Pay Attention to the Robustness of Chinese Minority Language Models! Syllable-level Textual Adversarial Attack on Tibetan Script

Xi Cao[1,2], Dolma Dawa[1,2], Nuo Qun[1,2*], Trashi Nyima[1,2]
[1]School of Information Science and Technology,
Tibet University, Lhasa, Tibet 850000, China
[2]Collaborative Innovation Center for Tibet Informatization by
MOE and Tibet Autonomous Region, Lhasa, Tibet 850000, China
metaphor@outlook.com, {da_zhui,q_nuo,nmzx}@utibet.edu.cn

Abstract

The textual adversarial attack refers to an attack method in which the attacker adds imperceptible perturbations to the original texts by elaborate design so that the NLP (natural language processing) model produces false judgments. This method is also used to evaluate the robustness of NLP models. Currently, most of the research in this field focuses on English, and there is also a certain amount of research on Chinese. However, to the best of our knowledge, there is little research targeting Chinese minority languages. Textual adversarial attacks are a new challenge for the information processing of Chinese minority languages. In response to this situation, we propose a Tibetan syllable-level black-box textual adversarial attack called TSAttacker based on syllable cosine distance and scoring mechanism. And then, we conduct TSAttacker on six models generated by fine-tuning two PLMs (pre-trained language models) for three downstream tasks. The experiment results show that TSAttacker is effective and generates high-quality adversarial samples. In addition, the robustness of the involved models still has much room for improvement.

1 Introduction

With the development of neural network models, methods based on the models have been widely used in many fields and achieved remarkable performance, such as computer vision, speech recognition, and natural language processing. However, neural network models are susceptible to adversarial attacks (Szegedy et al., 2013).

When textual adversarial attacks are performed on the NLP models for classification tasks, the models with high robustness will make predictions consistent with the original texts after perturbation, while the models with low robustness will make incorrect predictions. Section 2 will detail the current research status of textual adversarial attacks on English and Chinese. The information processing technology of Chinese minority languages started late, but in recent years, the emergence of Chinese minority PLMs has promoted development but brought new challenges, one of which is textual adversarial attacks. However, there is little research on this topic.

The main contributions of this paper are as follows:

(1) To fill the research gap of textual adversarial attacks on Tibetan script, this paper proposes TSAttacker, a Tibetan syllable-level black-box textual adversarial attack with a high attack success rate. This attack method can significantly reduce the accuracy of the models and generate adversarial samples with a low average Levenshtein distance.

(2) To evaluate the robustness of the Tibetan part in the first Chinese minority multilingual PLM, this paper conducts TSAttacker on six models generated by fine-tuning two versions of the PLM for three downstream tasks. During fine-tuning, we also find that training sets conforming to language standards can improve model performance.

(3) To facilitate future explorations, we open-source our work on GitHub (https://github.com/UTibetNLP/TSAttacker). We call on more researchers to pay attention to the security issues in the information processing of Chinese minority languages.

2 Related Work

2.1 Textual Adversarial Attacks on English

At present, most of the research on textual adversarial attacks concentrates on English. Jia and Liang (2017) first proposed generating adversarial samples for English public datasets and evaluating NLP models from a robustness perspective. Since then, various English textual adversarial attack methods with different strategies have emerged. According to the granularity of text perturbations, attacks can

* Corresponding author.

Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023), pages 35–46

be classified into character-, word-, and sentence-level (Du et al., 2021).

Character-level attacks are operations that perturb the characters of the original text, including adding, deleting, modifying, and changing the order of characters. Ebrahimi et al. (2018) proposed a character-level white-box attack method called HotFlip based on the gradients of the one-hot input vectors, Gao et al. (2018) proposed a greedy algorithm based on scoring called DeepWordBug for character-level black-box attacks, Eger et al. (2019) proposed a character-level white-box attack method called VIPER based on visual similarity, and so on.

Word-level attacks are to perturb the words of the original text, and the main method is word substitution. Such as, Jin et al. (2019) proposed a word-level black-box attack method called TextFooler which combines the cosine similarity of words with the semantic similarity of sentences, Garg and Ramakrishnan (2020) proposed a word-level black-box attack method based on the BERT mask language model called BAE, and Choi et al. (2022) proposed TABS, an efficient beam search word-level black-box attack method.

Sentence-level attacks generate adversarial sentences primarily through paraphrasing and text generation, which often result in a significant gap between the perturbed text and the original text. Moreover, it is difficult to control the quality of generated adversarial samples. The attack effect is also relatively average (Zheng et al., 2021).

2.2 Textual Adversarial Attacks on Chinese

The methods of generating adversarial texts are closely related to language characteristics, such as textual features and grammatical structure. Therefore, there are different methods of generating adversarial samples for various languages. The research on Chinese textual adversarial attacks started later than English, but there are also some related studies. Wang et al. (2019) proposed a Chinese word-level black-box attack method called WordHanding, which designed a new word importance calculation algorithm and utilized homonym substitution to generate adversarial samples. Tong et al. (2020) proposed a Chinese word-level black-box attack method called CWordAttacker, which used the targeted deletion scoring mechanism and substituted words with traditional Chinese and Pinyin. Zhang et al. (2022) proposed a Chinese character-level black-box attack method called PGAS, which generated adversarial samples with minor disturbance by replacing polyphonic characters. The relevant research on Chinese textual adversarial attacks is not sufficient, and the language features of Chinese are not fully integrated. So, there is still a lot of exploration space.

2.3 Textual Adversarial Attacks on Chinese Minority Languages

With the construction and development of information technology in Chinese minority areas like Inner Mongolia, Tibet, and Xinjiang, the corpus of Chinese minority languages has reached a certain scale. Recently, there have been some PLMs targeting or containing Chinese minority languages. It is worth mentioning that Yang et al. (2022) proposed CINO (Chinese mINOrity PLM), the first Chinese minority multilingual PLM, covering standard Chinese, Cantonese, Tibetan, Mongolian, Uyghur, Kazakh, Zhuang, and Korean. This model achieves SOTA (state-of-the-art) performance on multiple monolingual or multilingual datasets for text classification, significantly promoting the NLP research of Chinese minority languages.

Meanwhile, Morris et al. (2020) released an English textual adversarial attack frame called TextAttack, Zeng et al. (2021) released a textual adversarial attack toolkit called OpenAttack which supports both English and Chinese, Wang et al. (2021) released a robustness evaluation toolkit called TextFlint for English and Chinese NLP models, etc. These have provided a good research platform for other languages' textual adversarial attacks and model robustness evaluation.

However, to the best of our knowledge, there is little research involving textual adversarial attacks on Chinese minority languages such as Mongolian, Tibetan, and Uyghur. Without robustness evaluation, the NLP models with low robustness will face serious risks, such as hacker attacks, poor user experience, and political security problems, which pose a huge threat to the stable development and information construction of Chinese minority areas. Therefore, we should take precautions to study the textual adversarial attack methods of related languages and evaluate the robustness of related models to fill in the gaps in related research fields.

3 Attack Method

3.1 Textual Adversarial Attacks on Text Classification

For a K-class classification dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $x \in X$ (X includes all possible input texts) and $y \in Y$ (Y includes all K classifications). The classifier F obtains the classification y_{true} corresponding to the original input text x, denoted as

$$F(x) = arg\max_{y \in Y} P(y|x) = y_{true}. \quad (1)$$

The attacker achieves a successful textual adversarial attack by elaborately designing the adversarial text x' and making

$$F(x') = arg\max_{y \in Y} P(y|x') \neq y_{true}, \quad (2)$$

where x' is the result of adding ϵ-bounded, imperceptible perturbations δ to the original text x.

3.2 TSAttacker Algorithm

Tibetan is a phonetic script consisting of 30 consonant letters and 4 vowel letters. These letters are combined into Tibetan syllables according to certain rules. A Tibetan word is composed of one or more syllables separated by tsheg (·). Therefore, it is different from English and Chinese in that the syllable granularity in Tibetan is between character and word. Let the syllable in the original input text x be s (ignore tsheg and end-of-sentence punctuation), then

$$x = s_1 s_2 \dots s_i \dots s_n. \quad (3)$$

In this work, we propose a Tibetan syllable-level black-box textual adversarial attack called TSAttacker based on syllable cosine distance and scoring mechanism. We adopt syllable cosine distance to obtain syllables for substitution and a scoring mechanism to determine the order of syllable substitutions.

3.2.1 Syllable Substitution

Grave et al. (2018) released high-quality pre-trained word vectors for 157 languages, including Tibetan syllable embeddings, which were trained using fastText[1] (Joulin et al., 2016) on the dataset composed of a mixture of Wikipedia and Common Crawl. The Tibetan training result contains some unwanted vectors due to the nature of the training dataset, such as embeddings of "MP3", "PNG", and "File". Consequently, we cleaned the result and obtained 7,652 Tibetan syllable embeddings basically containing all commonly used syllables.

For each Tibetan syllable s in the original input text x, we use all syllables whose embedding's cosine distances from the embedding of s are within the range of $(0, d_{max}]$ as a candidate syllables' set C. Let the cosine distance between the embedding of s and the embedding of s' ($s' \in C$) be d, then d satisfies the following condition:

$$d = 1 - \frac{\mathbf{s} \cdot \mathbf{s}'}{|\mathbf{s}| \cdot |\mathbf{s}'|} \leq d_{max}. \quad (4)$$

By adjusting d_{max}, we can control the similarity between all syllables in set C and syllable s. The smaller d_{max} is, the more similar all syllables in set C are to syllable s. As a result, the size of set C can be adjusted. The larger d_{max}, the larger the size of set C.

For the i-th Tibetan syllable s_i in the original input text x, there is always a candidate syllables' set C_i corresponding to it. Assuming that the size of set C_i is m. We select a candidate syllable ${s_i}'$ from set C_i each time, and

$${x_i}' = s_1 s_2 \dots {s_i}' \dots s_n. \quad (5)$$

At the same time, we calculate

$$\Delta P_i = P(y_{true}|x) - P(y_{true}|{x_i}'). \quad (6)$$

After iterating set C_i, the syllable ${s_i}^*$ can be found, and

$${x_i}^* = s_1 s_2 \dots {s_i}^* \dots s_n. \quad (7)$$

At the moment,

$$\begin{aligned} \Delta {P_i}^* &= P(y_{true}|x) - P(y_{true}|{x_i}^*) \quad (8) \\ &= max\{\Delta P_{ij}\}_{j=1}^{m} \\ &= max\{P(y_{true}|x) - P(y_{true}|{x_i}'_j)\}_{j=1}^{m}, \end{aligned}$$

$$\begin{aligned} {s_i}^* &= arg\max_{{s_i}' \in C_i} \{\Delta P_{ij}\}_{j=1}^{m} \quad (9) \\ &= arg\max_{{s_i}' \in C_i} \{P(y_{true}|x) - P(y_{true}|{x_i}'_j)\}_{j=1}^{m}. \end{aligned}$$

The syllable ${s_i}^*$ obtained in this way can cause the most significant change in the classification probability after substitution and have the best attack effect. Therefore, we use syllable ${s_i}^*$ to substitute syllable s_i.

The pseudocode of the TSAttacker algorithm is as shown in Appendix A.

[1] https://fasttext.cc

3.2.2 Substitution Order

Word saliency (Li et al., 2016) refers to the degree of change in the classification probability after a word is set to unknown (out of vocabulary). Here, we use it to calculate syllable saliency. For the i-th Tibetan syllable s_i in the original input text x, we set it to "$< UNK >$", and

$$\hat{x_i} = s_1 s_2 \ldots < UNK > \ldots s_n. \quad (10)$$

Then, we calculate the saliency of syllable s_i as S_i:

$$S_i = P(y_{true}|x) - P(y_{true}|\hat{x_i}). \quad (11)$$

We incorporate the scoring formula in the probability weighted word saliency algorithm (Ren et al., 2019) to determine the substitution order of syllables in the original input text x. The score H_i is defined as follows:

$$\begin{aligned} H_i &= Softmax(S_i) \cdot \Delta P_i^* \quad (12) \\ &= \frac{e^{S_i}}{\Sigma_{j=1}^{n} e^{S_j}} \cdot \Delta P_i^*. \end{aligned}$$

From the above formula, it can be seen that the score H_i comprehensively considers the importance of the substituted syllable s_i and the substitution syllable ${s_i}^*$. After sorting n scores $\{H_1, H_2, \ldots, H_n\}$ corresponding to the original input text x in descending order, we sequentially substitute s_i with ${s_i}^*$. If $F(x') \neq F(x)$, the attack succeeds, and if always $F(x') = F(x)$, the attack fails.

4 Experiment

4.1 Datasets and Models

4.1.1 Datasets

Table 1 lists the detailed information of the datasets: TNCC-title, TNCC-document, and TU_SA, including task, number of classes, the average number of syllables, etc.

TNCC[1]. Qun et al. (2017) open-sourced the Tibetan News Classification Corpus (TNCC) collected from the China Tibet Online website (http://tb.tibet.cn). This corpus consists of two parts: TNCC-title, a news title classification dataset, and TNCC-document, a news document classification dataset. TNCC-title is a short text dataset with 9,276 samples and an average of 16 syllables per title. TNCC-document is a long text dataset with 9,204 samples and an average of 689 syllables per document. There are twelve classes both in TNCC-title and TNCC-document dataset: Politics, Economics, Education, Tourism, Environment, Language, Literature, Religion, Arts, Medicine, Customs, and Instruments.

TU_SA[2]. TU_SA is a Tibetan sentiment classification dataset consisting of 10,000 samples labeled as positive or negative, with 5,000 samples in each class. Zhu et al. (2023) selected 10,000 sentences from the public Chinese sentiment analysis datasets: weibo_senti_100k and ChnSentiCorp, then manually translated and proofread by professional researchers to form this dataset.

4.1.2 Models

The existing public PLMs targeting or containing Tibetan include the monolingual PLM TiBERT (Liu et al., 2022) based on BERT (Devlin et al., 2019) and the multilingual PLM CINO (Yang et al., 2022) based on XLM-R (Conneau et al., 2020), and CINO has achieved SOTA performance in relevant evaluations on Tibetan. We adopt two versions of CINO: cino-base-v2[3] and cino-large-v2[4], then fine-tune them for the three downstream tasks corresponding to the above datasets. Each dataset is split into a training set, a validation set, and a test set according to a ratio of 8:1:1. We select the best checkpoints based on the macro-F1 score for TNCC and the F1 score for TU_SA. The hyperparameters used for downstream fine-tuning are listed in Table 2.

It should be noted that the texts in TNCC have been pre-tokenized, which means that a space instead of a tsheg has been added between two syllables. When Yang et al. (2022) fine-tuned CINO on TNCC, they removed the spaces, but the processed texts do not conform to the standards of Tibetan script, and there should be a tsheg between two syllables. Therefore, we make a separate experiment that fine-tunes models on texts with a space between two syllables, texts with no space between two syllables, and texts with a tsheg between two syllables. The results of the validation sets are listed in the first 12 rows of Table 3 and show that models fine-tuned on the texts conforming to language standards can achieve better performance.

Table 3 list the performance of the models fine-tuned on TNCC and TU_SA. We

[1] https://github.com/FudanNLP/Tibetan-Classification

[2] https://github.com/UTibetNLP/TU_SA

[3] https://huggingface.co/hfl/cino-base-v2

[4] https://huggingface.co/hfl/cino-large-v2

Table 1: Detailed information of the datasets.

Dataset	Task	#Classes	#Average syllables	#Total samples	#Train samples	#Validation samples	#Test samples
TNCC-title	news title classification	12	16	9,276	7,422	927	927
TNCC-document	news document classification	12	689	9,204	7,364	920	920
TU_SA	sentiment classification	2	28	10,000	8,000	1,000	1,000

Table 2: Hyperparameters used for downstream fine-tuning.

Model	Dataset	Batch size	Epochs	Learning rate	Warmup ratio
cino-base-v2	TNCC & TU_SA	32	40	5e-5	0.1
cino-large-v2	TNCC & TU_SA	32	40	3e-5	0.1

adopt the following six models as victim models and conduct TSAttacker on the test sets: cino-base-v2+TNCC-title(tsheg), cino-base-v2+TNCC-document(tsheg), cino-large-v2+TNCC-title(tsheg), cino-large-v2+TNCC-document(tsheg), cino-base-v2+TU_SA, and cino-large-v2+TU_SA.

4.2 Evaluation Metrics and Experiment Results

We use *Accuracy Drop Value* (ADV) and *Attack Success Rate* (ASR) to evaluate both the attack effectiveness and the model robustness, and *Levenshtein Distance* (LD) to evaluate the quality of a generated adversarial sample. ADV refers to the difference in the model accuracy on the test set between pre-attack and post-attack. ASR refers to the percentage of the attack that successfully fool the victim model. The larger ADV or ASR, the more effective the attack and the less robust the model. LD refers to the minimum number of single-syllable edits between two texts, like insertions, deletions, and substitutions. The smaller LD, the higher the quality of the generated adversarial sample.

In this work, we set the maximum cosine distance d_{max} to 0.2929, in other words, the maximum angle between two syllable embeddings is $45°$. We use this parameter to determine the set of candidate substitution syllables according to Equation 4. Table 4 shows the experiment results and Appendix B lists some adversarial samples generated by TSAttacker.

The results show that our proposed attack method TSAttacker greatly reduces the model accuracy and has a high attack success rate, which shows the effectiveness of the attack method. For the dataset TNCC-title, the accuracy of the models cino-base-v2 and cino-large-v2 decreases by 0.3646 and 0.3430, and the attack success rate reaches 0.7605 and 0.7487, respectively; for the dataset TNCC-document, the accuracy of the models cino-base-v2 and cino-large-v2 decreases by 0.3859 and 0.3283, and the attack success rate reaches 0.7120 and 0.6696, respectively; for the dataset TU_SA, the accuracy of the models cino-base-v2 and cino-large-v2 decreases by 0.2240 and 0.2660, and the attack success rate reaches 0.6380 and 0.6570, respectively.

From a certain point of view, the robustness of Chinese minority NLP models still has much room for improvement. The model cino-base-v2 is a base version of CINO, with 12 layers, 768 hidden states, and 12 attention heads. The model cino-large-v2 is a large version of CINO, with 24 layers, 1024 hidden states, and 16 attention heads. However, for different datasets, the same attack method does not always achieve a higher attack success rate on the smaller model, and the larger model is not always the one with a smaller accuracy drop value. This seems to indicate that the model robustness is independent of the model size.

The results also show that our proposed attack method TSAttacker can generate high-quality adversarial samples because of the low average Levenshtein distance. The average number of syllables

Table 3: Model performance on TNCC and TU_SA..

Model (PLM+Dataset)	Accuracy	Macro-F1	Macro-Precision	Macro-Recall	Weighted-F1	Weighted-Precision	Weighted-Recall
cino-base-v2+ TNCC-title (space)	0.6624	0.6375	**0.6721**	0.6213	0.6564	0.6613	0.6624
cino-base-v2+ TNCC-title (no space)	0.6602	0.6385	0.6382	0.6454	0.6621	0.6716	0.6602
cino-base-v2+ TNCC-title (tsheg)	**0.6764**	**0.6488**	0.6523	**0.6556**	**0.6772**	**0.6853**	**0.6764**
cino-base-v2+ TNCC-document (space)	0.7380	0.6985	0.7039	0.6949	0.7382	0.7399	0.7380
cino-base-v2+ TNCC-document (no space)	0.7435	0.6967	0.7241	0.6817	0.7430	0.7501	0.7435
cino-base-v2+ TNCC-document (tsheg)	**0.7598**	**0.7317**	**0.7502**	**0.7180**	**0.7602**	**0.7630**	**0.7598**
cino-large-v2+ TNCC-title (space)	0.6785	0.6448	0.6489	0.6449	0.6767	0.6786	0.6785
cino-large-v2+ TNCC-title (no space)	0.6861	0.6568	0.6818	0.6429	0.6831	0.6874	0.6861
cino-large-v2+ TNCC-title (tsheg)	**0.7044**	**0.6759**	**0.6898**	**0.6672**	**0.7025**	**0.7062**	**0.7044**
cino-large-v2+ TNCC-document (space)	0.7380	0.6985	0.7039	0.6949	0.7382	0.7399	0.7380
cino-large-v2+ TNCC-document (no space)	0.7435	0.6967	0.7241	0.6817	0.7430	0.7501	0.7435
cino-large-v2+ TNCC-document (tsheg)	**0.7598**	**0.7317**	**0.7502**	**0.7180**	**0.7602**	**0.7630**	**0.7598**
cino-base-v2+ TU_SA	0.7530	0.7748 (F1)	0.7119 (Precision)	0.8500 (Recall)	-	-	-
cino-large-v2+ TU_SA	0.7970	0.7992 (F1)	0.7906 (Precision)	0.8080 (Recall)	-	-	-

Table 4: Experiment results.
ADV = Accuracy Drop Value, ASR = Attack Success Rate, LD = Levenshtein Distance.

Model (PLM+Dataset)	Accuracy (pre-attack)	Accuracy (post-attack)	ADV (↑)	ASR (↑)	Average LD (↓)
cino-base-v2+ TNCC-title(tsheg)	0.6731	0.3085	0.3646	0.7605	1.6411
cino-large-v2+ TNCC-title(tsheg)	0.6850	0.3420	0.3430	0.7487	1.7176
cino-base-v2+ TNCC-document(tsheg)	0.7576	0.3717	0.3859	0.7120	39.1800
cino-large-v2+ TNCC-document(tsheg)	0.7500	0.4217	0.3283	0.6696	41.9660
cino-base-v2+ TU_SA	0.7430	0.5190	0.2240	0.6380	2.9404
cino-large-v2+ TU_SA	0.7760	0.5100	0.2660	0.6570	2.7017

in the datasets TNCC-title, TNCC-document, and TU_SA is 16, 689, and 28. For the dataset TNCC-title, the average Levenshtein distance of the generated adversarial samples on the models cino-base-v2 and cino-large-v2 is 1.6411 and 1.7176, respectively; for the dataset TNCC-document, the average Levenshtein distance of the generated adversarial samples on the models cino-base-v2 and cino-large-v2 is 39.1800 and 41.9660, respectively; for the dataset TU_SA, the average Levenshtein distance of the generated adversarial samples on the models cino-base-v2 and cino-large-v2 is 2.9404 and 2.7017, respectively. Several examples in Appendix B intuitively demonstrate that the model's prediction transforms from one high-confidence classification to another after conducting TSAttacker.

4.3 Ablation Experiment

Since our experiments involve an artificially set parameter, the maximum cosine distance d_{max}, we explore the influence of d_{max} on various evaluation metrics through ablation experiments as follows. We set d_{max} to 0.1340, 0.2929, and 0.5, respectively, that is to say, we set the maximum angle between two syllable embeddings to $30°$, $45°$, and $60°$ to get the set of candidate substitution syllables, then we conduct TSAttacker on the six models. Figure 1 shows the results of the ablation experiments in the form of line charts. From the line charts, we can intuitively find that the larger d_{max}, the larger accuracy drop value and attack success rate, and the relationship between d_{max} and average Levenshtein distance is not significant. Although the larger d_{max}, the more effective the attack, the similarity between the substituted syllable and the substitution syllable may not be that high.

5 Discussion

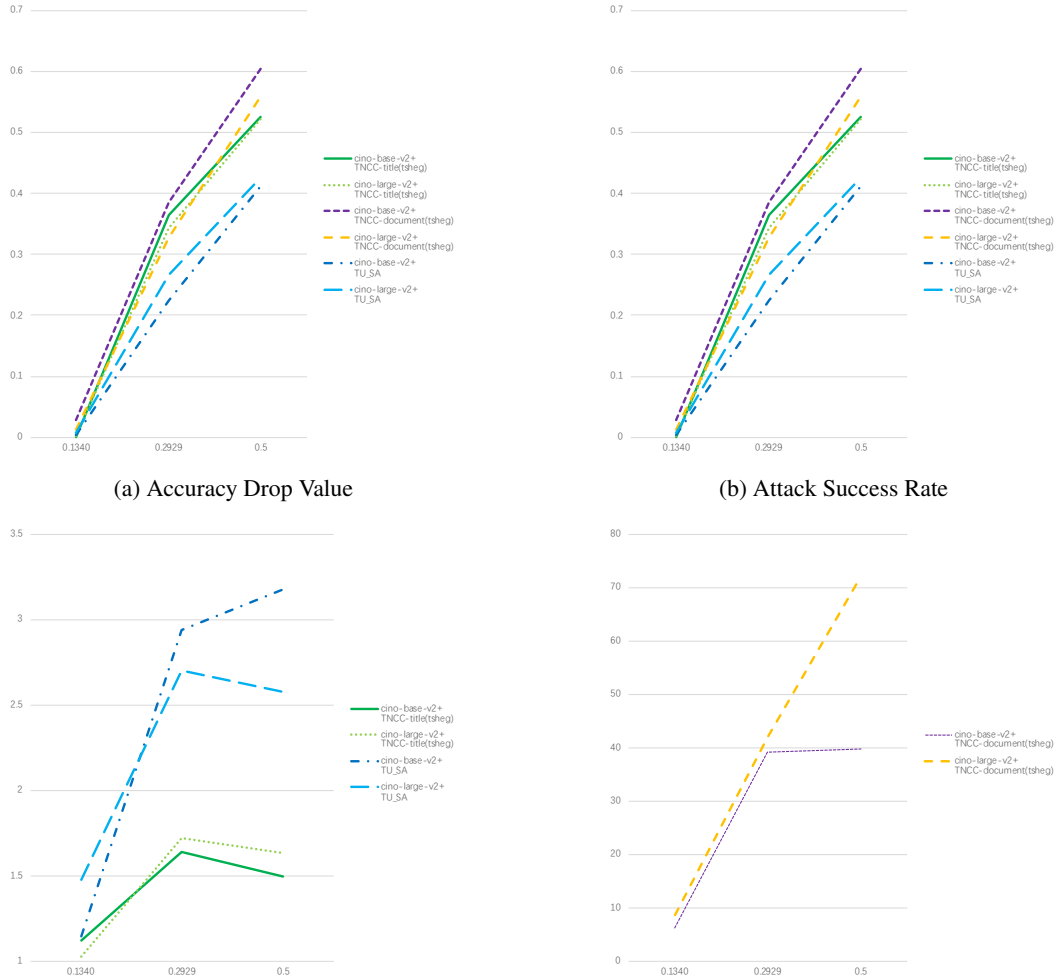
5.1 Textual Adversarial Attack is a Major Threat

Recently, Wang et al. (2023) evaluated the adversarial robustness of ChatGPT and found that the absolute performance of ChatGPT is far from perfection even though it outperforms most of the counterparts. Nowadays, more and more applications based on the services of foundation models appear, making various downstream scenarios face the risk of textual adversarial attacks worryingly. They also found that some small models achieve better performance on adversarial tasks while having much fewer parameters than the strong models. Therefore, there is still great space for research on the robustness and interpretability of neural network models.

5.2 Pay Attention to the Robustness of Chinese Minority Language Models

The textual adversarial attack is a new challenge for Chinese minority languages' information processing, which poses a major threat to the stable development and information construction of Chinese minority areas. China is a unified multi-ethnic

Figure 1: Results of ablation experiments.



(a) Accuracy Drop Value

(b) Attack Success Rate

(c) Average Levenshtein Distance

country. Due to the late start of information processing technology for Chinese minority languages, there is little research on the textual adversarial attack and defense of Chinese minority languages nowadays. With the development of neural network models, research in this field is now urgent.

From an attack perspective. The attack method proposed in this paper only preliminarily explores the field and evaluates the robustness of the Tibetan part in the first Chinese minority multilingual PLM. Moreover, the attack methods combined with the linguistic characteristics of Chinese minority languages need to be further proposed.

From a defense perspective. The overall performance of Chinese minority PLMs, including robustness, is far worse than that of English and Chinese PLMs. The main reason is that there is a huge gap in the quantity level between the corpus of Chinese minority languages and the corpus of English and Chinese. Therefore, this problem should be alleviated first. In addition, in response to the proposed textual adversarial attacks, a posterior defense is also an effective method.

6 Conclusion

In this work, we propose a Tibetan syllable-level black-box textual adversarial attack called TSAttacker. In TSAttacker, the syllable cosine distance is used to obtain syllables for substitution, and the scoring mechanism is used to determine the order of syllable substitutions. We conduct TSAttacker on six models generated by fine-tuning two versions of the PLM CINO for three downstream tasks. The experiment results show that TSAttacker greatly reduces the model accuracy and has a high attack success rate. Also, the adversarial samples generated by TSAttacker are high-quality. From a certain point of view, the robustness of the models still has much room for improvement.

Acknowledgements

We would like to express our sincere gratitude to the following funding sources for their support of this article: the “New Generation Artificial Intelligence" major project of Science and Technology Innovation 2030 (No. 2022ZD0116100), the National Natural Science Foundation of China (No. 62162057), and the Mount Everest Discipline Construction Project of Tibet University (Project No. zf22002001).

Limitations

Our work only preliminarily explores the field of textual adversarial attack on Chinese minority languages and evaluates the robustness of the Tibetan part in the first Chinese minority multilingual PLM. The textual adversarial attack is a major threat in the information processing of Chinese minority languages. We hope our attack method, experiment results, and discussions could provide experience for future research. In the future, we will continue to concentrate on the security issues faced in Tibetan information processing.

Ethics Statement

The purpose of this paper is to show that our proposed attack method is effective and the robustness of the SOTA Chinese minority PLM CINO still has much room for improvement, but not to attack it intentionally. Finally, we call on more researchers to pay attention to the security issues in the information processing of Chinese minority languages.

References

YunSeok Choi, Hyojun Kim, and Jee-Hyong Lee. 2022. TABS: Efficient textual adversarial attack for pre-trained NL code model using semantic beam search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5490–5498, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Xiaohu Du, Hongming Wu, Zibo Yi, Shasha Li, Jun Ma, and Jie Yu. 2021. Adversarial text attack and defense: A review. *Journal of Chinese Information Processing*, 35(08):1–15.

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.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. 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 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.

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.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI Conference on Artificial Intelligence*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *ArXiv*, abs/1612.08220.

Sisi Liu, Junjie Deng, Yuan Sun, and Xiaobing Zhao. 2022. Tibert: Tibetan pre-trained language model. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2956–2961.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Nuo Qun, Xing Li, Xipeng Qiu, and Xuanjing Huang. 2017. End-to-end neural text classification for tibetan. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 472–480, Cham. Springer International Publishing.

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.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Xin Tong, Luona Wang, Runzheng Wang, and Jingya Wang. 2020. A generation method of word-level adversarial samples for chinese text classification. *Netinfo Security*, 20(09):12–16.

Jindong Wang, Xixu Hu, Wenxin Hou, Haoxing Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Weirong Ye, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xingxu Xie. 2023. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *ArXiv*, abs/2302.12095.

Wenqi Wang, Run Wang, Lina Wang, and Benxiao Tang. 2019. Adversarial examples generation approach for tendency classification on chinese texts. *Journal of Software*, 30(08):2415–2427.

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

Ziqing Yang, Zihang Xu, Yiming Cui, Baoxin Wang, Min Lin, Dayong Wu, and Zhigang Chen. 2022. CINO: A Chinese minority pre-trained language model. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3937–3949, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Shunxiang Zhang, Houyue Wu, Guangli Zhu, Xin Xu, and Mingxing Su. 2022. Character-level adversarial samples generation approach for chinese text classification. *Journal of Electronics & Information Technology*.

Haibin Zheng, Jinyin Chen, Yan Zhang, Xuhong Zhang, Chunpeng Ge, Zhe Liu, Yike Ouyang, and Shouling Ji. 2021. Survey of adversarial attack, defense and robustness analysis for natural language processing. *Journal of Computer Research and Development*, 58(08):1727–1750.

Yulei Zhu, Kazhuo Deji, Nuo Qun, and Tashi Nyima. 2023. Sentiment analysis of tibetan short texts based on graphical neural networks and pre-training models. *Journal of Chinese Information Processing*, 37(02):71–79.

A Pseudocode of TSAttacker Algorithm

Algorithm 1: TSAttacker Algorithm

Input: Classifier F.
Input: Original text $x = s_1 s_2 \dots s_i \dots s_n$.
Input: Maximum cosine distance d_{max}.
Output: Adversarial text x'.

for $i \leftarrow 1$ *to* n **do**
  $\hat{x_i} \leftarrow s_1 s_2 \dots < UNK > \dots s_n$ // Equation 10
  $S_i \leftarrow P(y_{true}|x) - P(y_{true}|\hat{x_i})$ // Equation 11
end
Init H as a empty list.
for $i \leftarrow 1$ *to* n **do**
  Get the candidate syllables' set C_i according to syllable s_i and d_{max}.
  $m \leftarrow len(C_i)$
  for $j \leftarrow 1$ *to* m **do**
    ${s_i}' \leftarrow C_{ij}$
    ${x_i}' \leftarrow s_1 s_2 \dots {s_i}' \dots s_n$ // Equation 5
    $\Delta P_i \leftarrow P(y_{true}|x) - P(y_{true}|{x_i}')$ // Equation 6
  end
  $\Delta {P_i}^* \leftarrow max\{\Delta P_{ij}\}_{j=1}^{m}$ // Equation 8
  ${s_i}^* \leftarrow arg\max_{{s_i}' \in C_i}\{\Delta P_{ij}\}_{j=1}^{m}$ // Equation 9
  $H_i \leftarrow \frac{e^{S_i}}{\Sigma_{j=1}^{n} e^{S_j}} \cdot \Delta {P_i}^*$ // Equation 12
  Append $({s_i}^*, H_i)$ into H.
end
Sort H by the second parameter in descending order.
foreach *element in* H **do**
  $x' \leftarrow s_1 s_2 \dots {s_i}^* \dots s_n$
  if $F(x') \neq F(x)$ **then**
    Attack succeeds and return x'.
  end
end
Attack fails and return.

B Some Generated Adversarial Samples

Model	Input	Output (pre-attack)	Output (post-attack)
cino-large-v2+ TNCC-title (tsheg)	བོད་ལྗོངས་ཀྱི་མཐོ་རྒྱུགས་**ཀྱི**(ཞིང)་རེ་འདུན་རེའུ་མིག་རྒྱང་བའི་དུས་ཚོད་ཟླཚེསཉིན་ནས་ཟླཚེསཉིན་བར་ཡིན་པ	Education (92.95%)	Economics (97.35%)
cino-large-v2+ TNCC-document (tsheg)	བོད་ལྗོངས་ཉིན་རེའི་ཚགས་པར་མིག་སྔར་གསར་འགོད་པས་རང་སྐྱོང་ལྗོངས་སྲིད་ཁང་དང་གྲོང་དང་གྲོང་གསེབ་འཛུགས་སྐྲུན་ཐིང་ནས་རྒྱུས་ལོན་བྱས་པ་ལྟར་ན་ལོ་ལྔ་ཚན་བཅུ་གཉིས་པའི་རིང་ལ་བོད་ལྗོངས་ཀྱིས་གྲོང་དང་གྲོང་རྡལ་གྲོང་སྲིད་རྨང་གཞི་འཛུགས་སྐྲུན་སྒྲིག་བཀོད་ཐད་མ་དངུལ་འབོར་ཆེན་བཏང་ནས་ས་གནས་ཁག་གི་གྲོང་ཁྱེར་དང་གྲོང་རྡལ་གྱི་རྨང་གཞིའི་ཆ་རྐྱེན་སྔར་ལས་ལྷག་པར་ལེགས་བཅོས་བྱེད་རྒྱུ་རེད་རང་སྐྱོང་ལྗོངས་སྲིད་ཁང་དང་གྲོང་དང་གྲོང་གསེབ་འཛུགས་སྐྲུན་ཐིང་གི་གྲོང་དང་གྲོང་གསེབ་འཛུགས་སྐྲུན་དང་དོ་དམ་ཁང་གི་འགན་ཁུར་བས་མཚམས་སྦྱོར་བྱས་པ་ལྟར་ན་ལོ་ལྔ་ཚན་བཅུ་གཉིས་བའི་རིང་ལ་ལྗོངས་ཡོངས་ཀྱིས་གྲོང་དང་གྲོང་རྡལ་གྱི་རྨང་གཞི་སྒྲིག་བཀོད་ཀྱི་ཕྱོགས་སུ་མ་དངུལ་སྒོར་**དུང**(ཡིག)་ཕྱུར་༤༢ལས་མང་བར་བཏང་པའི་ཁྲོད་གཞུང་ལམ་དང་ཆུ་ཡུར་རམ་བཅོག་ཆུ་གཙང་བཟོར་བྱེད་པ་སོགས་ཀྱི་རྣམ་གྲངས་༡༢༤ཚུད་ཡོད་པར་མ་ཟད་ལྷ་ས་གྲོང་ཁྱེར་གྱི་སྐོར་ལམ་དང་ལྷ་སའི་སྣེའུ་དུས་རབས་ལམ་ཆེན་གཞིས་ཀ་རྩེ་གྲོང་གི་སྐོར་ལམ་ལྷོ་ཁ་རྩེ་ཐང་གཞུང་ལམ་ལྷ་ས་སྣེའུ་ཆུའི་བཟོ་གྲྭ་གཞིས་ཀ་རྩེ་ཆུའི་བཟོ་གྲྭ་ཆབ་མདོ་ཆུའི་བཟོ་གྲྭ་གཉིས་པ་སོགས་རྣམ་གྲངས་མང་དུ་འདུས་འདི་ནི་ལོ་ལྔའི་འཆར་འགོད་ཁྲོད་ཀྱི་མ་དངུལ་བཏང་ཚད་ཆེས་མང་བ་རེད་འགན་ཁུར་མི་སྣས་འདི་སྐད་ལོ་ལྔ་ཚན་བཅུ་གཉིས་པའིརིང་ལ་ལས་འགོ་འཛུགས་རྒྱུའི་རྣམ་གྲངས་འདི་དག་ནི་ས་གནས་ཁག་གམ་རྫོང་ཁག་ལ་མཁོ་ངེས་ཀྱི་རྣམ་གྲངས་རེད་ཐོག་མར་ང་ཚོས་རྫོང་མཁར་གྱི་མང་ཚོགས་ལ་བཏུང་བྱ་ཆུ་ཡི་ཕྱོགས་ནས་ཁག་ཐེག་བྱ་བ་དང་དེ་ནས་ཆུ་འབུད་དང་མཐའ་མར་གྲོང་ཁྱེར་གཞུང་ལམ་བཟོ་དགོས་ལོ་ལྔ་ཚན་བཅུ་གཉིས་ཀྱི་རིང་ལ་ལྗོངས་ཡོངས་ཀྱིས་རྫོང་མཁར་དུ་ཆུ་མགོ་འདོན་བྱས་ཚད་མ་དངུལ་སྒོར་དུང་ཕྱུར་གཉིས་ལ་སླེབས་ཡོད་ཅེས་བཤད་འདིར་བཀོད་པའི་ཡིག་རིགས་དང་པར་རིས་རྣམས་ཀྱི་པར་དབང་བོད་ལྗོངས་ཆ་འཕྲིན་ལྟེ་གནས་ལ་དབང་བས་ཕན་ཚུན་དབང་བཙོལ་དང་གྲོས་མོལ་མ་བྱས་གོང་སུ་ཞིག་གིས་ཀྱང་བཀོལ་སྤྱོད་བྱས་མི་ཆོག་པ་དགོངས་འཇགས་ཞུའོ་ཁ་པར	Economics (91.24%)	Environment (99.86%)
cino-large-v2+ TU_SA	སྣབས་ཐུག་དང་རིང་དུ་གྱེས་ནས་འཚོ་བར་གཅེས་སྤྲས་བྱེད་པ་དེ་རིང་རྒྱུན་དུ་ཆར་བ་བབས་**པ**(།)་མིན་ན་ངས་ཀྱང་རོལ་དབྱངས་དཀར་ཡོལ་གང་འཐུང་མི་སྲིད།	Negative (94.74%)	Positive (99.91%)

Substituted syllables are marked in bold and red. Substitution syllables are in parentheses.