Effective Homoglyph Attack Restoration by Utilizing Character-wise Language Model

효과적인 동형 문자 공격 복원을 위한 문자 관점 언어 모델 활용

Hanyong Lee

glhy0718@gmail.com

26th Oct. 2023

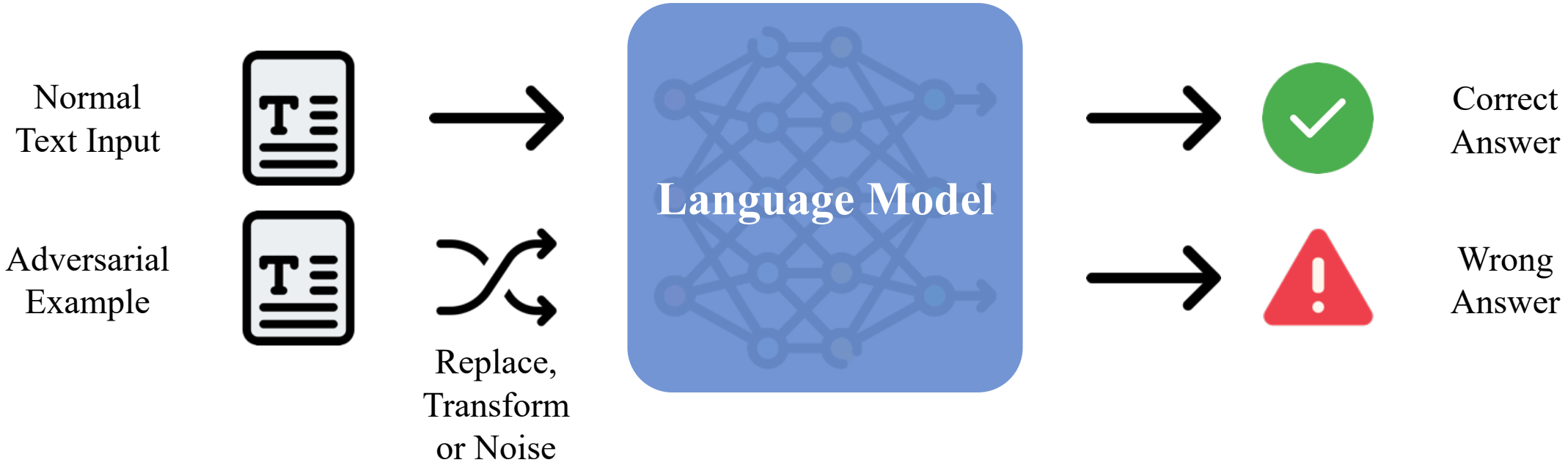
Contents

Introduction

NLP Attack

Adversarial examples in **Natural Language Processing**

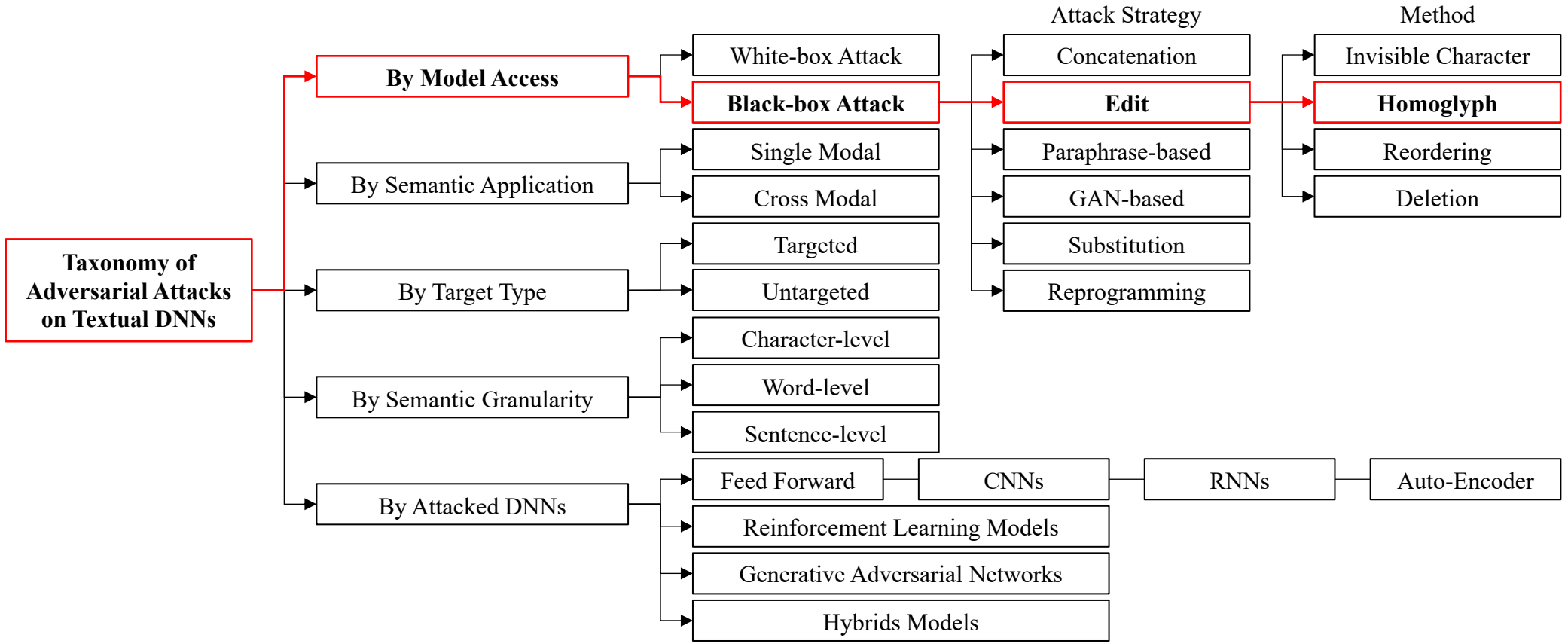
- Inspired by the adversarial examples in computer vision and deep neural networks.
- Various methods have been proposed to attack NLP applications.



Introduction

NLP Attack & Homoglyph Attack

A taxonomy of the adversarial examples in **Natural Language Processing**



Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Transactions on Intelligent Systems and Technology (TIST), 11(3), 1-41.

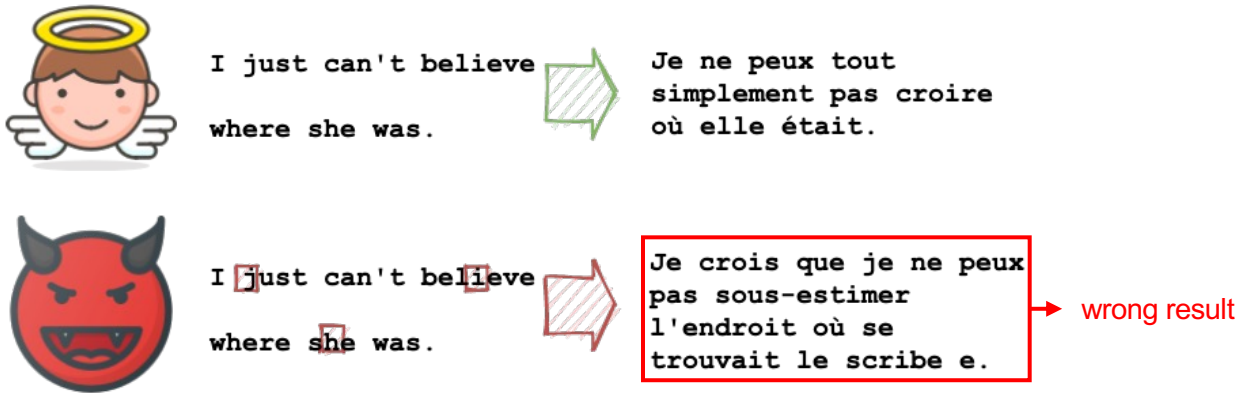
Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters: Imperceptible nlp attacks. In 2022 *IEEE Symposium on Security and Privacy (SP)* (pp. 1987-2004). IEEE.

Introduction

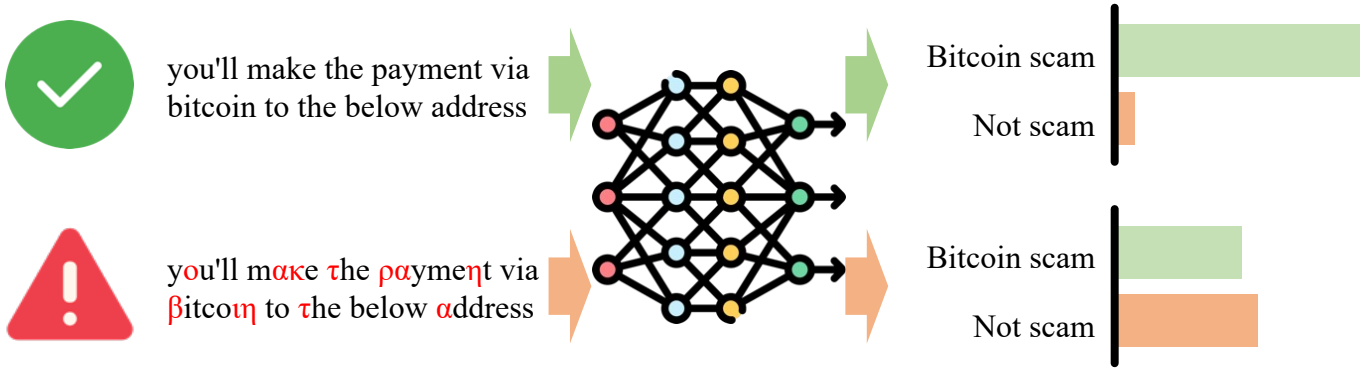
Homoglyph Attack

Example of an attack using homoglyph to confuse NLP tasks.

Homoglyph attack on the translation *



Homoglyph attack on the classification



* Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters: Imperceptible nlp attacks. In 2022 *IEEE Symposium on Security and Privacy (SP)* (pp. 1987-2004). IEEE.

Related Work

Problem Formulation

- Assume that the notations of:
  - a homoglyph character c_h
  - a restored character c_a
  - a homoglyph perturbed sentence $S = \{c^{(1)}, \ldots, c^{(n)}\}$
- Restore the homoglyph into the most probable alphabetic character.
  - $c_a^{(i)} = \text{argmax}_{c^{(i)}} P(c_h^{(i)}, S)$

$S =$ you'll mαke τhe ραymeηt via βitcoιη to τhe below αddress

o

$c_h^{(i)}$

$c_a^{(i)} = f\left(c_h^{(i)}, S\right) = \text{argmax}_{c^{(i)}} P\left(c^{(i)} \middle| o\right)$

you'll make the payment via bitcoin to the below address

Defense Methods of Homoglyph Attack

SimChar database

H. Suzuki et al. used the following formula to calculate the similarity of two bitmap images of characters.

$$\Delta = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_1(i,j) - I_2(i,j)|$$

pixel of image I_1, I_2 at coordinate (i, j)

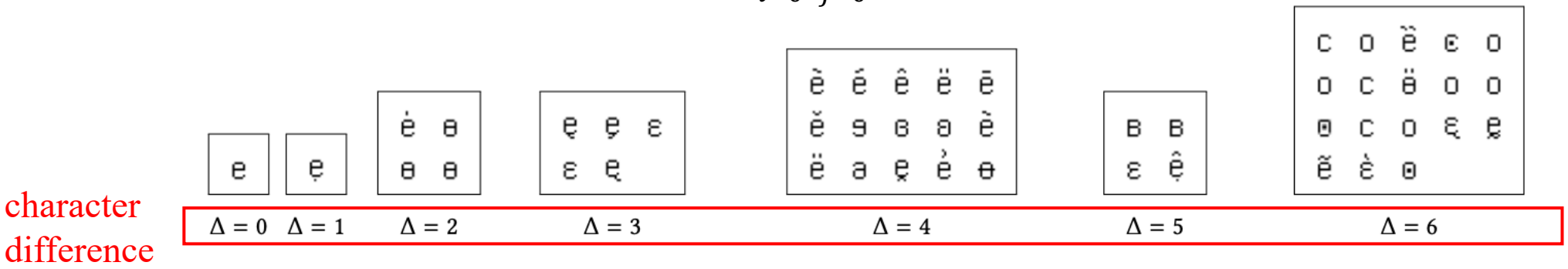


Figure 6: Basic Latin lowercase letter 'e' and characters under different values of the threshold Δ. In this work, characters with Δ ≤ 4 are detected as homoglyphs.

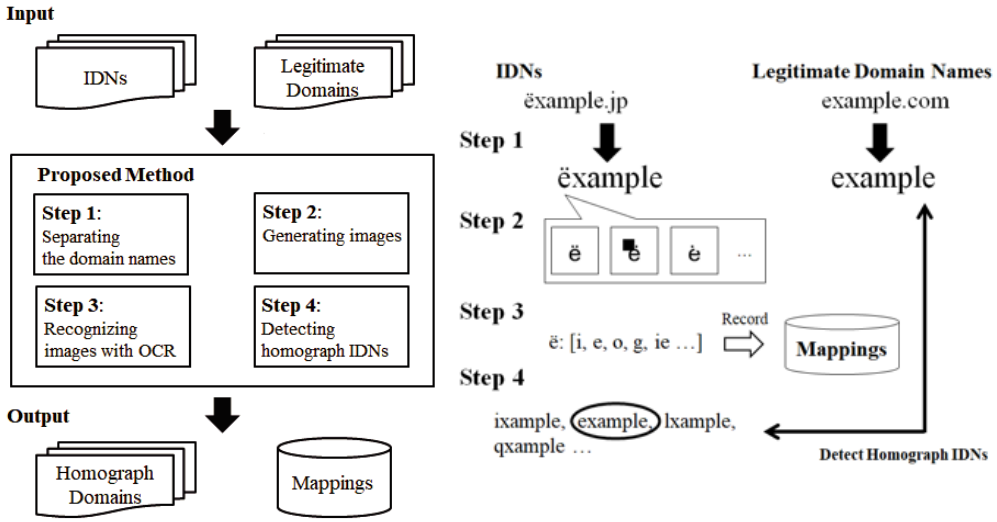
Suzuki, H., Chiba, D., Yoneya, Y., Mori, T., & Goto, S. (2019, October). ShamFinder: An automated framework for detecting IDN homographs. *In Proceedings of the Internet Measurement Conference* (pp. 449-462).

Related Work

Defense Methods of Homoglyph Attack

OCR-based method

Using OCR to restore the homoglyph to ASCII character.



Spellchecker-based method

Using Spellchecker to find the most similar words of homoglyph-perturbed words.

Table 2
Some of the adversarial text examples used in experiments.

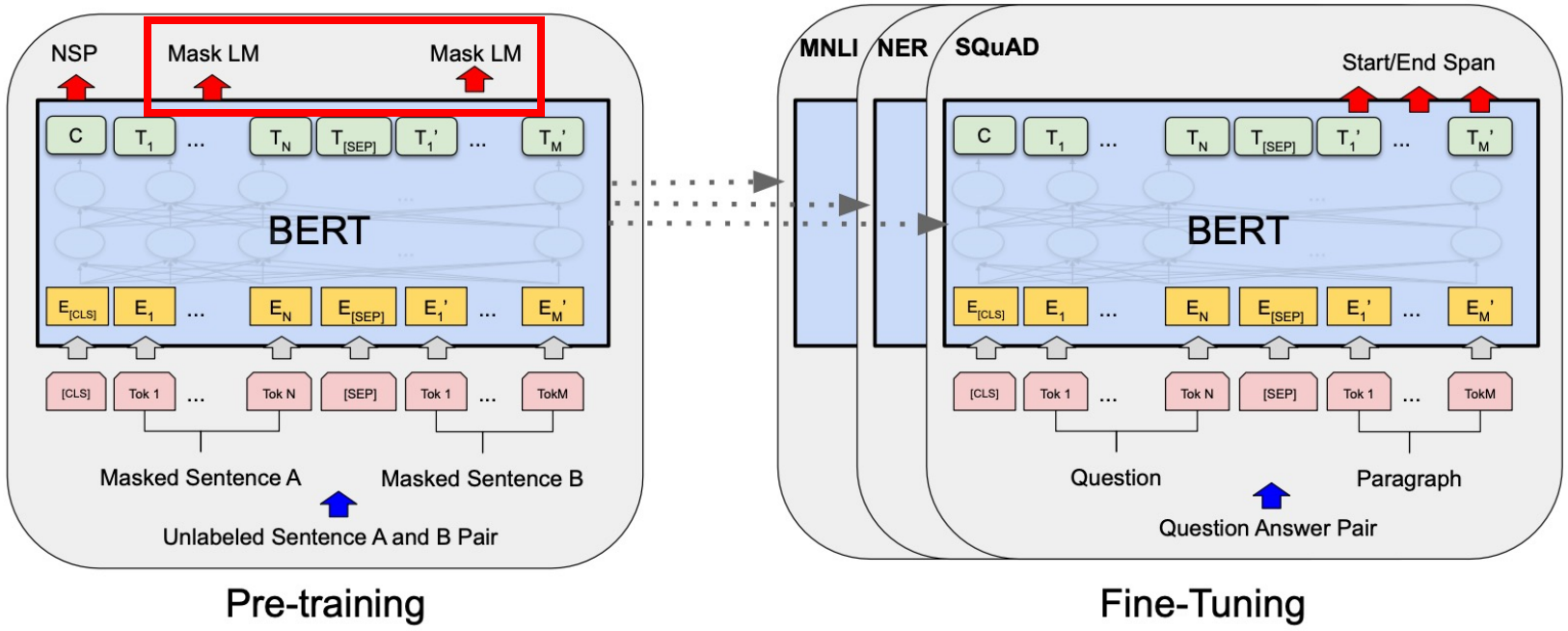
Original	Flipping	Swapping	Deletion
Busy	Bu$y	Bsuy	Bsy
Caller	C@1ler	Claler	Cller
Reply	Rep1y	Rpely	Rply
winner	w!nner	wniner	wnner

Sawabe, Y., Chiba, D., Akiyama, M., & Goto, S. (2019). Detection method of homograph internationalized domain names with OCR. *Journal of Information Processing*, *27*, 536-544.

Imam, N. H., Vassilakis, V. G., & Kolovos, D. (2022). OCR post-correction for detecting adversarial text images. *Journal of Information Security and Applications*, *66*, 103170.

BERT Masked Language Model

BERT using Masked Language Model pre-training to learn the randomly masked tokens with contextual embedding.



Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Proceedings of NAACL-HLT. 2019.

Related Work

Pros & Cons of Related Works

Algorithm	PROS	CONS
BERT	• Slightly robust on new homoglyphs	• Pre-raining the model takes a long time. • Subword-based tokenization is not helpful for the task.
OCR	• Robust on new homoglyphs • Automated workflow	• OCR is not performing well on homoglyphs
SimChar DB	• Automated workflow on finding homoglyph pairs	• Not robust on new homoglyphs • Must manually add homoglyph pairs to the dictio nary • Inability when the input & output lengths are diff erent
Spell Checker	• Robust on new homoglyphs	• Must manually add words to the dictionary

Common Drawbacks of Conventional Methods

Contextual Information is Important

$S =$ you'll mακe τhe ραymeηt via βitcoιη to τhe below αddress

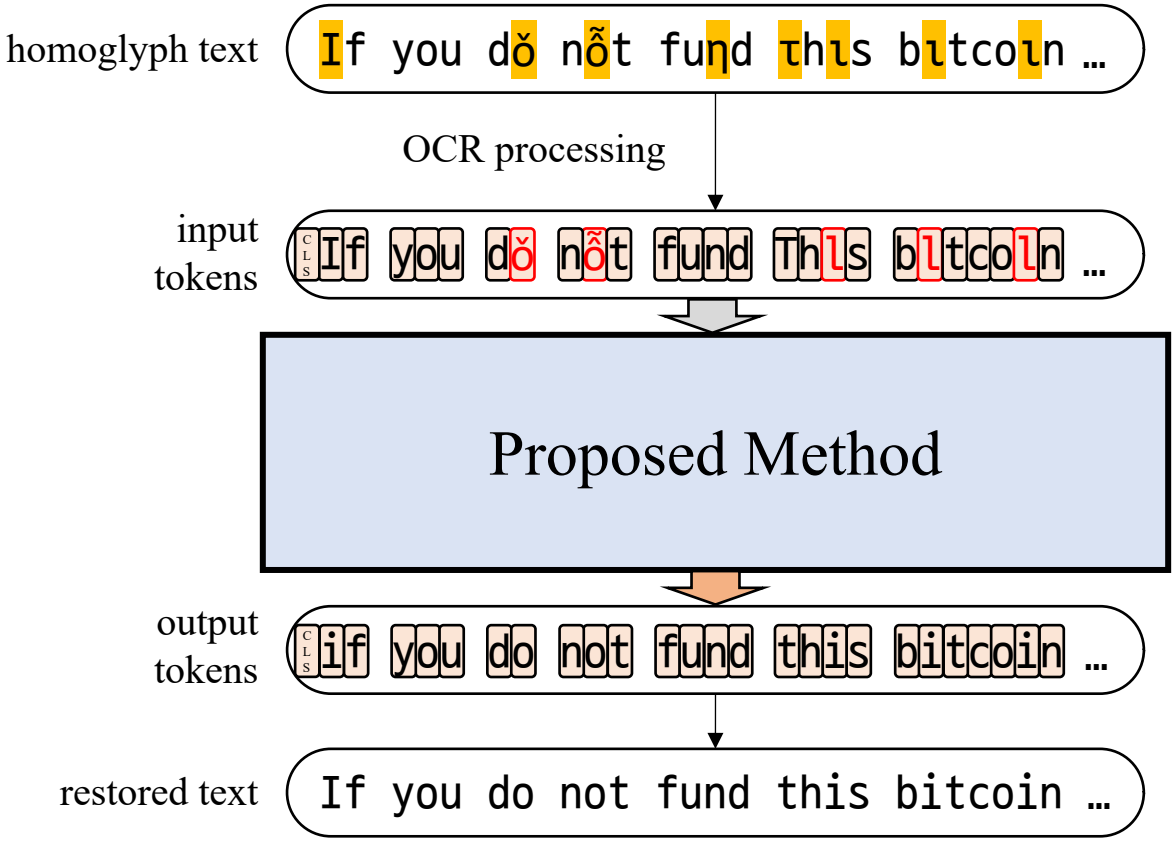
$c_h^{(i)}$ → $c_a^{(i)}$

you'll make the payment via bitcoin to the below address

Method	Word Accuracy	Restoration Process	Context Range	Unit of Context
BERT	67.13% ± 0.29	$c_a^{(i)} \in t_a^{(i)} = \text{argmax}_{t^{(i)}} P\left(t^{(i)} \middle\| t^{(1)}, \ldots, t^{(i-1)}, t^{(i+1)}, \ldots, t^{(n)}\right) \approx f_{BERT}\left(t_h^{(i)}, S\right)$ $t^{(i)}$: i-th token, n : number of tokens	Sentence	Subword
OCR	65.18% ± 0.28	$c_a^{(i)} = f_{OCR}\left(c_h^{(i)}\right)$	-	-
SimChar DB	55.12% ± 0.46	$c_a^{(i)} = f_{SimChar\_DB}\left(c_h^{(i)}\right)$	-	-
Spell Checker	**90.87**% ± 0.11	$c_a^{(i)} \in w_a^{(i)} = \text{argmin}_{w^{(i)}} D_{edit\_dist}(w^{(i)}, w_h^{(i)}) = f_{Spell\_Checker}\left(w_h^{(i)}\right)$ $w^{(i)}$: i-th word	Word	Character

Proposed Method

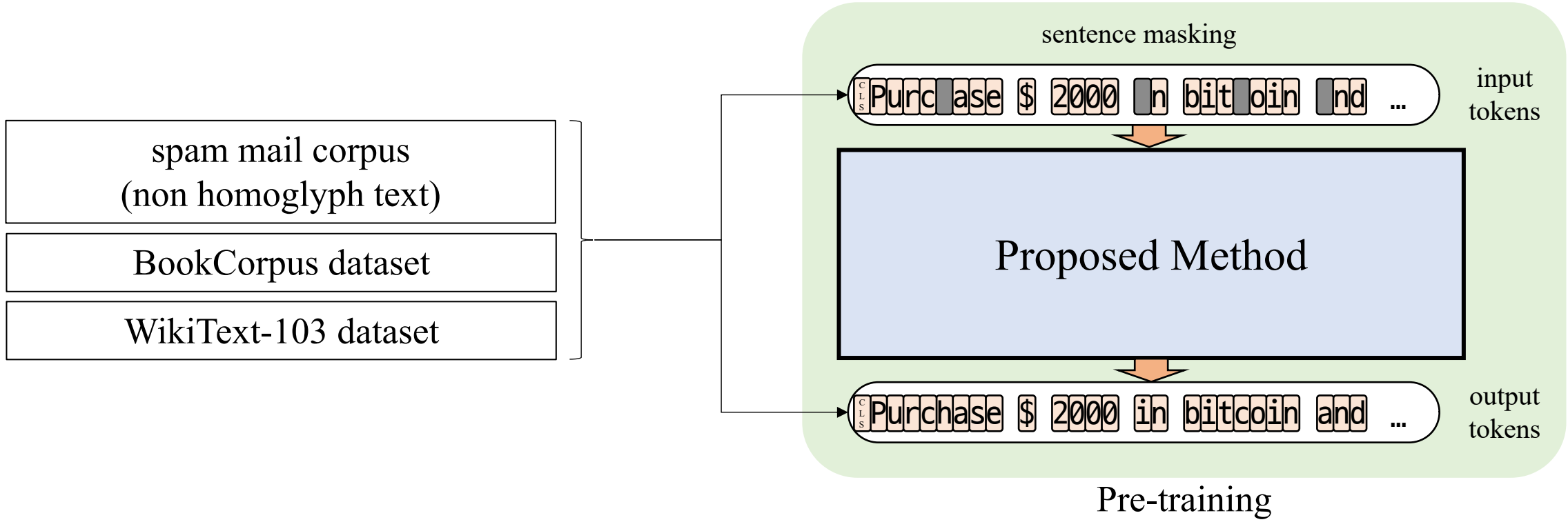
Restoring Process Overview



Proposed Method

Pre-training (Zero-Shot)

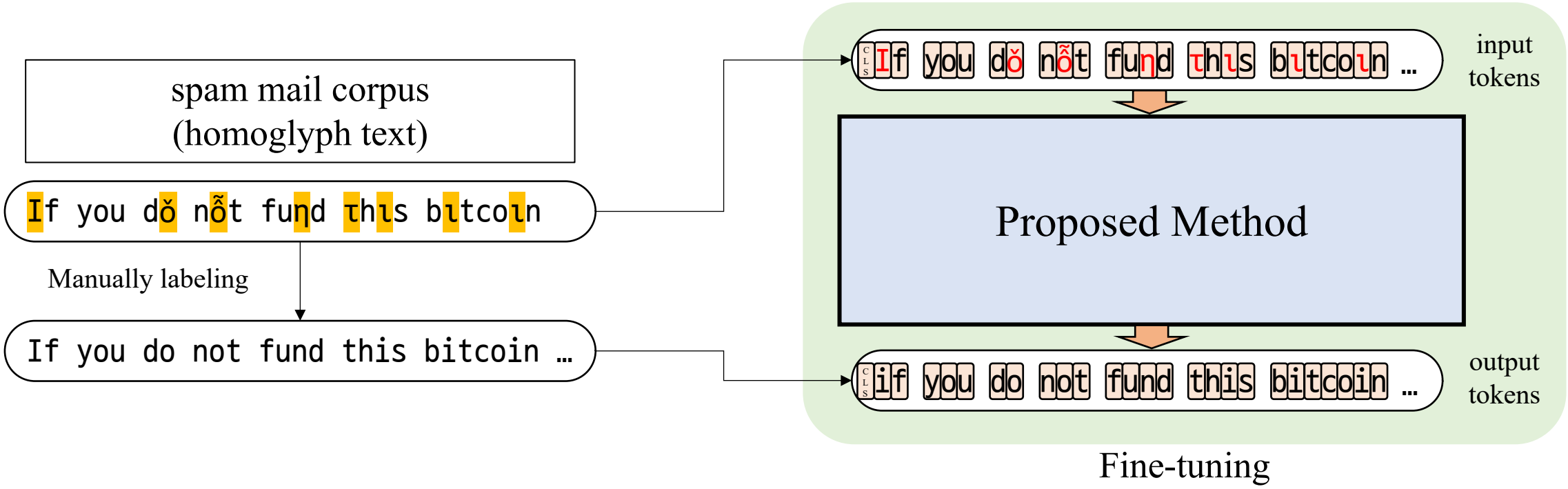
We pretrained the proposed model as a masked language model (MLM).



Proposed Method

Fine-tuning

Evaluated with a blind test (Train/Test set split ratio = 8:2 / Iteration = 50 times)



Experimental Settings

Dataset

		# of Examples		# of Tokens	
	Dataset	Train Set	Validation Set	Train Set	Validation Set
Pre-training (Zero-Shot) dataset	WikiText-103	1,165,029	2,461	285,324,545	612,967
	BookCorpus (about 2% of the total dataset)	1,480,085	3,126	79,346,399	143,939
	bitcoinabuse.com dataset (non-homoglyph)	299,239	633	23,152,956	54,680
	Total dataset	2,944,353	6,220	387,823,900	811,586
Finetuning dataset	bitcoinabuse.com dataset (homoglyph)	21,342	5,336	random split (80%)	random split (20%)

Overall Experimental Results

Fine-tuned Model Evaluation - Accuracy

Experimental Settings	
Train / Validation split	8 : 2
# of Test Iteration	50
Used Datasets	• bitcoinabuse.com dataset (homoglyph sentences)
Metric	Word Accuracy

Experimental Results	
Method	Word Accuracy
Proposed	**99.59%** ± 0.08
BERT	67.13% ± 0.29 ▼
OCR	65.18% ± 0.28 ▼
SimChar DB	55.12% ± 0.46 ▼
Spell Checker	90.87% ± 0.11 ▼

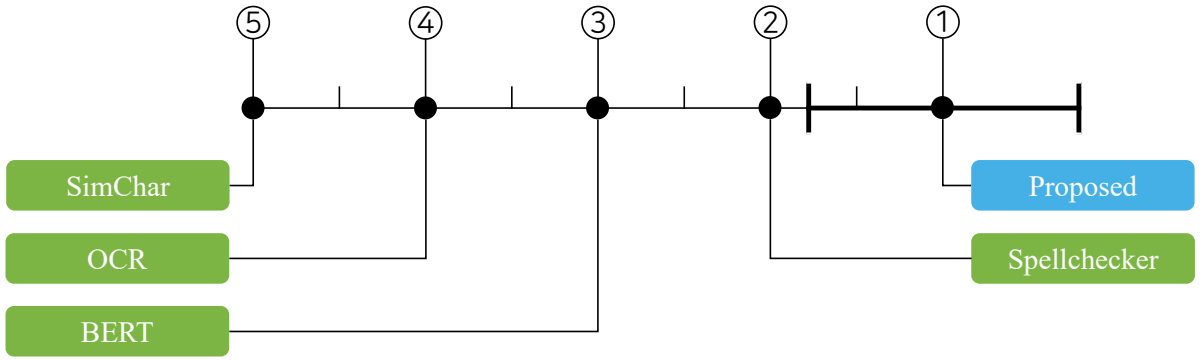
Word level accuracy and T-test with the accuracy of the proposed method
▼ means that the algorithm has a significantly lower performance than the proposed algorithm based on the t-test.
▼: $p < 0.01$

Overall Experimental Results

Fine-tuned Model Evaluation – Friedman test

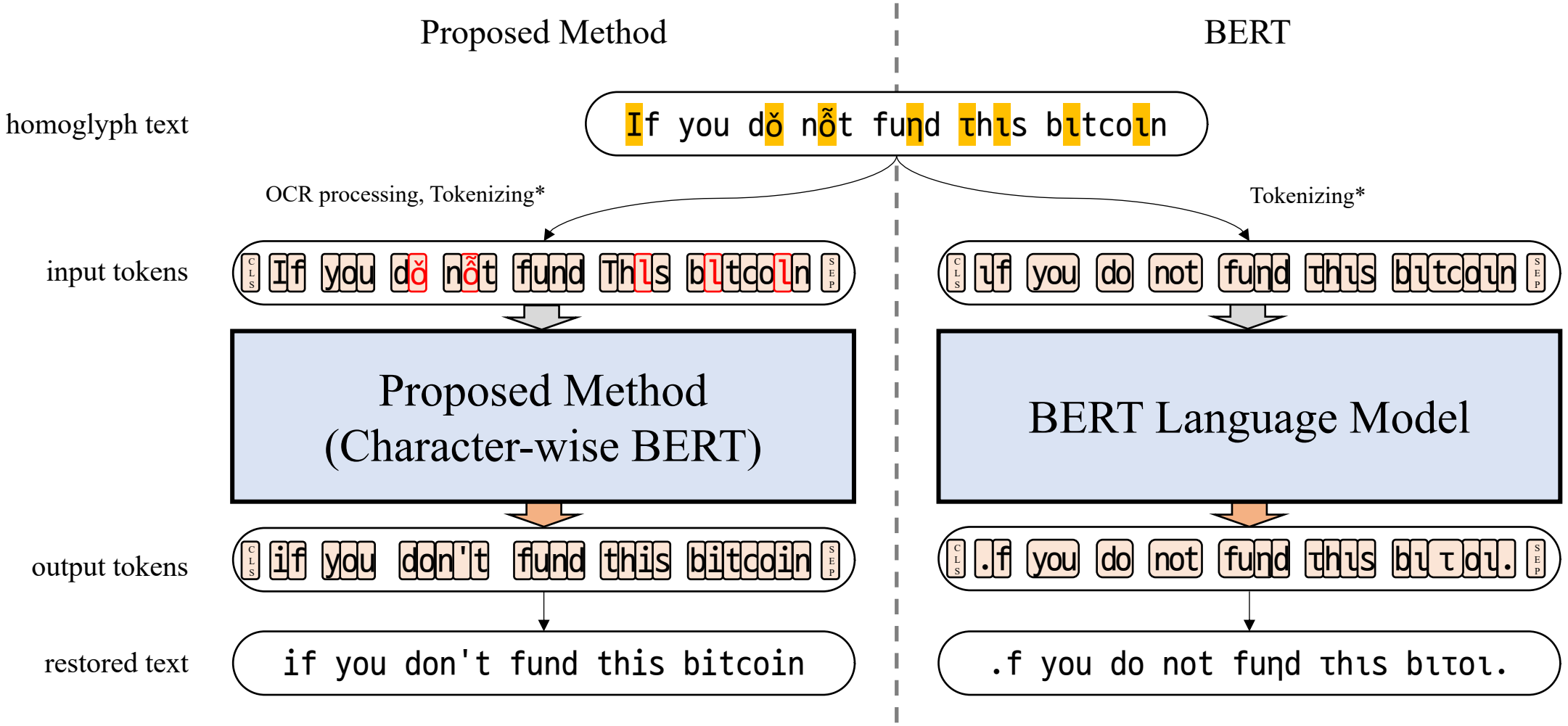
Statistic	p value
100.0	2.3643×10^{-5}

Fine-tuned Model Evaluation – Bonferroni–Dunn test (α = 0.05, CD = 0.7899)



In-Depth Analysis

Analysis – A comparison of the proposed method and BERT



* Tokenizing has a Unicode normalization process.

In-Depth Analysis

Analysis – Qualitative analysis

- Since the BERT model **Unicode normalization** is being processed **during tokenization**, it is partially effective for restoration but not predefined homoglyphs.
- BERT's sub-word tokenization is basically not suitable for homoglyph restoration because one token contains one or more characters.

Original Text	Ground Truth	Proposed Method	BERT model
ī am well aware is yōūr pãss.	i am well aware is your pass.	i am well aware is your pass.	l am well as is your pass.
you have 48 hours to make the paym ent.	you have 48 hours to make the paym ent.	you have 48 hours to make the paym ent.	you have 48 hours to make the paym ent.
I have α unιque pixel wiτhiη τhιs em αιl message	i have a unique pixel within this ema il message	i have a unique pixel within this ema il message	i have a unlqueid with the thls emall message
all íпgeиious is sìмple.	all ingenious is simple.	all ingenious is simple.	all ingenlous ls simple.
you'll mακe τhe ραymeηt via βitcoιη to τhe below αddress	you'll make the payment via bitcoin t o the below address	you'll make the payment via bitcoin t o the below address	you. ll make the paymeat via bitcan t o the above address

- homoglyph
- wrong word (token)

Analysis – Ablation study (Zero-Shot Setting)

- An ablation study was conducted on the OCR preprocessing module and character-level BERT in the proposed method.
- In a zero-shot setting, an ablation study showed that eliminating OCR preprocessing led to only a 14% decrease in performance.
- This confirms that, in terms of performance in the proposed method, character-wise BERT has a greater impact than OCR.

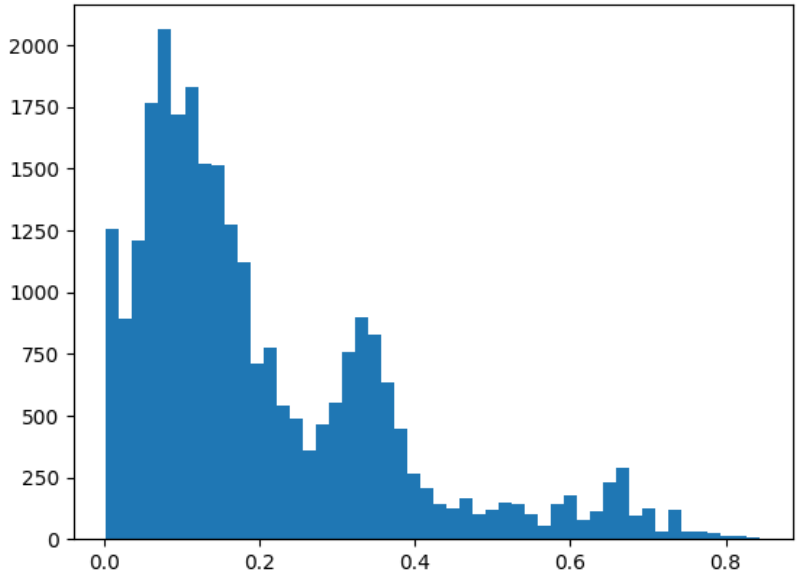
Method	Word Accuracy
Proposed w/ OCR preprocessing	**95.16%** ± 12.89
Proposed w/o OCR preprocessing	81.28% ± 29.19
OCR	65.19% ± 29.76

In-Depth Analysis

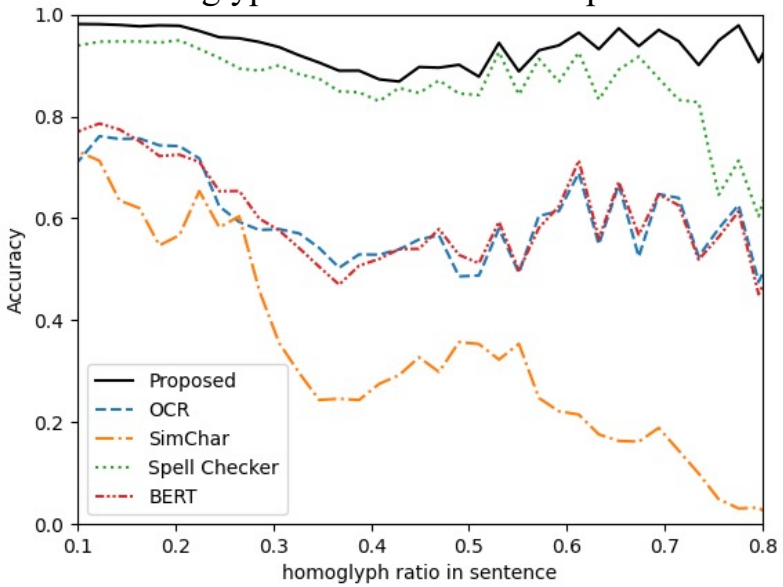
Analysis – Performances based on the ratio of homoglyphs

- In the bitcoinabuse[.]com dataset, most of the sentences had low homoglyph ratios.
- Regardless, the proposed method demonstrated robust results even at high homoglyph ratios.
- The character-level BERT excels at restoring sentences with low homoglyph ratios, but its performance declines as th e homoglyph ratio increases, reinforcing the need for the OCR module to preserve context during homoglyph attacks.

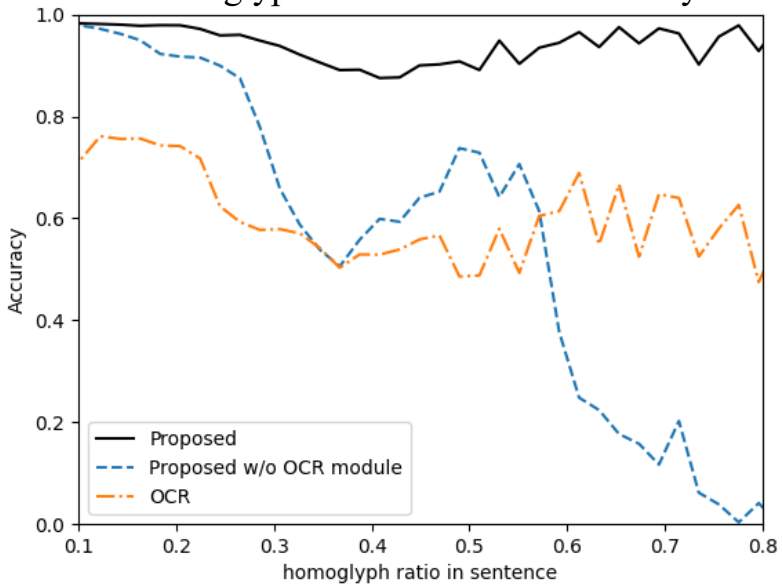
Number of samples
according to the homoglyph ratio



Performance graph based on the
homoglyph ratio in the main experiment

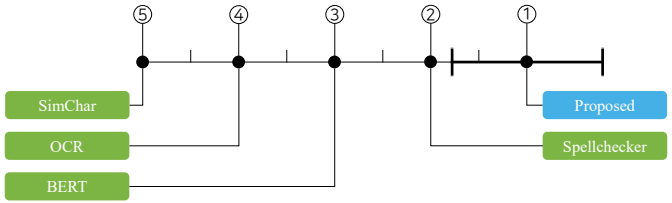


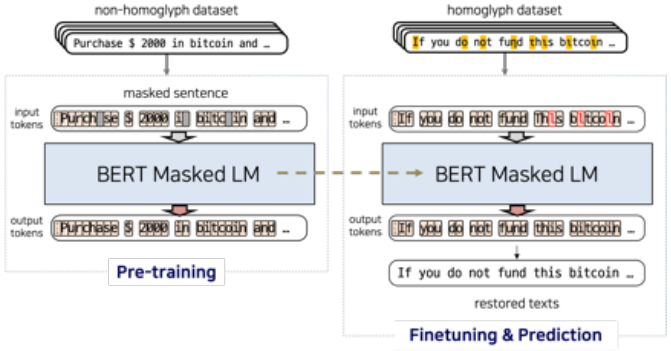
Performance graph based on the
homoglyph ratio in the ablation study



Conclusion

Contribution of the Study





1. We achieve **higher performance than traditional methods** such as OCR, SimChar DataBase, and spell checker.

2. We propose the **Character-level BERT algorithm** and introduce a dataset for the homoglyph restoration task.

3. Limitations of the existing methods were found, and in-depth analysis was conducted through **statistical verification through the t-test, Freidman test, and Bonferroni–Dunn test** to determine that the proposed model performed better than the existing model.

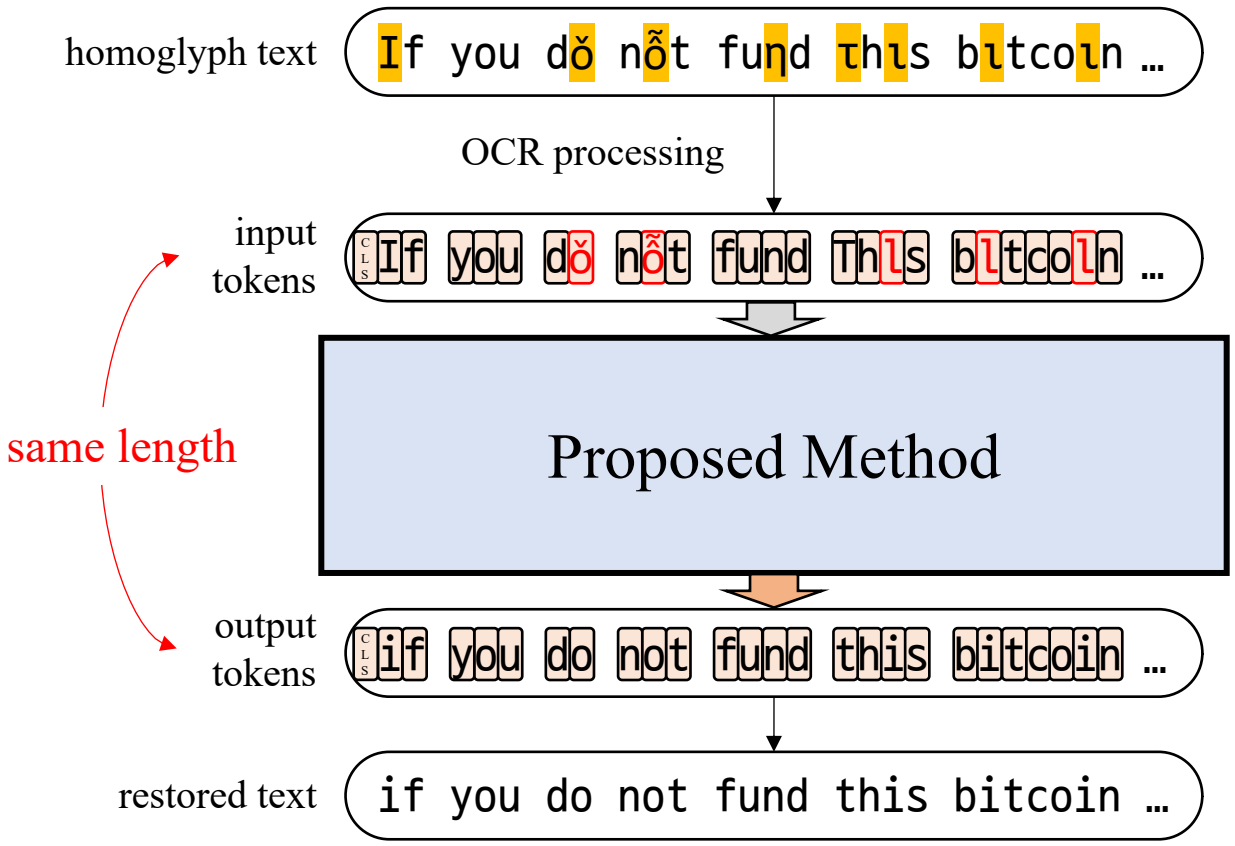
Conclusion

Limitation of Proposed Model

The proposed model is an encoder-only model, limited in that the input/output sentences should be the **same sequence length**.

- It can be used as a homoglyph of the alphabet ‘m’ by concatenating the letters ‘r’ and ‘n’. B ut the presented model does not solve this cas e.

The proposed model can be applied as a **sequence-to-sequence model** in future studies.



References

1. Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Transactions on Intelligent Systems and Technology (TIST), 11(3), 1-41.
2. Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters: Imperceptible nlp attacks. In 2022 IEEE Symposium on Security and Privacy (SP) (pp. 1987-2004). IEEE.
3. Unicode Consortium, "Unicode Utilities: Confusables." unicode.org. https://www.unicode.org/Public/security/15.0.0/confusables.txt (accessed 10 September 2022).
4. Suzuki, H., Chiba, D., Yoneya, Y., Mori, T., & Goto, S. (2019, October). ShamFinder: An automated framework for detecting IDN homographs. In Proceedings of the Internet Measurement Conference (pp. 449-462).
5. Sawabe, Y., Chiba, D., Akiyama, M., & Goto, S. (2019). Detection method of homograph internationalized domain names with OCR. Journal of Information Processing, 27, 536-544.
6. Imam, N. H., Vassilakis, V. G., & Kolovos, D. (2022). OCR post-correction for detecting adversarial text images. Journal of Information Security and Applications, 66, 103170.
7. Norvig, P., "How to write a spelling corrector." norvig.com https://norvig.com/spell-correct.html (accessed 10 September 2022).
8. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
9. Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016). Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843.
10. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision (pp. 19-27).

Thank you

Appendix

A. Model Parameters

Parameter	Value
Input Max Sequence	512
Vocab Size	881
# of Attention Heads	12
# of Hidden Layers	12
# of Model Parameters	86,720,369

Experimental Settings

B. Pre-training (Zero-Shot) Setting

Parameter	Value
# of Epochs	25
Train / Validation split	refer the slide 14.
Used Datasets	• WikiText-103 • BookCorpus (about 2% of the total dataset) • bitcoinabuse.com dataset (non-homoglyph sentences)
Metric	Accuracy

Appendix

C. Pre-trained Model Evaluation (Zero-Shot)

Evaluated with the whole test sentences
Character level BERT is only pretrained with the WikiText-103, BookCorpus & spam mail datasets.

Experimental Settings

Parameter	Value
# of Epochs	25
Train / Validation split	refer the slide 14.
Used Datasets	• WikiText-103 • BookCorpus (about 2% of the total dataset) • bitcoinabuse.com dataset (non-homoglyph sentences)
Metric	Accuracy

Experimental Results

Algorithm	Word Accuracy
Proposed (Zero-Shot)	**95.16%** ± 12.89
Normal BERT	67.13% ± 26.79 ▼
OCR	65.19% ± 29.76 ▼
SimChar DB	55.06% ± 36.27 ▼
Spell Checker	90.90% ± 16.20 ▼

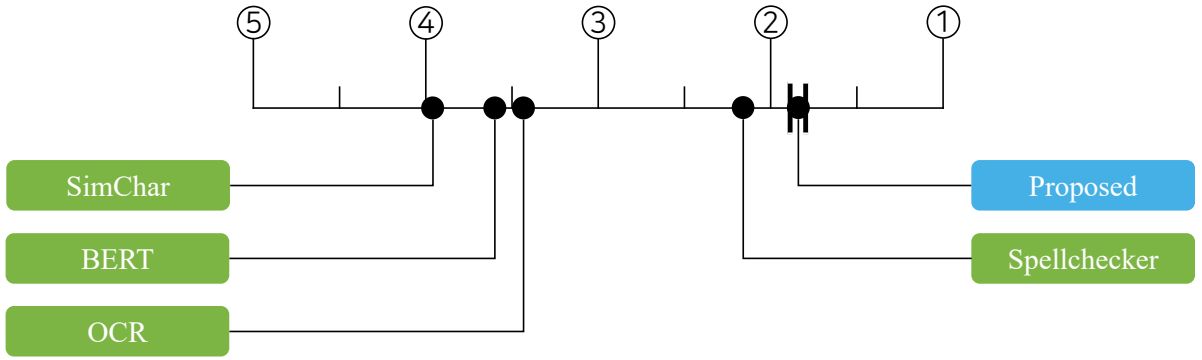
Word level accuracy and T-test with the accuracy of the proposed method
▼ means that the algorithm has a significantly lower performance than the proposed algorithm based on the t-test.
▼: $p < 0.01$

Appendix

D. Pre-trained Model Evaluation – Friedman test

Statistic	p value
50886.18	$0 < p < 2.2204 \times 10^{-16}$

E. Pre-trained Model Evaluation – Bonferroni–Dunn test (α = 0.05, CD = 0.0342)



Appendix

F. Test Output Examples of Pretrained Model

Example 1

- input sentence L Need ɗ y0ᑌᖇ TOTɑı ɑτTeNTıOɴ fOᴙ The c0MıNg TweNτʏ-fOᑌR hᖇ□,
- output sentence : i need your total attention for the coming twenty - four hrs,

Example 2

- input sentence : ɑftêr thät, i hȧvë ṡtȧrtèd trȧcкiṅġ ẏȯuг ṅtérпët ȧctívítìêṡ.
- output sentence : after that, i have started tracking your internet activities.

Appendix

F. Test Output Examples of Pretrained Model

Example 3

- input sentence : do ṅȯt trɣ tȯ coṅtáct the ṗolićê àɴd ȯtɳêr sëcurity servıces.
- output sentence : don't try to contact the police and other security services.

Example 4

- input sentence : yøuř åccøuňt wås åttåckěd.
- output sentence : yogr account was attacked.