relationship between the character and the method used to alter it, as "微" and "薇" share the same pinyin pronunciation "wei1". In this paper, we refer to this as an attacking path, denoted as $p(n_1, r_1, n_2)$, representing potential paths used to attack text content. The path is used to attack the text to hide the intention of inducing readers to add the attacker's (potentially fraudulent) WeChat account. Our variation graph can be readily integrated into PLMs using our proposed CVGI method, which we will discuss in the following section.

3.2. Variation Graph Integration

In this section, we elaborate on the techniques of our Chinese Variation Graph Integration method, namely **CVGI**. As presented in Figure 2, the method integrates the graph in a transformer-based PLM by reconstructing the input and building a 2D attention mask corresponding to the reconstructed input. The model takes a series of tokens, $(x_1, x_2, ..., x_n)$ as input, and use a stacked transformer encoder layer to generate the contextual representations H_i. Specifically, we use the following three steps to integrate the graph:

Recognizing Attacked Tokens(RAT): Our approach begins by utilizing language model probability to identify tokens that may have been subject to attack. These tokens serve as the targets for the addition of graph information. Given the context $C = (w_1, w_2, ..., w_n)$, the bert output $f_{w_i}(C)$ and bert vocabulary V, the language probability of w_i is:

$$P(w_i|C) = \frac{\exp\left(f_{w_i}(C)\right)}{\sum_{w_j \in V} \exp\left(f_{w_j}(C)\right)}$$

In an input sentence, we can rank the tokens according to their probability and take the lowest k% tokens as possibly potentially attacked tokens W^a. This approach is commonly used in Chinese Spelling Correction problems, where language model output probability is a recognized metric for identifying incorrect words or attacked characters within a sentence. Then, we can leverage the Variation Graph to retrieve possible candidate original words W^c for the attacked tokens $W^a = \{w_1^a, w_2^a, ..., w_n^a\}$ in the graph and the corresponding attacking paths $P = \{p_1, p_2, ..., p_n\}$.

Reconstructing Input Sentences: Based on the original input C and the attacking paths P we get, we will add a postfix to C which is generated from P. As shown in Figure 3 (a), an attacking path $p(微, pinyin, 薇)$ will generate an span formated as "[PIN]微[/PIN]" and appended to C. Specifically, the [PIN]([/PIN]) token corresponds to *pinyin variation*, while [VIS]([/VIS]) corresponds to *vision variation*, and [CTY]([/CTY]) corresponds to *character to pinyin variations*. Note that each w^c in the possi-

[*] https://github.com/mozillazg/python-pinyin

[†] https://github.com/contr4l/SimilarCharacter

[‡] https://pypi.org/project/opencv-python/

bly attacked tokens W^c may correspond to several attacking paths p, which may result in a very long reconstructed postfix and bring much noise to the original input C. To reduce the computational complexity and weaken the noise, we further use a 2D mask to eliminate the cross attention between the reconstructed span from attacking paths p and most of the original input C, except for only the attack word w_a in p.

Adversarial 2D attention map: As shown in Figure 3 (b), our model use a 2D attention mask instead of cross attention mask. For a reconstructed input, the part of the original sentence will use full cross-attention. For the postfix part, "[VIS]徽[/VIS]" for example, would only have attention with the attacked token "微". The 2D attention mask is calculated during the tokenization phase of the model input. The 2D attention mask is fluently combined with the reconstructed input, enabling the PLMs to gather the original word information to attacked tokens through the attacking path. PLMs have the potential to tell the right attacking path from attention and pick the right original word and inject its information into the original content.

Our CVGI method has broad applicability to various Chinese NLU tasks and can enhance the performance of most transformer-based PLMs. The reconstruction method is adaptable to various input formats, while the 2D mask is compatible with most of the transformer architectures and the CVGI method does not depend on a specific PLM. The effectiveness of our CVGI method is demonstrated in section 5.3. Since the objective of fine-tuning does not directly guide the PLMs and the integrated component to construct the attacking path, only fine-tuning is insufficient for the CVGI method to accurately recognize the attacked token and the attacking path from the Variation Graph. We further designed Variation Graph instructed pretrain tasks to help a target PLM to learn to identify the vital path of the graph for enhancing the robustness against attacks in the next section.

4. Variation Graph Instructed Pre-training

In this section, we present the details of how we further improve the effectiveness of CVGI by designing Variation Graph instructed pre-training tasks. The key point to strengthen the ability of CVGI is to improve the effectiveness of RAT and the ability to restore the attacking paths by PLMs. It's important to design tasks to help the PLMs learn to adjust the 2D attention mask to the reconstructed text input so that it can better integrate the adversarial information to the attacked tokens. The pre-training is designed to be light weighted so that it can be relatively easy and less costly to apply to ordinary PLMs. In our settings, it only uses a 14 GB corpus from Chinese Wikipedia, Baidu Baike, and THUCTC as the pre-training data.

4.1. pre-training tasks

The pre-training task is designed based on the CVGI paradigm to enhance the RAT. We develop a language probability modeling task using Masked Language Modeling (MLM) as the training objectives and create several Variation Graph instructed pre-training tasks. In contrast to common pre-training tasks, our tasks are conducted on attacked samples, which are constructed from a clean corpus using our Variation Graph. During the attacking process, we generate samples annotated with attacking paths. Next, we reconstruct input examples and produce 2D attention maps similar to CVGI but with slight differences for each task. Specifically, for the RAT enhancing task, we generate an attacked sample with annotations indicating which tokens have been attacked. As we know the ground truth attacking path, we design the reconstructed samples to contain only the true attacking path (p^T) and can additionally sample false attacking paths (p^F) to construct postfixes and append to the input. For MLM tasks, we mask tokens in the reconstructed samples following various strategies and conduct MLM training. We design 3 tasks: Attack Token Prediction, Attack Method Prediction, and Attacked Character Prediction, which includes the following:

Attack Token Prediction Attack Token Prediction (ATP) task is to predict the attack characters based on the raw input, corresponding to the RAT task. Suppose the token $y_i \in 0, 1$ is labeled as attacked (1) or normal (0), its LM probability is p_i. The loss of ATP is:

$$\mathcal{L}_{\text{ATP}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + (1 - y_i)\log(1 - p_i)$$

Note that the ATP task can be performed on the target PLM or a certain PLM. It is aimed to enhance the ability of the RAT task by constructing a rich RAT annotated train set using Variation Graph.

Attack Method Prediction Attack Method Prediction (AMP) predicts the attack method by predicting the masked attack method in the reconstructed input. In the reconstruction, 15% of the method tokens are masked if there are more than N attacking paths in the reconstructed sample, else one random method will be masked. If the $\hat{y}^M_{i,j}$ stands for the one-hot code of the ith method token, m stands for the one-hot code of masked method tokens, $\hat{y}^M_{i,j}$ stands for the probability. The MLM loss of AMP is:

$$\mathcal{L}_{\text{AMP}} = -\sum_{j=1}^{m} y^M_j \log(\hat{y}^M_j)$$

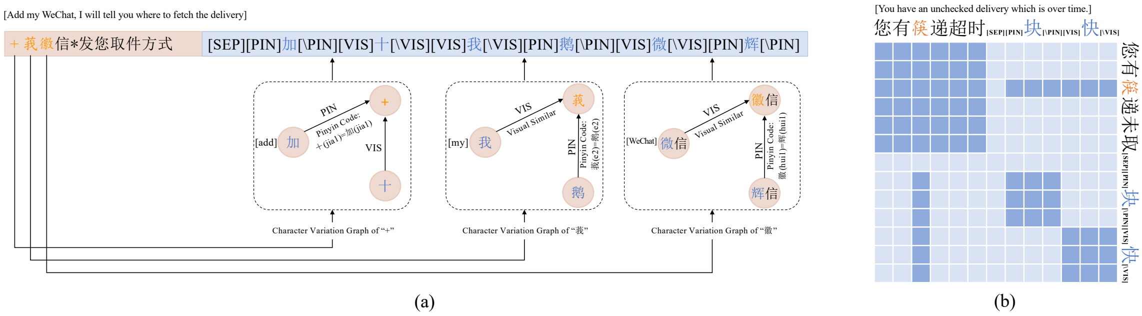
[Add my WeChat, I will tell you where to fetch the delivery]
+莪徽信*发您取件方式
[SEP][PIN]加[\PIN][VIS]十[\VIS][VIS]我[\VIS][PIN]鹅[\PIN][VIS]微[\VIS][PIN]辉[\PIN]
[add]
加
+
十
PIN
Pinyin Code:
+(jia1)=加(jia1)
VIS
Character Variation Graph of “+”
[my]
我
莪
鹅
VIS
Visual Similar
PIN
Pinyin Code:
莪(e2)=鹅(e2)
Character Variation Graph of “莪”
[WeChat]
微信
徽信
辉信
VIS
Visual Similar
PIN
Pinyin Code:
徽(hui1)=辉(hui1)
Character Variation Graph of “徽”
(a)
[You have an unchecked delivery which is over time.]
您有筷递超时[SEP][PIN]块[\PIN][VIS]快[\VIS]
您有筷递未取[SEP][PIN]块[\PIN][VIS]快[\VIS]
(b)

Figure 3: (a): An example of reconstruction. In the Variation Graph, the red “+” symbol has two variations: “加” through pinyin variation and “十” through visual variation. Similarly, the red “莪” character possesses two variations in the Variation Graph: “我” and “窝”, both derived from pinyin variation. The red “徽” character features two variations in the graph: “薇” through pinyin variation and “微” through visual variation. (b): An example of adversarial 2d attention map. In the whole reconstructed sentence, the raw text segment “您有筷递超时未取” employs full cross-attention. The identified attacked character “筷” exclusively attends to its variations in the postfix segment. And the candidate original characters, “[PIN]块[/PIN]” in example, only have attention with the attacked character “快”.

The AMP task aims to predict the relationship based on the head entity and the contextual environment.

Attacked Character Prediction Attacked Character Prediction (ACP) predicts the attack character by predicting the masked attacked character in the reconstructed input. In the reconstruction, 15% of the attacked tokens are masked if there are more than 6 attacking paths in the reconstructed sample, else 1 random attacked token will be masked. If the $y^m_{i,j}$ stands for the one-hot for the ith attacked character token, n stands for the num of masked attacked character tokens, $\hat{y^c}_{i,j}$ stands for the probability. The MLM loss of ACP is:

$$\mathcal{L}_{\text{ACP}} = -\sum_{j=1}^{n} y^c_j \log(\hat{y^c}_j)$$

The ACP task focuses on predicting the tail entity based on the head entity, the relation, and the contextual environment.

5. Experiment

We conducted an evaluation of our proposed method on three distinct datasets and compared its performance against several existing pre-trained language models. Our results demonstrate that our model consistently outperforms the baseline models across all datasets, providing strong evidence for the effectiveness of our approach.

5.1. Experiment Setup

Training Detail. For pre-training, we utilized a combined corpus comprising Chinese Wikipedia, Baidu Baike, and THUCTC[§]. Our model was trained for 100,000 steps with a batch size of 64, a learning rate of 1e-4, and a warm-up rate of 2,000 steps. The corpus contains 13GB of pure text. The training was conducted on 8 Tesla V100 GPUs using the DeepSpeed framework.

Baseline Models. We tested our approach based on several SOTA models: (1) Chinese-bert-wwm (Cui et al., 2019), (2) MacBERT (Cui et al., 2020), (3) RocBERT (Su et al., 2022). Chinese-bert-wwm uses the Whole Word Masking pre-training strategy to enhance BERT model especially for the Chinese language. MacBERT applies the MLM-As-Correlation pre-training strategy as well as the sentence-order prediction task. RocBERT is a pre-trained Chinese BERT model that is robust to various forms of adversarial attacks such as word perturbation, synonyms, and typos. It is pre-trained with a contrastive learning objective that maximizes the label consistency under different synthesized adversarial examples.

Tasks. Our proposed method is tested on two standard Chinese NLU tasks and a real-world toxic detection task, which are: (1) TNews[¶]: news title classification with 50k training data, (2) AFQMC[‖]: question matching with 34k training data, (3) Message: The Message Toxic dataset consists of real-world data collected from a real-world application that receives short message notifications and detects messages sent by frauds or black-grey industry practitioners. We manually annotated 15,000

[§] http://thuctc.thunlp.org/

[¶] https://www.cluebenchmarks.com/tnews_public

[‖] https://www.cluebenchmarks.com/afqmc_public

Base Model	Method	TNews		
		Clean	Attacked	Argot
Chinese-bert-wwm	BASE	54.04	50.68	49.63
	CVGI	54.47	51.84	50.94
	CHANGE	54.28	**53.91**	**52.87**
MacBERT	BASE	**56.12**	52.34	51.54
	CVGI	55.33	53.49	52.62
	CHANGE	55.79	**55.37**	**54.22**
RocBERT	BASE	56.22	54.88	51.83
	CVGI	56.37	55.74	52.84
	CHANGE	**57.09**	**56.94**	**54.22**

Table 1: Experimental results on TNews. Bold shows the best performance of method variants with the same base model.

Base Model	Method	Afqmc		
		Clean	Attacked	Argot
Chinese-bert-wwm	BASE	68.91	64.38	59.43
	CVGI	68.58	65.83	60.84
	CHANGE	69.46	**67.96**	**62.71**
MacBERT	BASE	**70.83**	66.75	61.50
	CVGI	69.98	67.67	62.40
	CHANGE	70.69	**68.99**	**64.20**
RocBERT	BASE	70.41	67.11	62.07
	CVGI	**70.95**	67.68	62.18
	CHANGE	69.85	**69.05**	**63.03**

Table 2: Experimental results on Afqmc. Bold shows the best performance of method variants with the same base model.

Base Model	Method	Message		
		$F1_{micro}$	Recall	Precision
Chinese-bert-wwm	BASE	82.76	91.28	75.70
	CVGI	84.06	92.59	76.97
	CHANGE	**85.93**	**94.66**	**78.67**
MacBERT	BASE	84.74	93.35	77.59
	CVGI	85.69	94.33	78.50
	CHANGE	**87.01**	95.30	**80.05**
RocBERT	BASE	85.94	95.81	77.92
	CVGI	86.81	96.53	78.87
	CHANGE	**87.61**	**97.37**	**79.63**

Table 3: Experimental results on Message. Bold shows the best performance of method variants with the same base model.

user inputs and identified 2,000 toxic contents (positive), of which 90% were in adversarial forms. We then randomly sampled 2,000 negative texts and split the entire dataset into training, development, and testing sets with an 8:1:1 ratio.

Settings. We conducted the experiment under several circumstances: (1) Clean: the uncontaminated dataset, (2) Attacked: based on the clean dataset, use the variation graph to attack the data. (3) Argot (Zhang et al., 2020b): Use the Argot algorithm which is designed for the Chinese language to attack the original data. (4) Toxic: The original toxic detection dataset collected from a real-world application. The test on the contaminated dataset proves the robustness of our method, while the test on the clean datasets proves the generativity of our method. The results are shown in Table 3.

5.2. Experiment Result

In Tnews, AFQMC and Message datasets, we test the NLU ability of our models under both attacked and clean circumstances, measured by F1-macro. For every task, we report the model performance under 2 adversarial algorithms, Att and Argot. We also report the performance of all base-version models for a fair comparison. The results are presented in Table 1, Table 2, and Table 3. On average, the performance of PLMs using our CHANGE framework negligibly decreased by only 0.05% on clean data, which is approximately equivalent to no significant change. In contrast, on attacked datasets, the performance of CHANGE improved by 1.21% (Att) and 1.03% (Argot), respectively. In the Att and Argot test settings, CHANGE consistently outperforms all baseline models, with particularly notable improvements over MacBERT and Chinese-bert-wwm. In the clean dataset, the performance of methods utilizing CHANGE is comparable to that of the baseline models. When employing CHANGE methods, the performance in attacked test sets more closely approximates that in the clean set than when using baseline models. Of all the baselines, RocBERT performs closest to $\text{RocBERT}_{\text{CHANGE}}$ under the two attacks, likely due to its multimodal input and its use of adversarial samples during training. Both BERT and MacBERT show significant improvement with CHANGE pre-training and integration. After being enhanced by CHANGE, their robustness performance surpasses that of RocBERT but still falls behind $\text{RocBERT}_{\text{CHANGE}}$. Since BERT and MacBERT were not specifically trained to handle adversarial circumstances, the improvement brought by CHANGE is particularly notable.

5.3. Ablation Study of CHANGE

We conducted a series of ablation studies to understand the functionality of different components in CHANGE. In the experiments, The models then tested on three datasets: Tnews, Afqmc and Message. For Tnews and Afqmc, there are three subdatasets: Clean, Att, and Argot, and the measurement is F1-score. For the message dataset, the measurement is F1-score, precision and recall. All the fine-tuning is conducted with the same base architecture for a maximum of 10 epochs and an early stop strategy on training text. We devised several

Base Model	Method	TNews			Afqmc			Message
		Clean	Attacked	Argot	Clean	Attacked	Argot	
Chinese-bert-wwm	CHANGE	54.28	**53.91**	**52.87**	69.46	**67.96**	**62.71**	**85.93**
	↪w/o ATP	**54.51**	52.87	51.99	**69.61**	66.99	61.88	85.07
	↪w/o AMP	53.89	52.36	51.61	68.83	66.35	61.27	84.53
	↪w/o ACP	54.12	52.62	51.69	69.19	66.47	61.56	84.74
MacBERT	CHANGE	55.79	**55.37**	**54.22**	70.69	**68.99**	**64.20**	**87.01**
	↪w/o ATP	55.48	54.92	53.97	70.15	68.79	63.71	86.90
	↪w/o AMP	55.62	53.94	53.14	70.30	67.97	62.89	86.13
	↪w/o ACP	55.68	54.19	53.29	70.58	68.19	63.08	86.27
RocBERT	CHANGE	**57.09**	**56.94**	**54.22**	69.85	**69.05**	**63.03**	**87.61**
	↪w/o ATP	56.45	55.32	53.06	69.89	67.80	62.24	86.44
	↪w/o AMP	56.11	56.34	53.56	69.34	68.36	62.45	87.42
	↪w/o ACP	56.44	56.43	53.27	69.72	68.25	62.64	87.29

Table 4: Ablation experimental results with different base models and different pre-training strategies on TNews, Afqmc and Message. Bold shows the best performance of method variants with the same base model.

Base Model	Method	TNews		
		Clean	Attacked	Argot
RocBERT	BASE	55.92	54.73	51.96
	CVGE	56.91	56.63	54.32
ChatGPT	gpt-3.5-turbo-0301	43.68	40.02	37.45

Table 5: Experimental results of performance(%) comparison with ChatGPT on TNews test sets.

variants of CHANGE, each excluding specific components of the strategy: w/o ATP, w/o AMP, and w/o ACP represent variants that exclude the ATP, AMP, and ACP tasks from CHANGE during pre-training. The results, as illustrated in Table 4, reveal that the inclusion of ATP, AMP, and ACP tasks led to average performance improvements of 0.53%, 0.47%, and 0.61% for CHANGE, respectively.

5.4. Robustness Analysis

As illustrated in Table 3, our approach demonstrates effectiveness on both clean and attacked datasets. In clean datasets, our method maintains the original performance of the models, while significantly improving their robustness in attacked datasets. The integration of variation graph-instructed pre-training tasks with our approach results in a noticeable improvement across all attacked experiment settings. In the best-case scenario, the performance improved by 4.3% in an attacked dataset, reducing the gap between clean and attacked datasets from 5.2% to 0.9%. This indicates that our method, CHANGE, can effectively mitigate the effects of attacks on PLMs without sacrificing their performance on normal data.

Moreover, we also conduct an experiment to evaluate the performance of ChatGPT on TNews. Due to space limitations, experiment results are shown in Table 5.

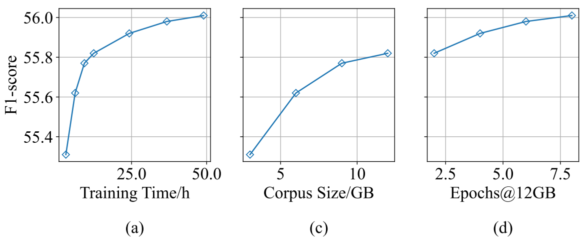

Figure 4: The impact of corpus size and training costs on the f1-score performance of CHANGE-enhanced PLM on the TNews Dataset.

5.5. Costs of pre-training

Our method follows a plug-and-play design to flexibly and generically enhance the robustness of PLMs. As such, it is essential that our model incurs low training costs. We conducted experiments on the TNews dataset using the Chinese-bert-wwm model. In one set of experiments (Figure 4 (b)), we trained CHANGE using 3GB, 6GB, 9GB, and 12GB of corpus data (evenly split across multiple sources) for a fixed 2 epochs. In another set of experiments (Figure 4 (c)), we trained on 12GB of corpus data for 2, 4, 6, and 8 epochs. The training time costs and the effectiveness of fine-tuning are shown in Figure 4 (a). As can be seen, using 9GB of corpus data yields satisfactory results. Training on 12GB of corpus data for 10 hours on a bert-based pretrain language model is sufficient to achieve strong robustness enhancement. These experiments demonstrate the plug-and-play nature and usability of our CHANGE method.

6. Limitation

In this section, we discuss the limitations of our proposed method. Although our approach demonstrates promising results in terms of performance and robustness, several challenges need to be addressed in future research: (1) Domain Adaptability: While our method exhibits improved performance on various NLP tasks, these tasks may not cover the complete range of domains that pre-trained language models might encounter in real-world applications. Therefore, the adaptability of the proposed approach to the adversarial robustness across different domains warrants further investigation. (2) Limited Adversarial Defense Scope: Our method enhances the adversarial robustness of pre-trained language models on several NLP tasks; however, potential attack strategies not covered in the experiments might exist. To comprehensively assess the robustness of our approach, it is essential to validate it under a broader range of attack scenarios. (3) Scalability: Our study focuses on the robustness of pre-trained language models in the Chinese context. However, due to structural and linguistic differences between languages, directly applying the proposed method to other languages may pose challenges. Consequently, appropriate adjustments and validation are required before extending our approach to other languages. (4) Computational Cost: Our proposed method necessitates constructing the Variation Graph and employing multi-task learning during the pre-training and fine-tuning processes. This might lead to increased computational costs, limiting the applicability of our method in resource-constrained environments.

In conclusion, despite our achievements made in enhancing the robustness of pre-trained language models, several limitations and challenges remain to be addressed. Investigating these issues will contribute to a better understanding and utilization of knowledge graphs and multi-task learning methods, ultimately improving the robustness of language models in practical applications.

7. Conclusion

In this paper, we introduce CHANGE, a universal method for integrating the Chinese character variation graph into Chinese language models to enhance their robust representation. Our approach involves designing a method for injecting the graph into transformer-based PLMs during the fine-tuning phase and enhancing adversarial graph injection during PLM pre-training. Experimental results demonstrate that all PLMs enhanced by our CHANGE outperform their respective baselines in robustness tests. Further analysis reveals that CHANGE can effectively capture various paths of common Chinese attacks. For future work, we plan to extend CHANGE to encompass additional types of attacks beyond character variation (e.g., substitution) and apply CHANGE to a broader range of downstream tasks.

8. Acknowledgements

This work was supported by National Science and Technology Major Project (2022ZD0116204) and Alibaba Group through Alibaba Innovation Research Program.

9. Bibliographical References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of l1-regularized log-linear models. In *International Conference on Machine Learning*.

Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3248–3258, Online. Association for Computational Linguistics.

Jacob Buckman and Graham Neubig. 2018. Neural lattice language models. *Transactions of the Association for Computational Linguistics*, 6:529–541.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Zhuo Chen, Jiawei Liu, and Haotan Liu. 2024. Research on the reliability and fairness of opinion retrieval in public topics. *2024 Network and Distributed System Security (NDSS) workshop on AI Systems with Confidential Computing*.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online. Association for Computational Linguistics.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2021. ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. 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 2063–2073, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

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

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Zhuoren Jiang, Zhe Gao, Guoxiu He, Yangyang Kang, Changlong Sun, Qiong Zhang, Luo Si, and Xiaozhong Liu. 2019. Detect camouflaged spam content via StoneSkipping: Graph and text joint embedding for Chinese character variation representation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6187–6196, Hong Kong, China. Association for Computational Linguistics.

Yuxuan Lai, Yijia Liu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2021. Lattice-bert: Leveraging multi-granularity representations in chinese pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1716–1731.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Jinfeng Li, Tianyu Du, Xiangyu Liu, Rong Zhang, Hui Xue, and Shouling Ji. 2021. Enhancing model robustness by incorporating adversarial knowledge into semantic representation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7708–7712. IEEE.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *CoRR*, abs/1812.05271.

Xiaobo* Liang, Lijun* Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *NeurIPS*.

Jiawei Liu, Yangyang Kang, Di Tang, Kaisong Song, Changlong Sun, Xiaofeng Wang, Wei Lu, and Xiaozhong Liu. 2022. Order-disorder: Imitation adversarial attacks for black-box neural ranking models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2025–2039.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI*

Conference on Artificial Intelligence, volume 34, pages 2901–2908.

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 arXiv:1907.11692*.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021a. Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1569–1576.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021b. Better robustness by more coverage: Adversarial and mixup data augmentation for robust finetuning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1569–1576, Online. Association for Computational Linguistics.

Hui Su, Weiwei Shi, Xiaoyu Shen, Zhou Xiao, Tuo Ji, Jiarui Fang, and Jie Zhou. 2022. Rocbert: Robust chinese bert with multimodal contrastive pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 921–931.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18*, pages 194–206. Springer.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CAT-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten De Rijke, Yixing Fan, and Xueqi Cheng. 2023. Prada: Practical black-box adversarial attacks against neural ranking models. *ACM Transactions on Information Systems*, 41(4):1–27.

Ronghai Yang, Xianbo Wang, Cheng Chi, Dawei Wang, Jiawei He, Siming Pang, and Wing Cheong Lau. 2021. Scalable detection of promotional website defacements in black hat seo campaigns. In *USENIX Security Symposium*, pages 3703–3720.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020a. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 882–890, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.

Zihan Zhang, Mingxuan Liu, Chao Zhang, Yiming Zhang, Zhou Li, Qi Li, Haixin Duan, and Donghong Sun. 2020b. Argot: Generating adversarial readable chinese texts. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2533–2539. International Joint Conferences on Artificial Intelligence Organization. Main track.