Human-in-the-Loop Generation of Adversarial Texts: A Case Study on Tibetan Script

Xi Cao[♡♠], **Yuan Sun**[♡♠✉],
Jiajun Li[◇♣], **Quzong Gesang**[◇♣], **Nuo Qun**[◇♣✉], **Tashi Nyima**[◇♣]
[♡]Institute of National Security, Minzu University of China, Beijing, China
[◇]School of Information Science and Technology, Xizang University, Lhasa, China
[♠]National Language Resource Monitoring & Research Center | Minority Languages Branch, Beijing, China
[♣]The State Key Laboratory of Tibetan Intelligence, Lhasa, China
caoxi@muc.edu.cn, sunyuan@muc.edu.cn, q_nuo@utibet.edu.cn

Abstract

DNN-based language models excel across various NLP tasks but remain highly vulnerable to textual adversarial attacks. While adversarial text generation is crucial for NLP security, explainability, evaluation, and data augmentation, related work remains overwhelmingly English-centric, leaving the problem of constructing high-quality and sustainable adversarial robustness benchmarks for lower-resourced languages both difficult and understudied. First, method customization for lower-resourced languages is complicated due to linguistic differences and limited resources. Second, automated attacks are prone to generating invalid or ambiguous adversarial texts. Last but not least, language models continuously evolve and may be immune to parts of previously generated adversarial texts. To address these challenges, we introduce HITL-GAT[1], an interactive system based on a general approach to human-in-the-loop generation of adversarial texts. Additionally, we demonstrate the utility of HITL-GAT through a case study on Tibetan script, employing three customized adversarial text generation methods and establishing its first adversarial robustness benchmark, providing a valuable reference for other lower-resourced languages.

1 Introduction

The adversarial attack refers to an attack method in which the attacker adds imperceptible perturbations to the original input, resulting in the incorrect judgment of a DNN-based model. The examples generated during textual adversarial attacks are called adversarial texts.

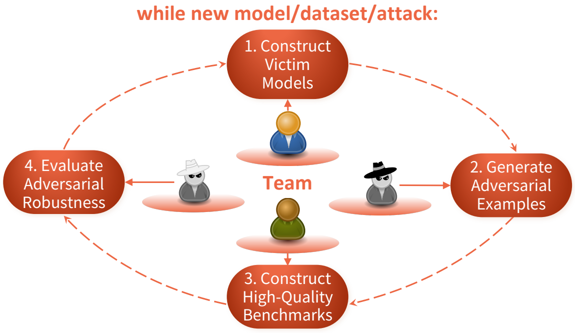
while new model/dataset/attack:
1. Construct Victim Models
2. Generate Adversarial Examples
3. Construct High-Quality Benchmarks
4. Evaluate Adversarial Robustness
Team

Figure 1: Workflow of HITL-GAT. While a new language model, downstream dataset, or textual adversarial attack method emerges, we can enter the loop to make the adversarial robustness benchmark evolve.

Due to the general adaptability of language models to classification tasks, adversarial robustness evaluation is mainly focused on the domain. Currently, most of the adversarial text generation methods target higher-resourced languages, especially English. Because of the differences in textual features and language resources, it is challenging to transfer these methods to other languages. **Problem 1: How do we generate adversarial texts for lower-resourced languages?**

Wang et al. (2021a) apply 14 textual adversarial attack methods to GLUE tasks (Wang et al., 2019) to construct the widely used adversarial robustness benchmark AdvGLUE. In their construction, they find that most textual adversarial attack methods are prone to generating invalid or ambiguous adversarial texts, with around 90% either changing the original semantics or hindering the annotators' unanimity. In our case study on Tibetan script, we also come to the same conclusion. **Problem 2: How do we construct high-quality adversarial robustness benchmarks?**

Wang et al. (2023) employ ANLI (Nie et al., 2020) and AdvGLUE (Wang et al., 2021a) to assess the adversarial robustness of ChatGPT and

✉ Corresponding Author

[1] Video Demonstration:
https://youtu.be/tXla4yAggwA
Code Repository:
https://github.com/CMLI-NLP/HITL-GAT
Victim Models:
https://huggingface.co/collections/UTibetNLP/tibetan-victim-language-models-669f614ecea872c7211c121c

Proceedings of The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations, pages 9–16
December 20-24, 2025 ©2025 Association for Computational Linguistics

several previous popular language models and find ChatGPT is the best. However, both ANLI and AdvGLUE are constructed using fine-tuned BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as victim models. Language models are evolving, while adversarial robustness benchmarks never. We argue that new language models may be immune to part of previously generated adversarial texts. Lower-resourced languages are at a very early stage of adversarial robustness evaluation compared to higher-resourced languages, and it is essential to envisage sustainable adversarial robustness evaluation in advance. **Problem 3: How do we update adversarial robustness benchmarks?**

To address the above problems, we introduce `HITL-GAT`, an interactive system for human-in-the-loop generation of adversarial texts. Figure 1 depicts the workflow of `HITL-GAT`. In a loop where a new language model, downstream dataset, or textual adversarial attack method emerges, our team starts to construct victim models, generate adversarial examples, construct high-quality benchmarks, and evaluate adversarial robustness. The loop allows adversarial robustness benchmarks to evolve along with new models, datasets, and attacks **(Problem 3)**. Figure 2 depicts the four stages in one pipeline detailedly. Firstly, we fine-tune the previous model and the new model on the same downstream datasets to construct victim models. Subsequently, we implement adversarial attacks on the victim models constructed from the previous model upon downstream datasets to generate adversarial examples. Afterward, we customize filter conditions and conduct human annotation to construct a high-quality adversarial robustness benchmark **(Problem 2)**. Finally, we evaluate the adversarial robustness of the new model on the benchmark. Additionally, we make a case study on one lower-resourced language, Tibetan, based on the general human-in-the-loop approach to adversarial text generation **(Problem 1)**.

The contributions of this paper are as follows:

(1) We propose a general human-in-the-loop approach to adversarial text generation. This approach can assist in constructing and updating high-quality adversarial robustness benchmarks with the emergence of new language models, downstream datasets, and textual adversarial attack methods.

(2) We develop an interactive system called `HITL-GAT` based on the general approach to human-in-the-loop generation of adversarial texts. This system is successfully applied to a case study on one lower-resourced language.

(3) We demonstrate the utility of `HITL-GAT` through a case study on Tibetan script, employing three customized adversarial text generation methods and establishing its first adversarial robustness benchmark, providing a valuable reference for other lower-resourced languages.

(4) We open-source both the system and the case study under GNU General Public License v3.0 to facilitate future explorations. Our code repository received 42 stars, and our 12 victim models were downloaded more than 5,000 times before paper submission on November 15, 2025.

2 Related Work

2.1 Textual Adversarial Attack Frameworks

TextAttack (Morris et al., 2020) and OpenAttack (Zeng et al., 2021) are two powerful and easy-to-use Python frameworks for textual adversarial attacks. They are both for text classification, supporting English and Chinese, with similar toolkit functionality and complementary attack methods. From a developer's perspective, TextAttack utilizes a relatively rigorous architecture to unify different attack methods, while OpenAttack is more flexible. SeqAttack (Simoncini and Spanakis, 2021) and RobustQA (Boreshban et al., 2023) are textual adversarial attack frameworks for named entity recognition and question answering, respectively, supporting English only. These frameworks provide an excellent platform to stress-test the adversarial robustness of models targeting higher-resourced languages. However, the weaponization of lower-resourced languages against NLP security (Lent, 2025; Yoo et al., 2025; Lent et al., 2025) highlights the urgent need for research in this area. To our knowledge, `HITL-GAT` is the first interactive system to build adversarial robustness benchmarks from scratch for a truly low-resource language.

2.2 Human-in-the-Loop Adversarial Text Generation

Wallace et al. (2019) guide human authors to keep crafting adversarial questions to break the question answering models with the aid of visual model predictions and interpretations. They conduct two rounds of adversarial writing. In the first round, human authors attack a traditional ElasticSearch model `A` to construct the adversarial set `x`. Then, they use `x` to evaluate `A`, a bidirectional recurrent neural network model `B`, and a deep averaging net-

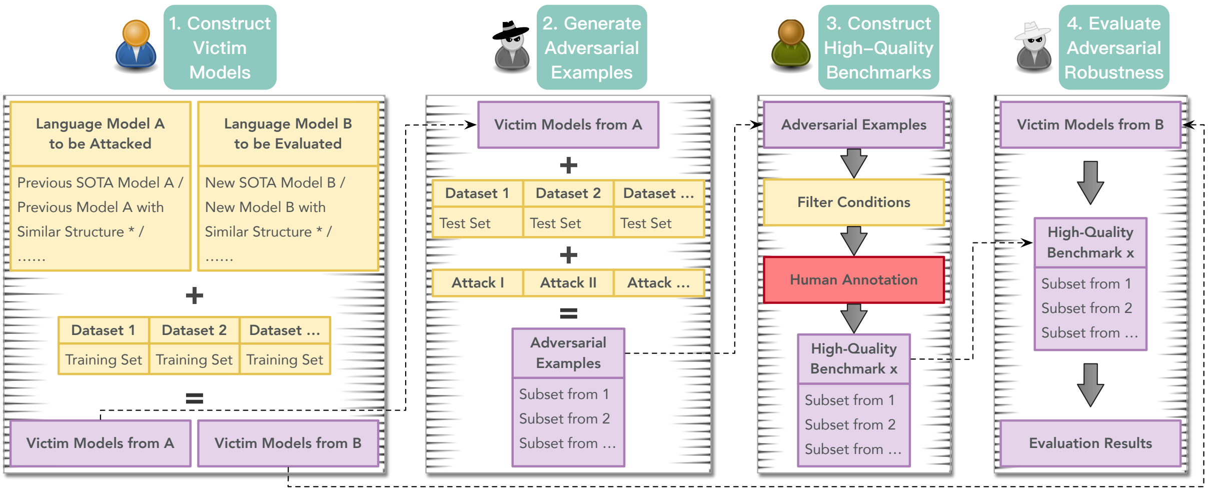

Figure 2: Flowchart of HITL-GAT. Our system contains four stages in one pipeline: **victim model construction**, **adversarial example generation**, **high-quality benchmark construction**, and **adversarial robustness evaluation**. System outputs are highlighted in purple background . Human choices are highlighted in yellow background . Human annotation is highlighted in red background .

work model C. In the second round, they train A, B, and C on a larger dataset. Human authors attack A and B to construct the adversarial set x and x'. Then, they use x and x' to evaluate A, B, and C. We see their human-in-the-loop approach as an embryo of adversarial robustness benchmark evolution, despite the high labor cost of relying on human authors to think and write adversarial texts. Most goals of using a human-in-the-loop approach in NLP tasks are to improve the model performance in various aspects (Wang et al., 2021b). With these goals, language models evolve. As continuous advancement of model capabilities, it is imperative to explore the paradigm for benchmark evolution. To our knowledge, even though our work is preliminary, we are the first to explore the evolution of adversarial robustness benchmarks.

3 Implementation

Definition Due to the general adaptability of language models to the text classification task, our work focuses on the adversarial robustness evaluation of language models on this task. The definition of textual adversarial attacks on text classification is as follows. For a text classifier F, let x ($x \in X$, X includes all possible input texts) be the original input text and y ($y \in Y$, Y includes all possible output labels) be the corresponding output label of x, denoted as $F(x) = \arg\max_{\dot{y} \in Y} P(\dot{y}|x) = y$. For a successful textual adversarial attack, let $x' = x + \delta$ be the perturbed input text, where δ is the imperceptible perturbation, denoted as $F(x') = \arg\max_{\dot{y} \in Y} P(\dot{y}|x') \neq y$.

Overview Our system for human-in-the-loop generation of adversarial texts, HITL-GAT, contains four stages in one pipeline: **victim model construction**, **adversarial example generation**, **high-quality benchmark construction**, and **adversarial robustness evaluation**. Figure 2 depicts the flowchart of HITL-GAT. These four stages will be detailed in the following four subsections respectively. Our flexible interactive system allows users to either go through the entire pipeline or directly start at any stage. Gradio (Abid et al., 2019) is an open-sourced Python package that allows developers to quickly build a web demo or application for machine learning. LlamaBoard is the user-friendly GUI (Graphical User Interface) of LlamaFactory (Zheng et al., 2024). The GUI of our system is powered by Gradio and draws inspiration from the design of LlamaBoard.

3.1 Construct Victim Models

This stage aims at constructing victim language models via a fine-tuning paradigm.

When a new language model B emerges, in order to better evaluate the adversarial robustness of B, we need to quantitatively and thoroughly perform evaluation on multiple downstream tasks. For the purpose of stress-testing the adversarial robustness of B more effectively, i.e., constructing a stronger adversarial robustness benchmark with high quality, we can choose at least one previous SOTA or similar-structured language model A to implement textual adversarial attacks on it to generate updated adversarial texts. We can also follow this stage when a new downstream dataset n is available.

In this stage, we fine-tune A and B on the training set of the same downstream datasets 1,2,...,n to construct victim language models. The victim model construction stage is depicted in the first part of Figure 2.

3.2 Generate Adversarial Examples

This stage aims at automatically generating the first-round adversarial texts with the help of various textual adversarial attack methods.

The way human authors keep thinking and writing adversarial texts (Wallace et al., 2019) is high-labor-cost. With the emergence of automated textual adversarial attacks, such as TextFooler (Jin et al., 2020), BERT-ATTACK (Li et al., 2020), SemAttack (Wang et al., 2022), and TextCheater (Peng et al., 2024), adversarial text generation has become relatively easy. We can directly enter this stage when a new textual adversarial attack N appears.

In this stage, we implement textual adversarial attacks I,II,...,N on the victim language models constructed from language model A upon the test set of downstream datasets 1,2,...,n to generate the first-round adversarial texts automatically. The adversarial example generation stage is depicted in the second part of Figure 2.

3.3 Construct High-Quality Benchmarks

This stage aims at constructing a high-quality adversarial robustness benchmark by customizing filter conditions and conducting human annotation.

The construction process of AdvGLUE (Wang et al., 2021a), a widely used adversarial robustness benchmark, tells us that most textual adversarial attack methods are prone to generating invalid or ambiguous adversarial texts, with around 90% either changing the original semantics or hindering the annotators' unanimity. Therefore, human annotation is indispensable and can make benchmarks more practical and relevant. In order to reduce the cost of human annotation, the first-round adversarial texts need to be screened automatically first using appropriate filter conditions. Due to the fact that humans perceive texts through their eyes and brains, both filter conditions and human annotation should follow the visual and semantic similarity between adversarial texts and original texts. Filter conditions can be the following metrics: Edit Distance, Normalized Cross-Correlation Coefficient (from the perspective of visual similarity); Cosine Similarity, BERTScore (Zhang et al., 2020) (from the perspective of semantic similarity); and so on. Human annotation still requires additional consideration of annotators' unanimity so that adversarial texts can be deemed human-acceptable. For example, given an original text and an adversarial text, we ask several annotators to score the human acceptance of the adversarial text based on the visual and semantic similarity between the two texts, from 1 to 5. The higher the score, the higher the human acceptance. If all annotators score the human acceptance of the adversarial text as 4 or 5, the adversarial text will be included in the adversarial robustness benchmark.

In this stage, we screen out the examples that do not satisfy the customized filter conditions from the first-round adversarial texts, and then manually annotate the remaining examples to construct the high-quality adversarial robustness benchmark x. The high-quality benchmark construction stage is depicted in the third part of Figure 2.

3.4 Evaluate Adversarial Robustness

This stage aims at quantitatively and thoroughly evaluating the adversarial robustness of new language models using the constructed high-quality adversarial robustness benchmark.

The adversarial robustness benchmark x is a collection of n subsets, each of which contains high-quality adversarial texts generated from the test set of the corresponding downstream dataset. We take the average accuracy on n subsets as the adversarial robustness ($AdvRobust$) of the new language model B on x, denoted as:

$$AdvRobust = \frac{\sum_{i=1}^{n} Accuracy_{\,i}}{n}. \tag{1}$$

In this stage, we utilize the constructed high-quality adversarial robustness benchmark x to evaluate the adversarial robustness of the language model B quantitatively and thoroughly. The adversarial robustness evaluation stage is depicted in the fourth part of Figure 2.

4 Case Study

In this section, we go through the entire pipeline under the existing conditions to construct the first adversarial robustness benchmark for Tibetan script and conduct the adversarial robustness evaluation on Tibetan language models. We will introduce the existing conditions and the whole process in the following two subsections respectively.

4.1 Existing Conditions

Below is the involved language models, downstream datasets, and attack methods.

4.1.1 Language Models

Tibetan-BERT[1] (Zhang et al., 2022). A BERT-based monolingual model targeting Tibetan. It is the first Tibetan BERT model and achieves a good result on the specific downstream Tibetan text classification task.

CINO[2] (Yang et al., 2022). A series of XLM-RoBERTa-based multilingual models including Tibetan. It is the first multilingual model for Chinese minority languages and achieves a SOTA performance on multiple downstream monolingual or multilingual text classification task.

4.1.2 Downstream Datasets

TNCC-title[3] (Qun et al., 2017). A Tibetan news title classification dataset. This dataset contains a total of 9,276 Tibetan news titles, which are divided into 12 classes.

TU_SA[4] (Zhu et al., 2023). A Tibetan sentiment analysis dataset. It is built by translating and proofreading 10,000 sentences from two public Chinese sentiment analysis datasets. In this dataset, negative or positive class each accounts for 50%.

4.1.3 Attack Methods

Over the past few years, we have developed several Tibetan textual adversarial attack methods, aiming to draw attention to the NLP security in lower-resourced languages, as listed below. Our past work (Cao et al., 2023) is the only one engaged with a truly low-resource language among the research samples in the literature *NLP Security and Ethics, in the Wild* (Lent et al., TACL 2025, page 719).

TSAttacker (Cao et al., 2023). An embedding-similarity-based Tibetan textual adversarial attack. It utilizes the cosine distance between static syllable embeddings to generate substitution syllables.

TSTricker (Cao et al., 2024). A context-aware-based Tibetan textual adversarial attack. It utilizes two BERT-based masked language models with tokenizers of two different granularities to generate substitution syllables or words respectively.

TSCheater (Cao et al., 2025). A visual-similarity-based Tibetan textual adversarial attack. It utilizes a self-constructed Tibetan syllable visual similarity database to generate substitution candidates.

4.2 Whole Process

Figure 2 and Section 3 introduce the four stages of HITL-GAT. Below we use a case study on Tibetan script to illustrate the whole process, which is also demonstrated in the video and Figure 3.

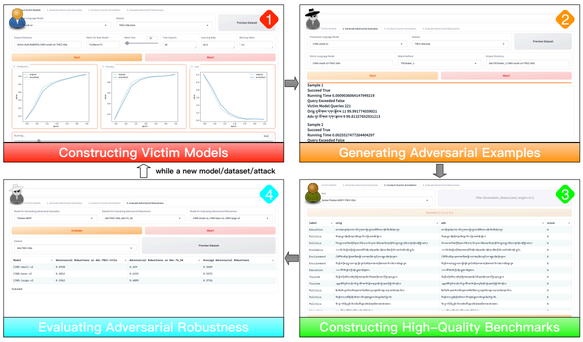
1
Constructing Victim Models
2
Generating Adversarial Examples
while a new model/dataset/attack
4
Evaluating Adversarial Robustness
3
Constructing High-Quality Benchmarks

Figure 3: Screenshots of HITL-GAT.

In the victim model construction stage, we choose the language model and downstream dataset, and then the default fine-tuning hyperparameters will be loaded. Once the “Start” button is clicked, the fine-tuning starts and the GUI displays a progress bar, metric plots (F1/macro-F1, Accuracy, and Loss) and running logs. Here, we fine-tune Tibetan-BERT and CINO series on the training set of TNCC-title and TU_SA to construct the victim language models.

Next, in the adversarial example generation stage, we choose the language model and downstream dataset, and then the victim language model will be loaded. Once the “Start” button is clicked, the attack starts and the GUI displays generated examples. Here, we implement TSAttacker, TSTricker, and TSCheater on the victim language models constructed from Tibetan-BERT upon the test set of TNCC-title and TU_SA to generate the first-round adversarial texts.

Thereafter, in the high-quality benchmark construction stage, we screen out the examples that do not satisfy the customized filter condition $levenshtein\_distance/text\_length <= 0.1$

[1] https://huggingface.co/UTibetNLP/tibetan_bert

[2] https://huggingface.co/hfl/cino-small-v2
https://huggingface.co/hfl/cino-base-v2
https://huggingface.co/hfl/cino-large-v2

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

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

from the first-round adversarial texts, and then manually annotate the remaining examples to construct the first Tibetan adversarial robustness benchmark called `AdvTS`. Given an original text and an adversarial text, we ask 3 annotators to score the human acceptance of the adversarial text based on the visual and semantic similarity between the two texts, from 1 to 5. The higher the score, the higher the human acceptance. If all annotators score the human acceptance of the adversarial text as 4 or 5, the adversarial text will be included in `AdvTS`. Below is the guidelines for human annotation.

Score 1: Definite Reject. Humans can intuitively perceive that the perturbations significantly alter the appearance or semantics of the original text.

Score 2: Reject. Humans can intuitively perceive that the perturbations do alter the appearance or semantics of the original text.

Score 3: Marginal Reject or Accept. Humans can intuitively perceive that the perturbations alter the appearance or semantics of the original text not too much.

Score 4: Accept. After careful observation or thought for 5 seconds, humans find that perturbations only slightly alter the appearance or semantics of the original text.

Score 5: Definite Accept. After careful observation for 5 seconds, humans can not find that perturbations alter the appearance of the original text. Or, after careful thought for 5 seconds, humans find that perturbations do not alter the semantics of the original text.

Finally, in the adversarial robustness evaluation stage, we utilize `AdvTS` to evaluate the adversarial robustness of `CINO` series with Equation 1. The $AdvRobust$ of `CINO-small-v2`, `CINO-base-v2`, and `CINO-large-v2` is 0.5609, 0.5572, and 0.5726 respectively.

While a new language model, downstream dataset, or textual adversarial attack method emerges, we can enter the loop again to make the adversarial robustness benchmark evolve.

5 Discussion

Due to the fact that humans perceive texts through their eyes and brains, when the perturbed text tends to the original text in visual or semantic similarity, we consider such perturbations to be imperceptible. To construct imperceptible perturbations, we can start from the following three aspects.

Transplanting existing general methods. From the perspective of semantic approximation, using synonyms for substitution is a general method. Sources of synonyms can be static word embeddings (Alzantot et al., 2018), dictionaries (Ren et al., 2019), and predictions of masked language models (Li et al., 2020).

Using intrinsic textual features. Different languages have different features inherent in their texts. For example, in abugidas (Tibetan script, Devanagari script, etc.), many pairs of confusable letters result in visually similar syllables (Kaing et al., 2024; Cao et al., 2025).

Using extrinsic encoding features. In the process of historical development, there are many cases of "same language with different encodings". For example, due to the technical problems in history, there are two Tibetan coded character sets in national standards of P.R.C (basic set: GB 16959-1997 and extension set: GB/T 20542-2006, GB/T 22238-2008); due to the simplification of Chinese characters, simplified and traditional Chinese exist. Encoding issues between different languages also deserve attention. For example, the Latin letter `x` (U+0078) and the Cyrillic letter `х` (U+0445) look the same; ZWNJ (zero width non-joiner, U+200C) is used extensively for certain prefixes, suffixes and compound words in Persian, but it is invisible and useless in most other languages.

6 Conclusion

This paper introduces `HITL-GAT`, an interactive system for human-in-the-loop generation of adversarial texts. Our approach employs a four-stage iterative loop: victim model construction, adversarial example generation, high-quality benchmark construction, and adversarial robustness evaluation. The loop ensures adversarial robustness benchmarks to co-evolve with advancements in language models, downstream datasets, and textual adversarial attack methods. Additionally, we demonstrate the utility of `HITL-GAT` through a case study on Tibetan script, employing three customized adversarial text generation methods and establishing its first adversarial robustness benchmark. Our work provides a valuable reference for other lower-resourced languages, especially languages in the Asia-Pacific that use abugidas as their writing system. The weaponization of lower-resourced languages against NLP security highlights the critical gap and the urgent need for research in this area.

Limitations

The system and the case study presented in this paper are the crystallization of our research on the adversarial robustness of Tibetan language models over the past few years. The summarized approach is only applicable to classification tasks. Given the heightened sensitivity necessary for working with lower-resourced languages, our case study is conducted on insensitive tasks with ethical best practices in mind. Due to the existing conditions of lower-resourced languages, our case study can only be conducted this far. However, this does not prevent it from serving as an early paradigm for researching the evolution of adversarial robustness benchmarks. We will continue to develop `HITL-GAT` and conduct more case studies on other minority languages.

Ethical Considerations

Our work adheres to the ACM Code of Ethics. The purpose of this paper is to promote research on NLP security, especially for lower-resourced languages. The textual adversarial attack methods mentioned in this paper must be used positively, thus preventing any malicious misuse. Additionally, adherence to the model or dataset license is mandatory when using our system or fork versions, thus preventing any potential misuse.

Acknowledgments

This work benefited greatly from the encouragement, advice, and help of others. Thank you to Heather Lent at Aalborg University for her encouragement and detailed feedback on our manuscript. Thank you to Chen Zhang at Peking University and Andong Chen at Harbin Institute of Technology for their early advice. We also thank the anonymous reviewers of *ACL for their insightful comments.

Finally, thanks to the following open-sourced projects: OpenAttack (Zeng et al., 2021), Gradio (Abid et al., 2019), LlamaFactory (Zheng et al., 2024), Transformers (Wolf et al., 2020), Datasets (Lhoest et al., 2021), etc.

This work is supported by the Science and Technology Strategic Consulting Project of the Chinese Academy of Engineering (2025-XZ-16-06), the Key Project of Xizang Natural Science Foundation (XZ202401ZR0040), the National Social Science Foundation of China (22&ZD035), the National Natural Science Foundation of China (61972436), and the MUC (Minzu University of China) Foundation (2025XYCM39).

References

Abubakar Abid, Ali Abdalla, and 4 others. 2019. Gradio: Hassle-free sharing and testing of ML models in the wild. *Preprint*, arXiv:1906.02569.

Moustafa Alzantot, Yash Sharma, and 4 others. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Yasaman Boreshban, Seyed Morteza Mirbostani, and 5 others. 2023. RobustQA: A framework for adversarial text generation analysis on question answering systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Xi Cao, Dolma Dawa, and 2 others. 2023. Pay attention to the robustness of Chinese minority language models! Syllable-level textual adversarial attack on Tibetan script. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing*.

Xi Cao, Quzong Gesang, and 3 others. 2025. TSCheater: Generating high-quality Tibetan adversarial texts via visual similarity. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Xi Cao, Nuo Qun, and 3 others. 2024. Multi-granularity Tibetan textual adversarial attack method based on masked language model. In *Companion Proceedings of the ACM Web Conference 2024*.

Jacob Devlin, Ming-Wei Chang, and 2 others. 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)*.

Di Jin, Zhijing Jin, and 2 others. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *34th AAAI Conference on Artificial Intelligence, AAAI 2020*.

Hour Kaing, Chenchen Ding, and 2 others. 2024. Robust neural machine translation for abugidas by glyph perturbation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Heather Lent. 2025. Beyond Weaponization: NLP security for medium and lower-resourced languages in their own right. *Preprint*, arXiv:2507.03473.

Heather Lent, Erick Galinkin, and 4 others. 2025. NLP security and ethics, in the wild. *Transactions of the Association for Computational Linguistics*, 13:709–743.

Quentin Lhoest, Albert Villanova del Moral, and 30 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Linyang Li, Ruotian Ma, and 3 others. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yinhan Liu, Myle Ott, and 8 others. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Preprint*, arXiv:1907.11692.

John Morris, Eli Lifland, and 4 others. 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*.

Yixin Nie, Adina Williams, and 4 others. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Hao Peng, Shixin Guo, and 6 others. 2024. TextCheater: A query-efficient textual adversarial attack in the hard-label setting. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3901–3916.

Nuo Qun, Xing Li, and 2 others. 2017. End-to-end neural text classification for Tibetan. In *Proceedings of the 16th Chinese National Conference on Computational Linguistics*.

Shuhuai Ren, Yihe Deng, and 2 others. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Walter Simoncini and Gerasimos Spanakis. 2021. SeqAttack: On adversarial attacks for named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Eric Wallace, Pedro Rodriguez, and 3 others. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Alex Wang, Amanpreet Singh, and 4 others. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Boxin Wang, Chejian Xu, and 3 others. 2022. SemAttack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*.

Boxin Wang, Chejian Xu, and 6 others. 2021a. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 2021*.

Jindong Wang, Xixu Hu, and 11 others. 2023. On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

Zijie J. Wang, Dongjin Choi, and 2 others. 2021b. Putting humans in the natural language processing loop: A survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*.

Thomas Wolf, Lysandre Debut, and 20 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Ziqing Yang, Zihang Xu, and 5 others. 2022. CINO: A Chinese minority pre-trained language model. In *Proceedings of the 29th International Conference on Computational Linguistics*.

Haneul Yoo, Yongjin Yang, and Hwaran Lee. 2025. Code-switching red-teaming: LLM evaluation for safety and multilingual understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Guoyang Zeng, Fanchao Qi, and 7 others. 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*.

Jiangyan Zhang, Kazhuo Deji, and 3 others. 2022. Research and application of Tibetan pre-training language model based on BERT. In *Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics*.

Tianyi Zhang, Varsha Kishore, and 3 others. 2020. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020*.

Yaowei Zheng, Richong Zhang, and 3 others. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*.

Yulei Zhu, Kazhuo Deji, and 2 others. 2023. Sentiment analysis of Tibetan short texts based on graphical neural networks and pre-training models. *Journal of Chinese Information Processing*, 37(2):71–79.