TextShield: Robust Text Classification Based on Multimodal Embedding and Neural Machine Translation

Jinfeng Li Tianyu Du Shouling Ji Rong Zhang Quan Lu Min Yang Ting Wang

USENIX Security 2020

Deep Learning For Natural Language Processing

Fake News Detection

Email Anti-spam

Hate Speech Detection

Security-sensitive NLP Tasks

Pornography detection

Political Content Detection

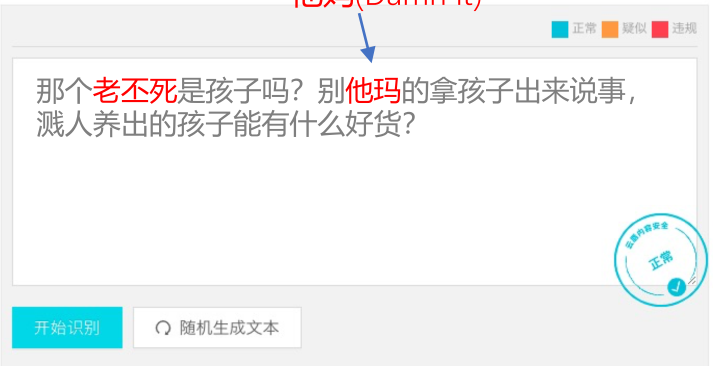
Real-world Applications For Toxic Content Detection

- It was reported that major social media platforms (e.g., Twitter and Facebook) were all criticized for not doing enough to curb the diffusion of toxic content and under pressure to cleanse their platforms.

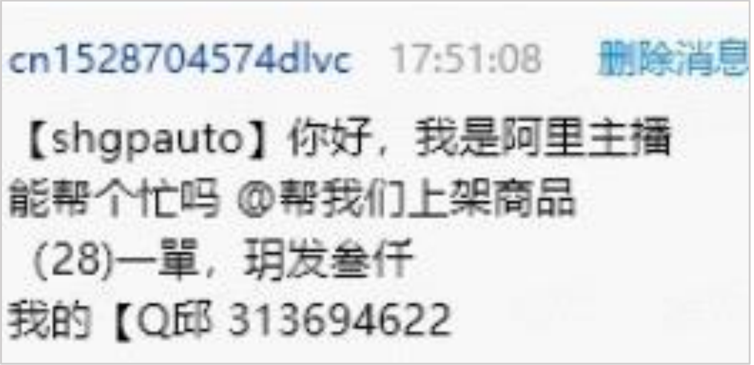
Background： Real-world Adversarial Scenario

Real-word Adversarial Texts

他妈(Damn it)



(a) Abusive content



(b) spam message

(b) Pornographic content

- Chinese-based toxic content censorship system is suffering from the vulnerability to adversarial texts manually crafted by real-world malicious netizens.
- Manually reviewing these adversarial texts is usually time-consuming and laborious.
- In the arm race of adversarial attacks and defenses, existing defenses are obviously at a disadvantage.

Preliminaries

Related Works: Attacks and Defenses

➢ Adversarial Attacks for Text

- A plenty of attacks have been proposed in recent years. [Papernot *et al*., MILCOM' 18, Ebrahimi *et al.* ,NAACL' 18, Gao *et al.* SPW' 18, Li *et al.,* NDSS' 19]

➢ Defenses against Adversarial Text

☑ Adversarial Training

- Retrain the machine comprehension model with diversified adversarial training. [Wang *et al.,* NAACL'18]
- Improve the robustness of the character-level NMT models by adversarial training. [Ebrahimi *et.al.,* COLING'18]

☑ Spelling Correction

- Gao et al. used the Python autocorrector to mitigate DeepWordBug and significantly improved the model accuracy under adversarial setting. [Gao *el at.,* SPW'18]
- Li et al. applied a context-aware spelling correction service to defend against TextBugger. [Li *et al.,* NDSS'19]

Related Works: Attacks and Defenses

Unique Property of Chinese-based Adversarial Attacks

- Chinese is a compositional language in which each text consists of a set of characters that are individually meaningful and the modification of a single character may drastically alter the semantics of the text, **making Chinese-based NLP models inherently more vulnerable to adversarial attacks**.
- There is an extremely large character space (i.e., more than 50,000 characters) in Chinese in which each character may be perturbed by various strategies (e.g., glyph-based and phonetic-based strategies), **making the adversarial perturbations more sparse.**
- Most of the Chinese adversarial texts are manually crafted by real-world malicious netizens, **which are more diverse** due to the various word variation strategies adopted by different netizens.

Related Works: Attacks and Defenses

Challenges of Extending existing defenses to Chinese NLP Tasks

☑ **Adversarial Training**

- ➢ There currently exists no automatic attack for generating Chinese adversarial texts while the manual collection of user generated obfuscated texts for adversarial training is often laborious and costly.
- ➢ The unique sparseness and dynamicity of Chinese adversarial perturbations also weaken the efficacy of adversarial training.

☑ **Spelling Correction**

- ➢ These is no word boundary in Chinese writing system while variant characters can only be determined at the word-level, and hence it is more difficult to perform spelling correction in Chinese.
- ➢ Spelling correction heavily relies on the semantic context of the input texts, which can also be ruined by adversarial perturbations.
- ➢ The diversity and dynamicity of Chinese adversarial perturbations also challenge such approaches.

TextShield

Problem Definition and Threat Model

Threat Model

Given a classification model $F(\cdot)$, an attacker who has query access to the classification confidence returned by this model, aims to generate a Chinese adversarial text $\boldsymbol{x_{adv}}$ from its benign counterpart $\boldsymbol{x} \in \mathcal{X}$ whose label is $y \in \mathcal{Y}$, such that $F(\boldsymbol{x_{adv}}) = t \ (t \neq y)$.

Problem Definition

We aim to defend such attacks by leveraging neural machine translation (NMT) to restore $\boldsymbol{x_{adv}}$, and universally improving the robustness of $F(\cdot)$ by multimodal embedding. Formally, our defense is defined as

$$\mathcal{F}\left(E_{sgp}\left(\underset{\boldsymbol{x}^* \in \mathcal{X}}{\arg\max}\, p(\boldsymbol{x}^* | \boldsymbol{x}_{adv}; \theta)\right)\right) = y,$$

where $E_{sgp}(\cdot)$ is the multimodal embedding function, $\boldsymbol{x}^*$ is a candidate text corrected from $\boldsymbol{x_{adv}}$, $p(\boldsymbol{x}^* | \boldsymbol{x_{adv}}; \boldsymbol{\theta})$ is the probability of outputting $\boldsymbol{x}^*$ given $\boldsymbol{x_{adv}}$, and $\boldsymbol{\theta}$ is the parameters of the NMT model.

Framework For TextShield

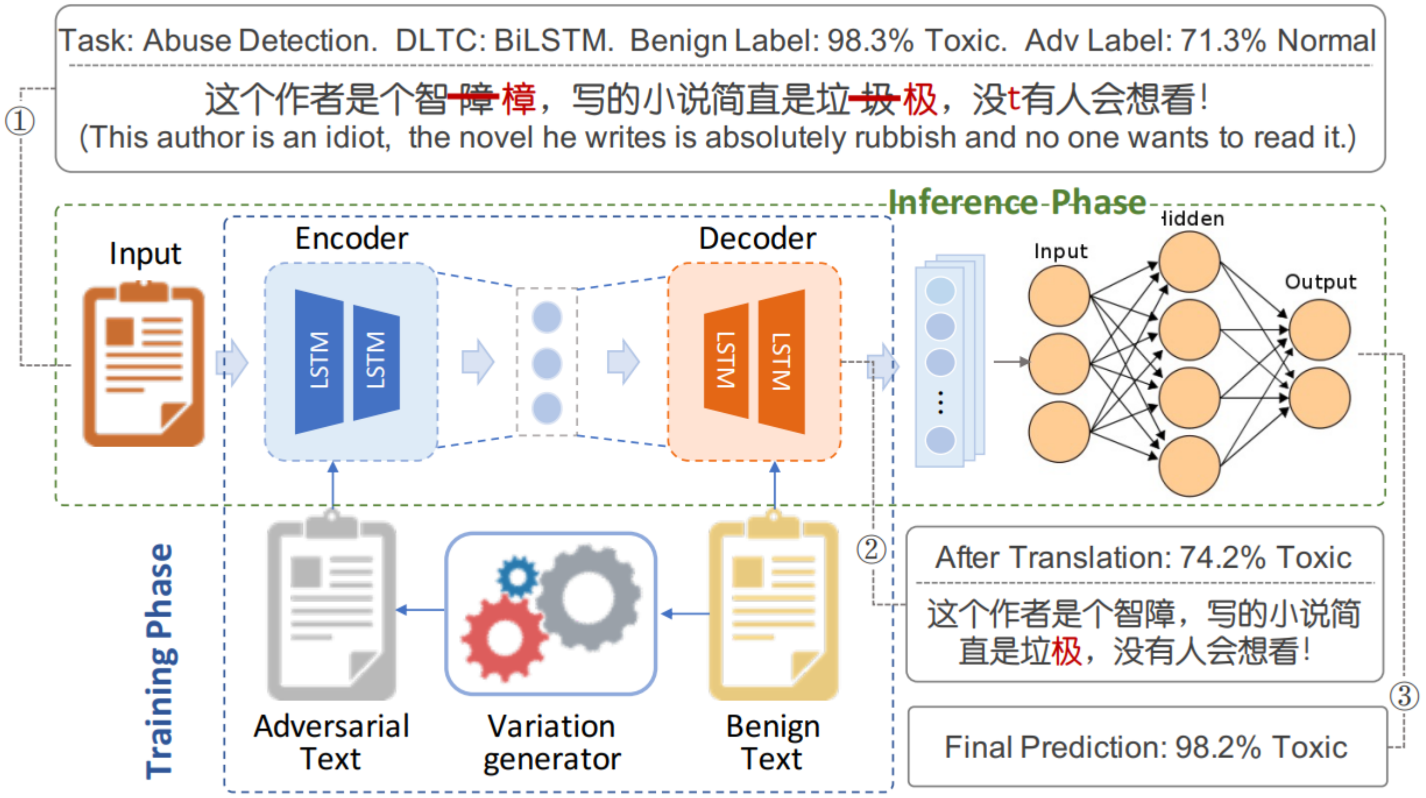


Figure 1: The framework of TEXTSHIELD.

First Stage: Adversarial Machine Translation

➢ Model Design

- Use LSTM to implement the encoder and decoder.
- Use Bahdanau's attention mechanism to align the source input and the target translation.

➢ Model Training

- Construct a large adversarial parallel corpus $\mathcal{D}_{adv}$
- Minimize the negative log probability on $\mathcal{D}_{adv}$

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{D}_{adv}|} \sum_{(\boldsymbol{x}_{adv}, \boldsymbol{x}_{ori}) \in \mathcal{D}_{adv}} \log p(\boldsymbol{x}_{ori} | \boldsymbol{x}_{adv}).$$

- Use teacher forcing technique to avoid the error being amplified

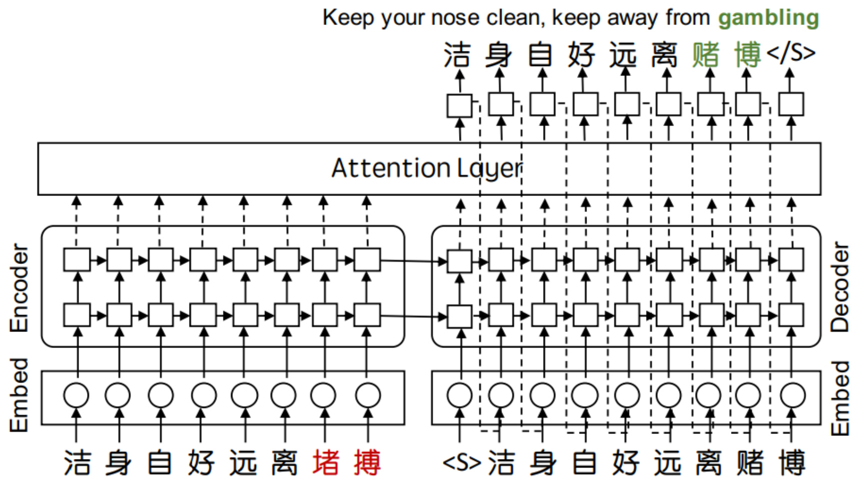


Figure 2: Architecture of adversarial NMT model.

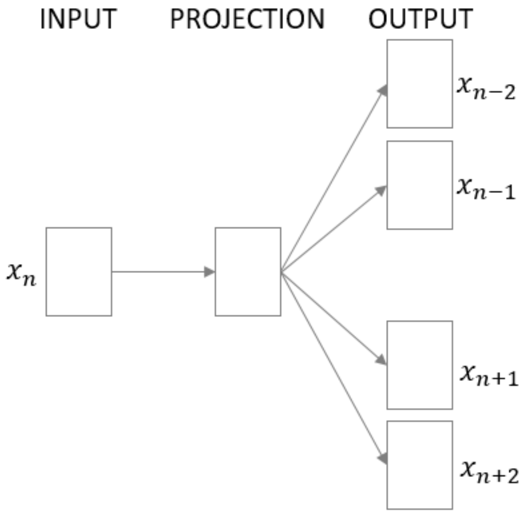
➢ Adversarial Correction

- Reconstruct the original text from $\boldsymbol{x}_{adv}$ by maximizing $\boldsymbol{x}^*_{opt} = \arg\max_{\boldsymbol{x}^* \in X} p(\boldsymbol{x}^* | \boldsymbol{x}_{adv}; \theta).$
- Apply beam-search for improving the translation performance

Second Stage: Multimodal Embedding

➢ Semantic Embedding

- Use word2vec scheme to map each character to a semantic embedding $V^{(S)}$

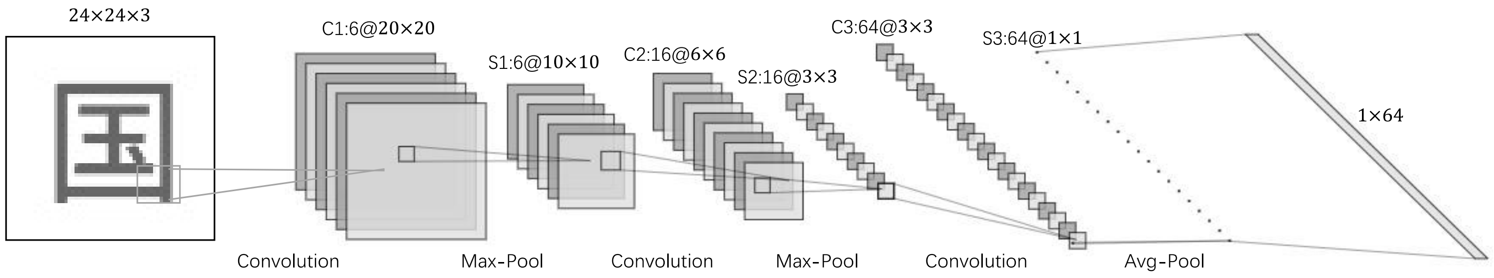


➢ Phonetic Embedding

- Aim to handle the phonetic perturbation, e.g, 色情 (porn) --> 涩情 or *seqing*
- Convert each character into its Pinyin form.
- Use word2vec to learn embedding vector $V^{(P)}$ over the Pinyin form.

➢ Glyph Embedding

- Aim to handle the glyph-based perturbation, e.g, 赌博 (gamble) --> 堵搏
- Convert each character into an image and use g-CNN to learn the glyph embedding vector $V^{(G)}$.



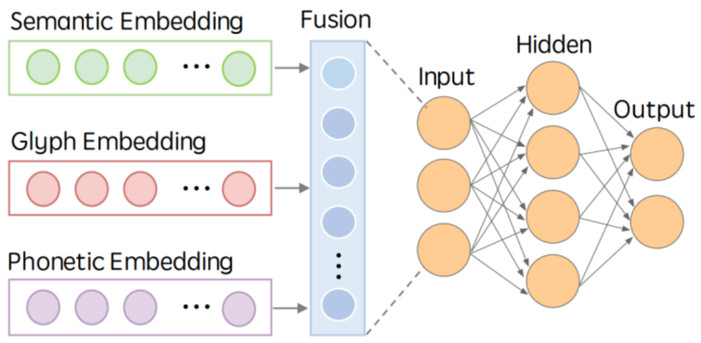
Third Stage: Multimodal Fusion

➢ Early Multimodal Fusion (EMF)

- The multimodal embedding vector is fused at the input-level, i.e.,

$$\boldsymbol{V} = [\boldsymbol{V}^{(S)} \oplus \boldsymbol{V}^{(G)} \oplus \boldsymbol{V}^{(P)}].$$

- EMF is easy to implement and requires less model parameters.



(a) EMF

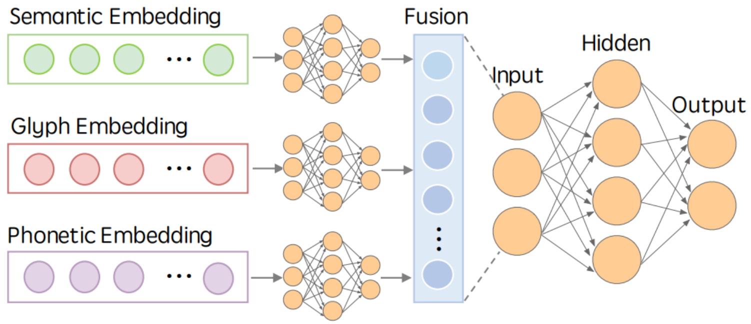
➢ Intermediate Multimodal Fusion (IMF)

- The multimodal embedding vector is fused based on the output of modality-specific networks, i.e.,

$$\boldsymbol{V} = [F_s(\boldsymbol{V}^{(S)}) \oplus F_g(\boldsymbol{V}^{(G)}) \oplus F_p(\boldsymbol{V}^{(P)})]$$

where $F_s(\cdot)$, $F_g(\cdot)$ and $F_p(\cdot)$ are the unimodal network specialized for semantics, glyphs and phonetics.

- It is a fine-grained fusion scheme.



(b) IMF

Defense Evaluation

Defense Evaluation

Dataset

- Abusive UGC and Pornographic UGC that collect from online social media
- Each dataset contains 10,000 toxic and 10,000 normal samples

Evaluated Model

- **Tasks:** Abusive content detection, Pornographic content detection
- **Offline models:** TextCNN, BiLSTM

Attack Method

- TextBugger

Baseline Algorithms

- Pycorrector, Baidu TextCorrector
- Industy-leading detection platforms:

Alibaba Cloud Baidu 百度 网易易盾 HUAWEI 华为云

Evaluation Metrics

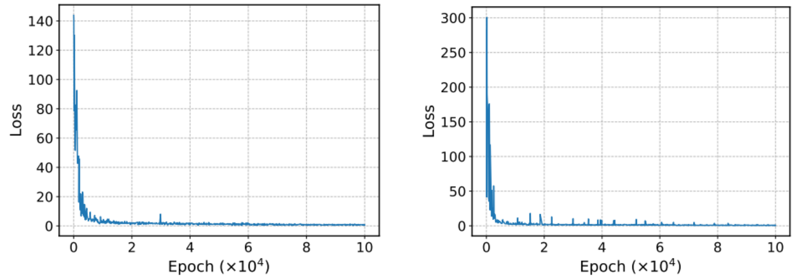
- **Translation Evaluation**
  - Word Error Rate (WER) $WER = \frac{S+D+I}{N}$
  - Bilingual Evaluation Understudy (BLEU) $BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n)$
  - Semantic Similarity (SS) $S(\boldsymbol{p}, \boldsymbol{q}) = \frac{\boldsymbol{p} \cdot \boldsymbol{q}}{||\boldsymbol{p}|| \cdot ||\boldsymbol{q}||} = \frac{\sum_{i=1}^{n} p_i \times q_i}{\sqrt{\sum_{i=1}^{n} (p_i)^2} \times \sqrt{\sum_{i=1}^{n} (q_i)^2}}$
- **Robustness Evaluation**
  - Attack Success Rate $\text{Success Rate} = \frac{\#\text{ success samples}}{\#\text{ total examples}}$
  - Perturbed Word
  - Query
- **Utility Evaluation**
  - Edit Distance
  - Jaccard Similarity Coefficient $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$
  - Semantic Similarity $S(\boldsymbol{p}, \boldsymbol{q}) = \frac{\boldsymbol{p} \cdot \boldsymbol{q}}{||\boldsymbol{p}|| \cdot ||\boldsymbol{q}||} = \frac{\sum_{i=1}^{n} p_i \times q_i}{\sqrt{\sum_{i=1}^{n} (p_i)^2} \times \sqrt{\sum_{i=1}^{n} (q_i)^2}}$

Defense Evaluation: Model Performance

Accuracy on Benign Texts

Table 2: The model accuracy under non-adversarial setting.

Model	Abuse Detection	Porn Detection
Common TextCNN	0.88	0.90
TextCNN + Pycorrector	0.84	0.88
TextCNN + TextCorrector	0.85	0.90
TextCNN + EMF	0.85	0.89
TextCNN + IMF	0.87	0.89
TextCNN + NMT	0.87	0.89
TextCNN + EMF + NMT	0.86	0.88
TextCNN + IMF + NMT	0.88	0.89
Common BiLSTM	0.86	0.87
BiLSTM + Pycorrector	0.82	0.84
BiLSTM + TextCorrector	0.83	0.87
BiLSTM + EMF	0.84	0.86
BiLSTM + IMF	0.85	0.88
BiLSTM + NMT	0.84	0.86
BiLSTM + EMF + NMT	0.84	0.85
BiLSTM + IMF + NMT	0.85	0.87



(a) Abuse (b) Porn

Figure 4: The training loss of the adversarial NMT model.

Table 3: The error correction performance.

Dataset	Abuse Detection			Porn Detection		
	WER	BLEU	SS	WER	BLEU	SS
Baseline	0.198	0.744	0.939	0.199	0.749	0.937
Pycorrector	0.223	0.687	0.906	0.213	0.701	0.911
TextCorrector	0.181	0.767	0.939	0.173	0.777	0.938
Adversarial NMT	**0.051**	**0.923**	**0.988**	**0.056**	**0.916**	**0.985**

Remarks

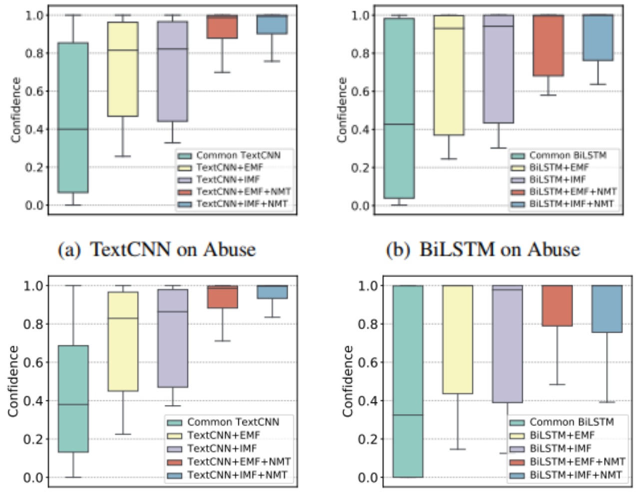
- TextShield has little impact on the model performance over benign texts and outperforms the baselines .
- It is feasible and easy to apply NMT to restore adversarial perturbations and also very effective.

Defense Evaluation: Effectiveness

Effectiveness in The Real-world Adversarial Scenario

Table 4: The detection performance on user generated obfuscated texts.

Model \ # of Perturbation	Abuse Detection				Porn Detection			
	≤ 1	≤ 2	≤ 3	> 3	≤ 1	≤ 2	≤ 3	> 3
Common TextCNN	0.488	0.483	0.480	0.458	0.496	0.448	0.426	0.398
TextCNN + Pycorrector	0.491	0.488	0.506	0.490	0.504	0.481	0.468	0.449
TextCNN + TextCorrector	0.498	0.484	0.485	0.457	0.568	0.563	0.558	0.555
TextCNN + EMF	0.790	0.783	0.760	0.736	0.753	0.742	0.732	0.718
TextCNN + IMF	0.714	0.725	0.732	0.729	0.777	0.767	0.751	0.730
TextCNN + NMT	0.857	0.886	0.869	0.836	0.909	0.899	0.887	0.870
TextCNN + EMF + NMT	**0.923**	**0.931**	0.919	**0.906**	0.928	0.921	0.908	0.901
TextCNN + IMF + NMT	0.922	0.931	**0.920**	0.904	**0.944**	**0.933**	**0.926**	**0.915**
Common BiLSTM	0.350	0.343	0.341	0.328	0.477	0.467	0.462	0.473
BiLSTM + Pycorrector	0.356	0.356	0.364	0.355	0.475	0.471	0.473	0.481
BiLSTM + TextCorrector	0.356	0.349	0.352	0.348	0.465	0.435	0.433	0.446
BiLSTM + EMF	0.604	0.616	0.620	0.605	0.746	0.725	0.730	0.724
BiLSTM + IMF	0.631	0.646	0.643	0.645	0.744	0.708	0.710	0.713
BiLSTM + NMT	0.801	0.791	0.764	0.707	0.856	0.804	0.778	0.757
BiLSTM + EMF + NMT	0.900	0.890	0.871	0.848	**0.933**	**0.913**	**0.903**	**0.890**
BiLSTM + IMF + NMT	**0.892**	**0.894**	**0.881**	**0.851**	0.932	0.906	0.891	0.882



(a) TextCNN on Abuse (b) BiLSTM on Abuse

(c) TextCNN on Porn (d) BiLSTM on Porn

Figure 5: The comparison of classification confidence

Remarks

- The models shielded by TextShield achieved a considerable high detection accuracy with high confidence over user generated obfuscated texts.
- The combined defense scheme is more effective and significantly outperforms the baselines.

Defense Evaluation: Robustness

Robustness Against Adaptive Attack

Table 7: The performance of adaptive attack against all the target models.

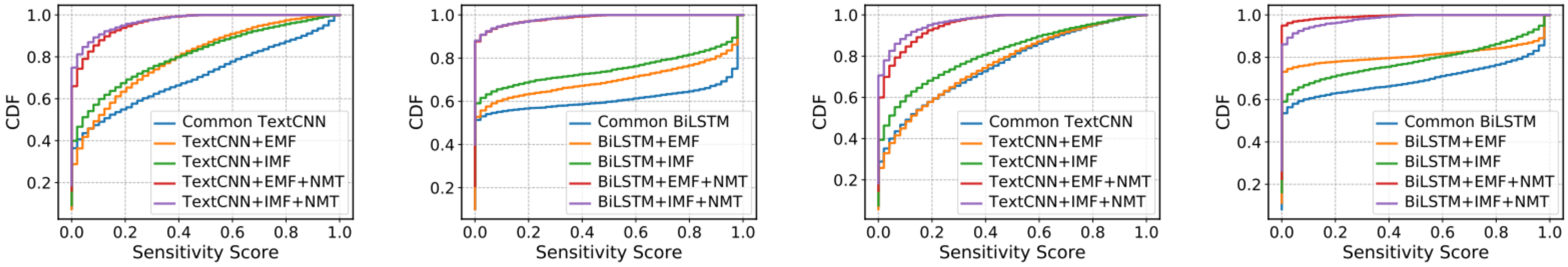
Model	Abuse Detection			Porn Detection		
	ASR	Perturbed Word	Query	ASR	Perturbed Word	Query
Common TextCNN	0.860	2.19	65.8	0.839	2.12	61.1
TextCNN + Pycorrector	0.830	1.91	61.9	0.823	2.01	59.4
TextCNN + TextCorrector	0.786	2.03	66.3	0.773	2.13	60.4
TextCNN + EMF	0.687	2.35	69.2	0.706	2.02	58.9
TextCNN + IMF	0.622	2.32	68.5	0.595	2.18	61.7
TextCNN + NMT	0.375	2.05	63.7	0.428	2.34	64.3
TextCNN + EMF + NMT	0.240	2.00	63.9	0.339	2.15	60.8
TextCNN + IMF + NMT	**0.219**	1.93	62.7	**0.236**	2.03	59.4
Common BiLSTM	0.891	1.87	61.7	0.846	2.11	61.3
BiLSTM + Pycorrector	0.872	1.68	58.7	0.835	1.75	55.9
BiLSTM + TextCorrector	0.866	1.83	59.5	0.821	1.95	60.9
BiLSTM + EMF	0.726	1.97	63.8	0.548	2.12	61.6
BiLSTM + IMF	0.555	1.87	62.0	0.550	2.14	61.8
BiLSTM + NMT	0.450	1.93	62.5	0.548	2.20	62.7
BiLSTM + EMF + NMT	0.268	1.85	62.2	**0.247**	2.03	60.3
BiLSTM + IMF + NMT	**0.238**	1.73	60.2	0.289	1.80	55.7

Remarks

- TextShield can significantly reduce the attack success rate and is more robust than the baselines.

Defense Evaluation: Robustness

Model Sensitivity Against Bug Replacement



(a) TextCNN on Abuse (b) BiLSTM on Abuse (c) TextCNN on Porn (d) BiLSTM on Porn

Figure 9: The sensitivity of the target models against bug replacement.

Remarks

- Both multimodal embedding and adversarial translation can significantly reduce the model sensitivity against the bug replacement.

Defense Evaluation: Practicality

Comparison With Industry-leading Platforms

Table 9: The comparison with real-world online detection services.

Targeted API	Abuse Detection				Porn Detection			
	Ori Accuracy	Success Rate	Perturbed Word	Query	Ori Accuracy	Success Rate	Perturbed Word	Query
Alibaba GreenNet	0.778	0.868	1.34	40.1	0.869	0.884	1.71	48.2
Baidu TextCensoring	0.763	0.938	1.36	33.4	**0.892**	**0.897**	**1.88**	**49.9**
Huawei Moderation	0.704	0.888	1.34	35.3	0.710	0.814	1.67	46.7
Netease Yidun	**0.805**	**0.903**	**1.38**	**42.1**	0.823	0.818	1.90	51.1
TextCNN + IMF + NMT	0.880	0.219	1.93	62.7	0.890	0.236	2.03	59.4
BiLSTM + EMF + NMT	0.840	0.268	1.85	62.2	0.850	0.247	2.03	60.3

Remarks

- The four industry-leading platforms who have claimed to be successful in tackling variant texts are still vulnerable to adversarial attack, and TextShield outperforms them by a big margin.
- We are currently in the process of integrating TextShield with Alibaba GreenNet to enhance its robustness.

Defense Evaluation: Generalizability

Extension to English-based NLP Models

Table 10: The results of adaptive attacks against English-based DLTC models with TEXTSHIELD.

Model	Accuracy	ASR	Perturbed Word	Query
Common TextCNN	0.754	0.880	1.60	36.7
TextCNN + EMF + NMT	0.757	0.283	1.53	37.5
TextCNN + IMF + NMT	0.752	**0.265**	1.38	36.4
Common BiLSTM	0.766	0.782	1.80	38.4
BiLSTM + EMF + NMT	0.751	0.351	1.54	37.7
BiLSTM + IMF + NMT	0.763	**0.285**	1.26	36.1

Experiment Setup

- Language: English
- Task: Sentiment Analysis
- Dataset: Rotten Tomatoes Movie Reviews (MR)
- Attack: The same adaptive setting

Remarks

- TextShield shows good generalizability across languages and can be extended some other languages.

Summary

We proposed TextShield, a defense specifically designed for Chinese-based DLTC models.

- **Effective:** It is effective in real-word adversarial scenarios while having little impact on the model performance under the non-adversarial setting.
- **Robust:** it significantly reduces the attack success rate even under the setting of adaptive attacks.
- **Generic:** it can be applied to any Chinese-based DLTC models without requiring re-training**.**

We compared TextShield with four industry-leading platforms

- **Practical:** It is of great practicability and superiority in decreasing the attack success rate .

We extend TextShield to English-based NLP models

- **Generalizability :** It shows good generalizability across language.

Q&A

lijinfeng0713@zju.edu.cn