C2C: A Framework for Critical Token Classification in Transformer-based Inference Systems

Myeongjae Jang, Jesung Kim, Haejin Nam, Sihyun Kim, Soontae Kim
School of Computing, *KAIST*, Daejeon, Republic of Korea
{myeongjae0409, jesung.kim, haejinnam, sihyun.kim, kims}@kaist.ac.kr

***Abstract*—Because embedding vectors in a Transformer-based model represent crucial information about input texts, attacks or errors affecting them can cause severe accuracy degradation. We observe critical tokens for the first time, that determine the overall accuracy but their embedding vectors take only a small portion of the embedding table. Therefore, we propose a framework called C2C that classifies the critical tokens to facilitate their protection in a Transformer-based inference system with a small overhead. Using BERT with GLUE datasets, critical embedding vectors take only 13.8% of the embedding table. Compromising critical embedding vectors can reduce accuracy by up to 44.8% even if other parameters are not corrupted.**

***Index Terms*—Transformer, Embedding Vectors, Security**

I. Introduction

A Transformer-based inference system must be secure against malicious attacks to ensure a trustworthy computing environment. Unfortunately, protecting the entire model is difficult because of its large size. Instead, we can reduce performance degradation by applying differential security levels according to data sensitivity when we utilize the typical memory protection schemes on Transformer-based models.

We observed for the first time that overall accuracy is determined by only a small amount of tokens, whose embedding vectors take a small portion of the embedding table. We define these tokens and embedding vectors as **"critical tokens"** and **"critical embedding vectors (CEVs)"**. They are naturally created during training with an attention mechanism [1]. The attention mechanism strengthens some tokens that represent the core information of the input texts, enhancing their effects on the inference result. Based on this observation, we propose C2C, an efficient critical token classification framework. C2C classifies critical tokens by profiling a few sampled input texts. While profiling, C2C intentionally injects errors into embedding vectors and measures their criticality according to changes in output logit values that affect inference failure and accuracy degradation. C2C also acquires additional critical tokens using vector similarity even if they are not included in the sampled input texts. We conduct C2C with BERT [2] and nine GLUE datasets [3]. C2C classifies CEVs in an hour. CEVs account for only 13.8% of the embedding table. However, attacking CEVs reduces accuracy by 44.8% on average even if other parameters in BERT are not corrupted.

II. Observation of Critical Tokens

Fig. 1 shows the portions of accessed tokens in BERT with nine GLUE task datasets. The tokens used in the validation set account for 21.2% of the embedding table on average. The

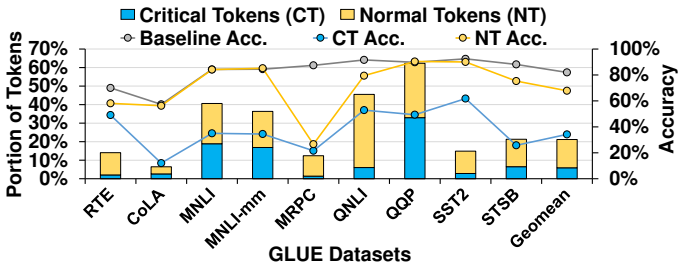

Fig. 1. Portions of tokens and accuracy degradation in BERT.

QQP uses the most tokens with 62.3%. Less than half of the tokens are never accessed during inference.

Next, we found for the first time through systematic observation that only a small number of critical tokens among accessed tokens primarily determine inference results. Semantically, articles and prepositions are frequently used but do not severely affect the inference results compared to other tokens, such as nouns and verbs. Systematically, BERT is a basic Transformer-based model that uses an attention mechanism [1]. When one embedding vector is more important than others, the attention mechanism amplifies its influence. This attention-weighted embedding vector becomes a CEV. We distinguish tokens as critical and normal tokens. A "normal token" means a token found in the input texts is less important to inference than a critical token. As shown in Fig. 1, only 5.9% of tokens in the embedding table are critical tokens on average.

Fig. 1 also shows accuracy drops when CEVs are corrupted to zero vectors. To intensively observe the effect of CEVs on accuracy, we simulate an extreme scenario where we corrupt CEVs and the embedding vectors for normal tokens but leave other weights and parameters unmodified. The baseline accuracy is 82.0% but decreases to 34.2% when CEVs are corrupted, whereas corrupting embedding vectors for normal tokens results in 67.8% accuracy, on average. Even though the number of critical tokens is less than half of normal tokens, their effect on accuracy is 4.0 times. MRPC shows similar accuracy, but the number of tokens differs by 7.7 times.

III. C2C: Critical Token Classification

C2C in Fig. 2 is designed on the basic Transformer-based inference system. It receives a profiling dataset, the pre-trained model, and user configurations. It mainly consists of two phases: critical token profiling and expansion phases. After profiling and classification, Protected Critical Tokens (PCTs) are managed as a list and served for applications.

978-3-9826741-0-0/DATE25/© 2025 EDAA

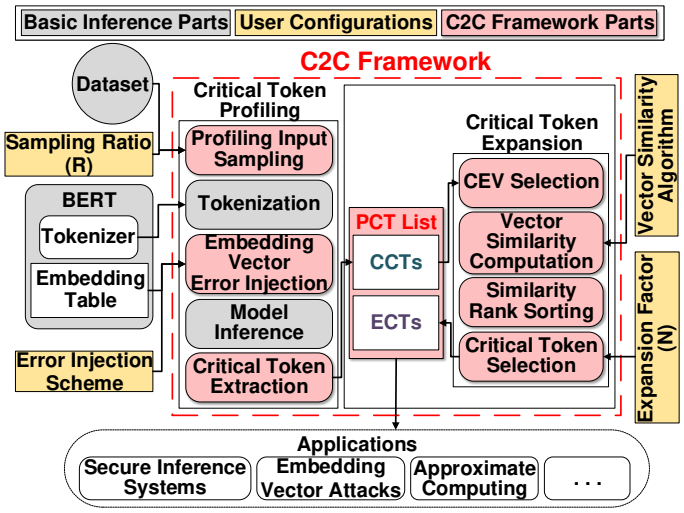

Fig. 2. C2C framework.

The profiling phase proceeds as follows. ① C2C randomly samples profiling inputs with sampling ratio **R**. For each sample input, C2C conducts a normal inference and temporarily stores the output logit values as $\{BaseLogitValue\}$. ② A tokenizer tokenizes the sample input. For each token, ③ C2C intentionally injects errors into the embedding vector using the user-configured error injection scheme. ④ C2C infers again using the erroneous embedding vector. ⑤ If the inference fails, C2C stores the selected token as a Classified Critical Token (CCT) and computes its criticality as a **"critical value"** calculated by $abs|\{CurrentLogitValue\} - \{BaseLogitValue\}|$. C2C repeats STEPS ③ - ⑤ for every non-duplicate token in the sample input. Finally, C2C arranges the CCT list by critical values to determine the priority of criticality.

The expansion phase proceeds as follows. ① C2C selects one CEV for a CCT and ② computes the vector similarity with other embedding vectors not classified as CEVs. The user can apply any vector similarity algorithms or combine multiple algorithms by considering the algorithmic complexity and increased correctness of the critical token expansion. Next, ③ vector similarity results are sorted and ④ C2C selects the top **N** tokens as Expanded Critical Tokens (ECTs). The user can configure **N** as the expansion factor.

C2C can be utilized in various applications. The first is a secure inference system with a small protection overhead. C2C also can be used to design effective embedding vector attacks. If the system focuses on normal tokens, C2C can help improve a low-power or approximate computing system by adopting energy-saving schemes for normal tokens.

IV. Experimental Results

Fig. 3 shows the profiling time and the number of PCTs. Since we conduct critical token classification for each dataset with several combinations of sampling ratios and expansion factors, several results can be obtained from one dataset. The average profiling time is 68 minutes. The worst is 583 minutes for the QNLI dataset but can be reduced to 199 minutes by changing configurations. The average number of PCTs is 4201.

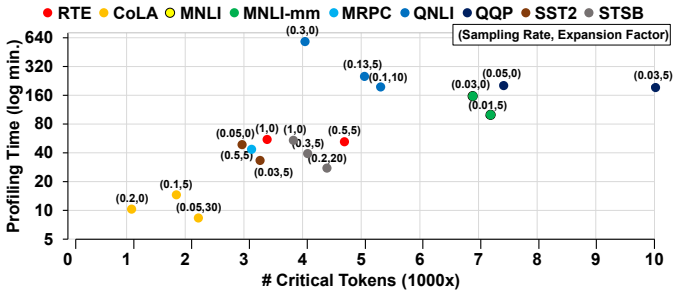

Fig. 3. Profiling time and the number of PCTs for the GLUE datasets. MNLI and MNLI-mm datasets show almost the same results.

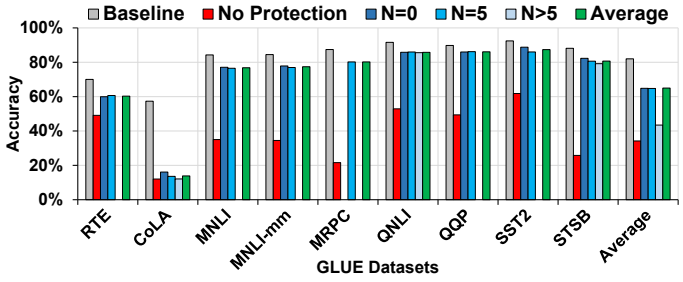

Fig. 4. Accuracy on the GLUE datasets with PCT protection.

The highest number of PCTs is 9983 for the QQP dataset. This implies that C2C can help the system reduce the protection overhead compared to protecting all embedding vectors.

To validate the effect of PCT protection, we perform the same attack simulation in Fig. 1 but with PCT protection. We simulate all configuration combinations in Fig. 3. Fig. 4 compares the accuracy drops after the attack and protection. The accuracy decreases to 34.2% without PCT protection but to 65.0% with PCT protection on average. Except for the CoLA dataset, the average accuracy with all the configuration combinations is 78.9%. It means that C2C can be practical in supporting protection schemes.

V. Conclusion

For the first time, we observe the existence of critical tokens that can affect accuracy more than other tokens and propose an effective critical token classification framework, C2C. According to our simulation, classified critical tokens represent only 13.8% of all tokens but their corruption reduces accuracy by 44.8%.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (2022R1A2C200632113).

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.