Improving the Robustness of Transformer-based Large Language Models with Dynamic Attention

Lujia Shen Yuwen Pu Shouling Ji Changjiang Li Xuhong Zhang
Chunpeng Ge Ting Wang

NDSS 2024

Background

Problem: Large Language Model (LLM) suffers from adversarial attack

Existing Defenses:

Input: Detection, Restoration, …

Model: Adversarial Training, Certified Robustness Approach.

Adversarial Training: computationally expensive, difficult to apply on pre-trained model;
Certified Robustness Training: degrades model’s performance, hard to generalized to different types of attacks, long running time and trivial certified bound.

Solution: Dynamic attention which rectifies the attention mechanism and incorporates dynamic modeling to mitigate adversarial attacks’ influence.

Intuition

1. **Tokens with high attention in adversarial texts are different from those in their original texts.**
   Whether the adversarial examples mislead the attention mechanism and cause the model to misclassify them.

TABLE I: The prediction confidence difference between the attentive tokens of adversarial texts and their original texts.

Dataset	Original	TextBugger	TextFooler	Average
Amazon	0.1899	0.3618	0.3807	0.3713
Twitter	0.0059	0.5458	0.5152	0.5305

2. **Replacing the attention of the adversarial text with the attention of its original text helps the model correctly classify the text.**
   Adversarial example misleads the attention mechanism and leads to the model's misbehavior.

TABLE II: The prediction accuracy of adversarial texts with attention replaced by their benign version.

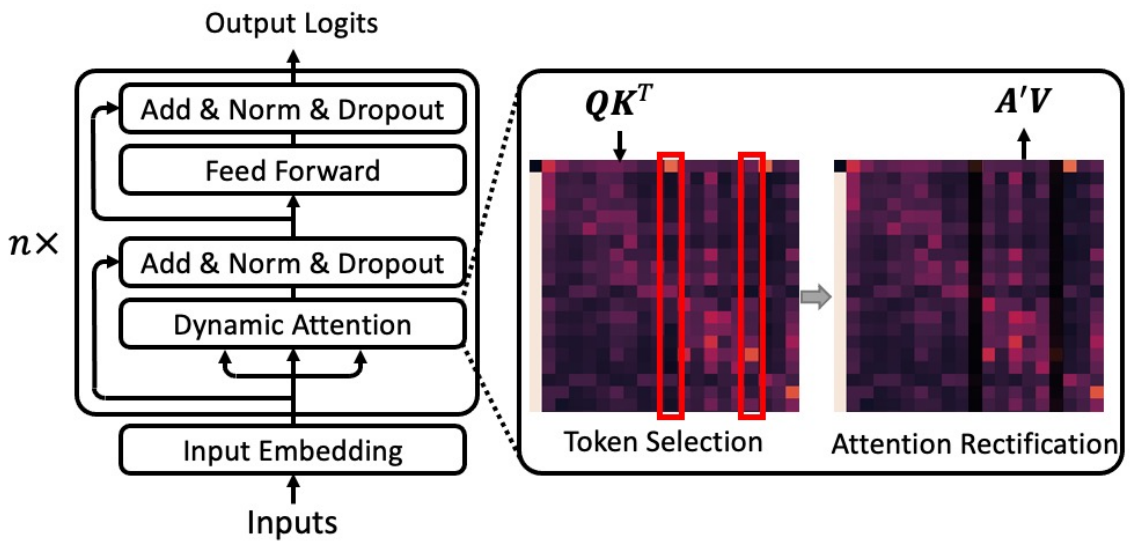
Tuning Method	TextBugger	TextFooler	PWWS
Fine-tuning	86.96%	90.62%	87.27%
Prefix-tuning	82.61%	80.65%	75.81%
Prompt-tuning	94.11%	95.65%	100.0%

3. **Most adversarial examples are inherently unstable.**
   Incorporating dynamic modeling to mitigate adversarial effects.

TABLE III: The transferability rate of adversarial texts under models trained from the same data.

Dataset	TextBugger	TextFooler	PWWS
Amazon	47.16%	41.30%	57.74%
Enron	39.62%	29.49%	26.04%

Dynamic Attention



Attention Rectification

$$A = \sum_t \text{softmax}\left(\frac{Q_t K_t^T}{\sqrt{d}}\right)$$ *Obtain the global attention*

$$A_s = \sum_i A[i,j]$$ *Calculate the attention for each token*

$$\mathcal{T} = \arg\max_m (A_s)$$ *Collect top m token indices by attention value*

$$A'_t[i,j] = \begin{cases} A_t[i,j] & j \notin \mathcal{T} \\ \beta \cdot A_t[i,j] & j \in \mathcal{T} \end{cases}$$ *Rectify the attention with a factor β*

$$H = \text{Concat}_t(A'_t \cdot V_t) \cdot W$$ *Multiply the rectified attention with value*

Dynamic Modeling

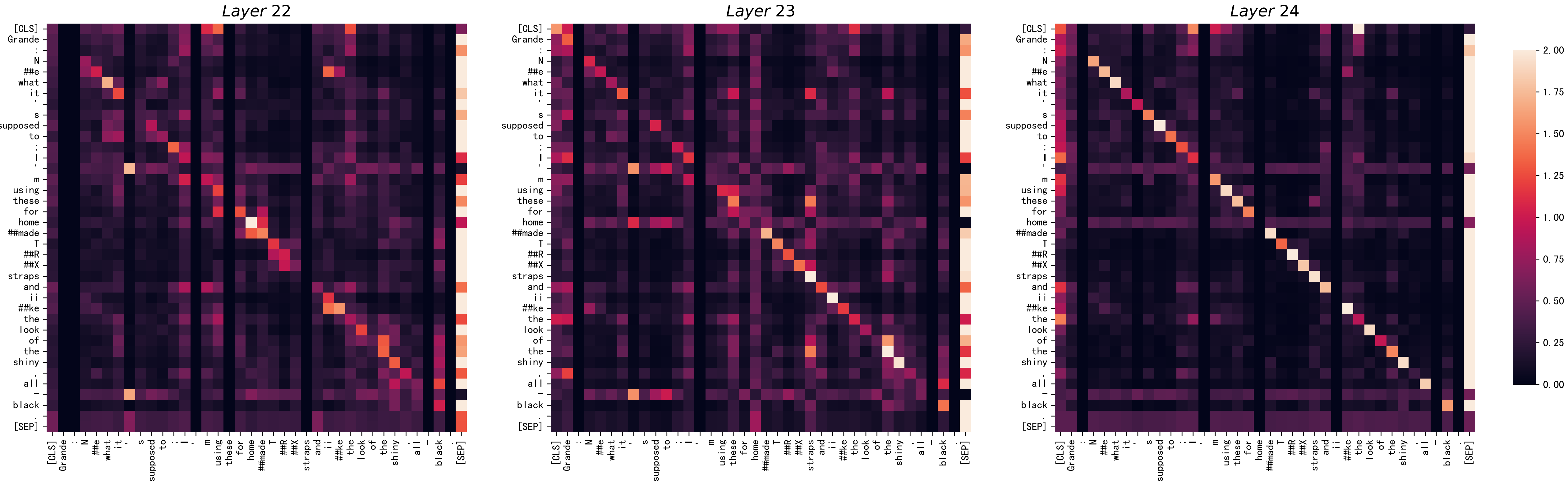
Change the token indices in $\mathcal{T}$ in each layer and change each time they run to achieve dynamization.

Toy Example

Great: Does what it's supposed to; I'm using these for homemade TRX straps and love the look of the shiny, all-black.

Grande: Ne what it's supposed to; I'm using these for homemade TRX straps and iike the look of the shiny, all-black.

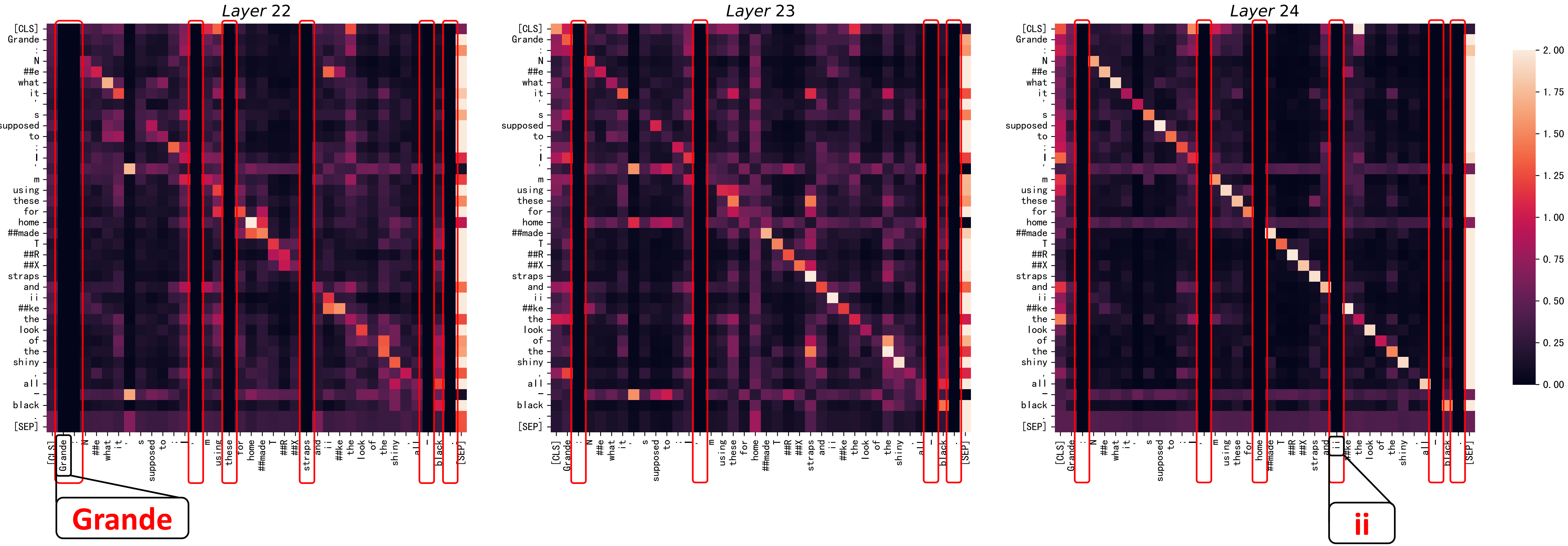
36 tokens, $m_i \sim \mathrm{discrete\_uniform}(\lfloor 0.1\times 36\rfloor, \lfloor 0.2\times 36\rfloor)$, that is $m_i \in \{3,4,5,6,7\}$



Toy Example

Great: Does what it's supposed to; I'm using these for homemade TRX straps and love the look of the shiny, all-black.
Grande: Ne what it's supposed to; I'm using these for homemade TRX straps and iike the look of the shiny, all-black.

36 tokens, $m_i \sim \text{discrete\_uniform}(\lfloor 0.1 \times 36 \rfloor, \lfloor 0.2 \times 36 \rfloor)$, that is $m_i \in \{3,4,5,6,7\}$



Experimental Settings

Datasets

Classification:

Amazon (sentiment analysis),
Twitter (toxic comment detection),
Enron (spam detection)

Generation:

TED Talk (translation)
Gigaword (summarization)

Baselines

No defense (Original)
Defensive Dropout (dropout)
Empirical Adversarial Training (AT)
Information-Bottleneck (IB)

Threat Models

Query Attack (Q)

Direct target model access
Goal: lower ASR, increase queries

Dynamic Transfer Attack (D)

Local dynamic model access or API
Goal: lower transfer ASR

Static Transfer Attack (S)

Local static model access
Goal: lower transfer ASR

Experiment

Model type	ACC	TextFooler			
		ASR_Q	$Query$	ASR_D	ASR_S
original model	93.00%	47.53%	379.42	100.00%	100.00%
dynamic attention	93.07%	52.90%	650.65	24.80%	**30.77%**
dropout	93.20%	45.18%	**744.54**	26.30%	46.56%
fusion	92.27%	50.87%	656.44	**12.88%**	31.67%
IB	95.07%	49.68%	693.89	68.82%	33.48%
dynamic attention + IB	94.07%	48.31%	708.99	27.19%	29.41%
fusion +IB	94.00%	52.48%	639.44	19.75%	28.96%
AT	94.60%	53.70%	333.12	100.00%	100.00%
dynamic attention + AT	94.53%	55.06%	670.92	37.55%	45.93%

Takeaway

1. Dynamic attention is effective in increasing query numbers in query attack;
2. Dynamic attention is effective in decreasing ASR in transfer attack;
3. Dynamic attention can be incorporated with other robustness enhancement module like dropout, information bottleneck and adversarial training to improve robustness.

Experiment

Dataset	Model type	ACC	TextFooler			
			ASR_Q	$Query$	ASR_D	ASR_S
Twitter	original	93.60%	41.67%	115.53	100.00%	100.00%
	dynamic attention	92.13%	**45.32%**	142.14	61.38%	62.74%
	dropout	93.67%	49.15%	**156.67**	48.92%	69.57%
	fusion	91.73%	46.61%	152.16	**42.88%**	**62.22%**
Enron	original	98.27%	44.02%	1706.55	100.00%	100.00%
	dynamic attention	96.73%	15.98%	2670.41	23.93%	37.79%
	dropout	98.33%	**14.23%**	**2746.04**	23.89%	39.18%
	fusion	96.20%	15.38%	2653.1	**11.26%**	**28.88%**

Takeaway

1. Dynamic attention is effective in protecting security-related models against attacks;
2. Fusion model demonstrates superior performance in defending against adversarial attacks.

Stableness Evaluation

Dataset	Model	σ_{adv}	σ_{clean}	ASR_M
Amazon (Fine-Tuning)	dynamic attention	**0.1040**	**0.0273**	**47.51%**
	dropout	0.3742	0.0292	93.21%
	fusion	0.1708	0.0604	55.66%

Takeaway

1. The dynamic attention model offers more consistent predictions than the other two dynamic models;
2. Dropout introduces excessive randomness and results in high variance;
3. Fusion model exhibits improved stability compared to the dropout model.

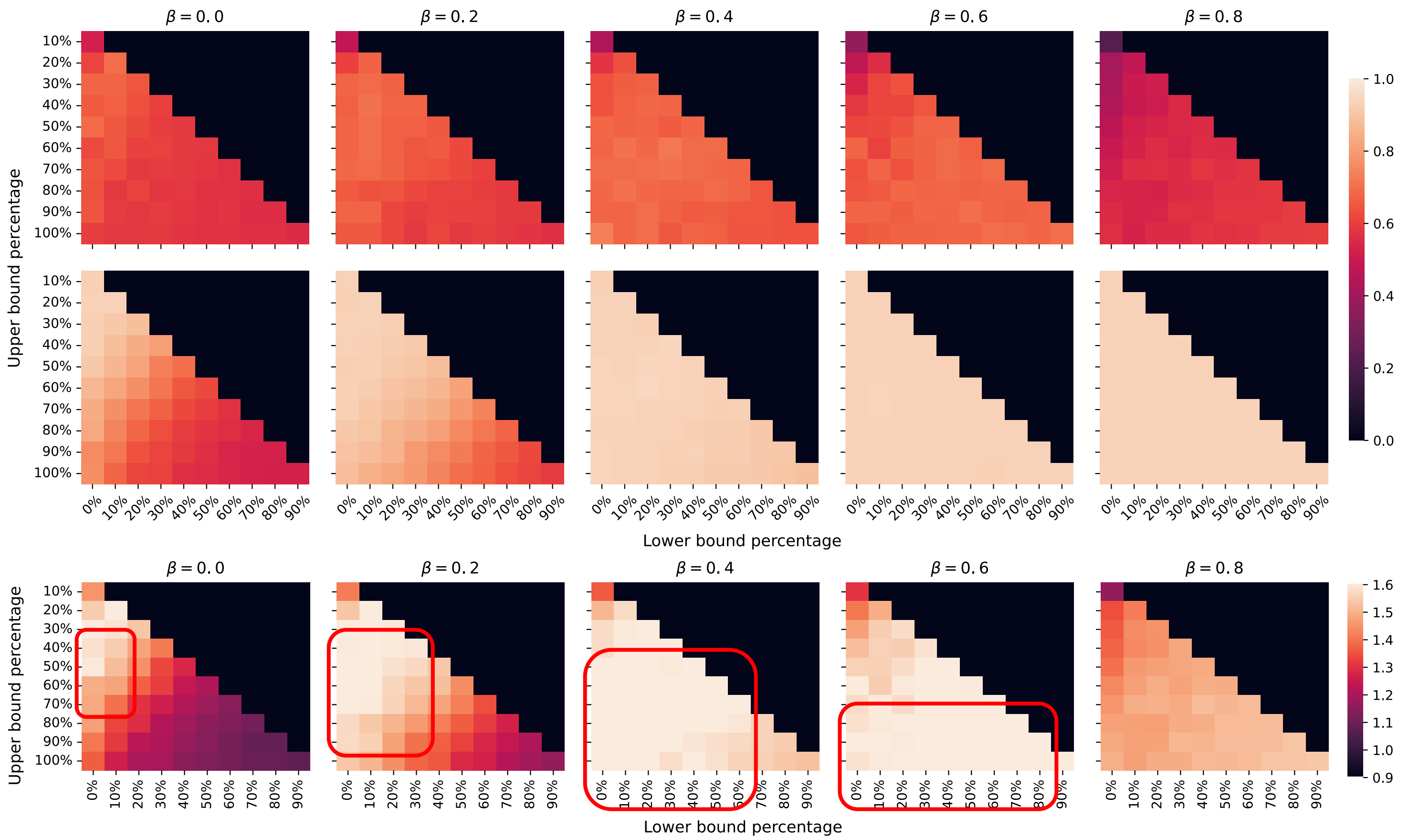
Neural Machine Translation and Summarization

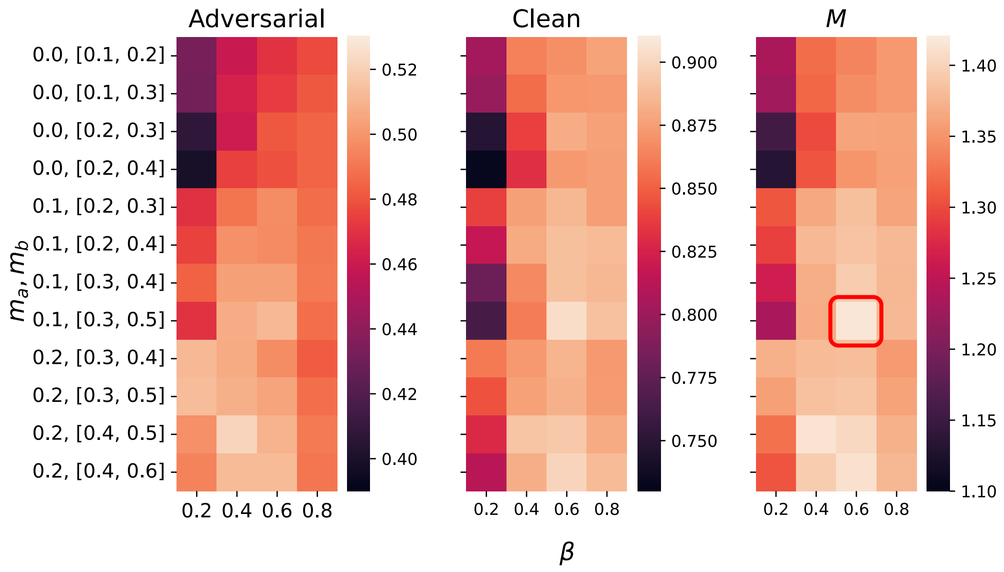
Task	Model	Clean	TextBugger	TextFooler
English to French	original model	1.0000	0.4698	0.4807
	dynamic attention	0.8228	**0.4905**	**0.5194**
	dropout	0.6186	0.3977	0.3949
	fusion model	0.6022	0.3601	0.3983
Summarization	original model	1.0000	0.6159	0.5344
	dynamic attention	0.8120	**0.6276**	**0.5765**
	dropout	0.6149	0.5008	0.4838
	fusion model	0.5960	0.4687	0.3861

Takeaway

1. Dynamic attention models have improved the translation quality of adversarial texts;
2. The performance of the dropout model has deteriorated, which contrary to the results from text classification tasks

Sensitivity Analysis

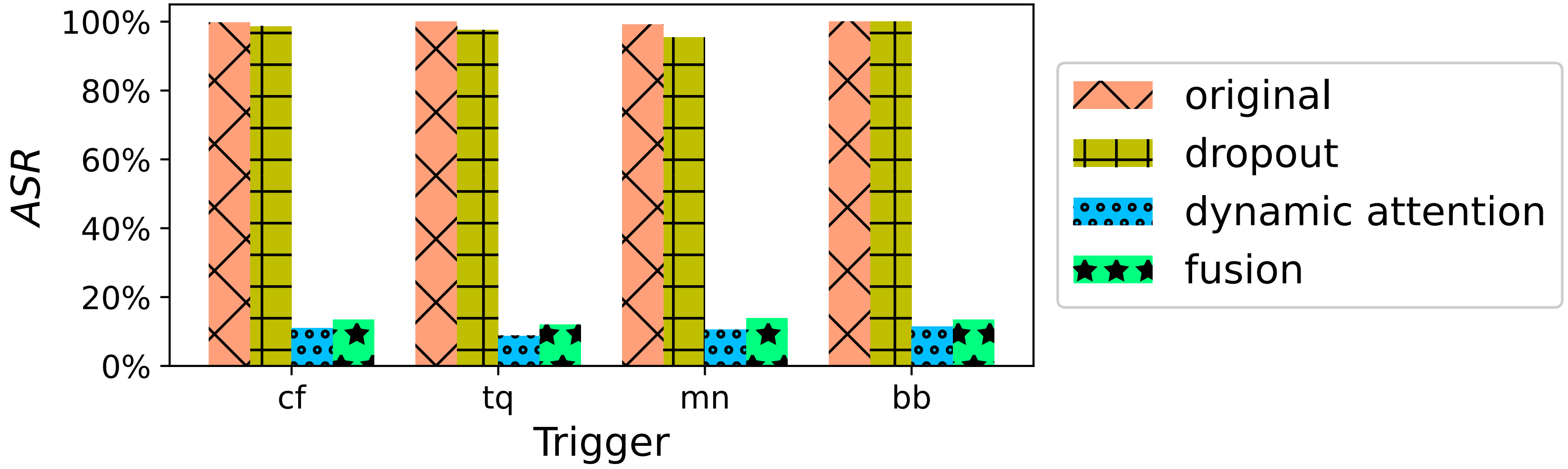


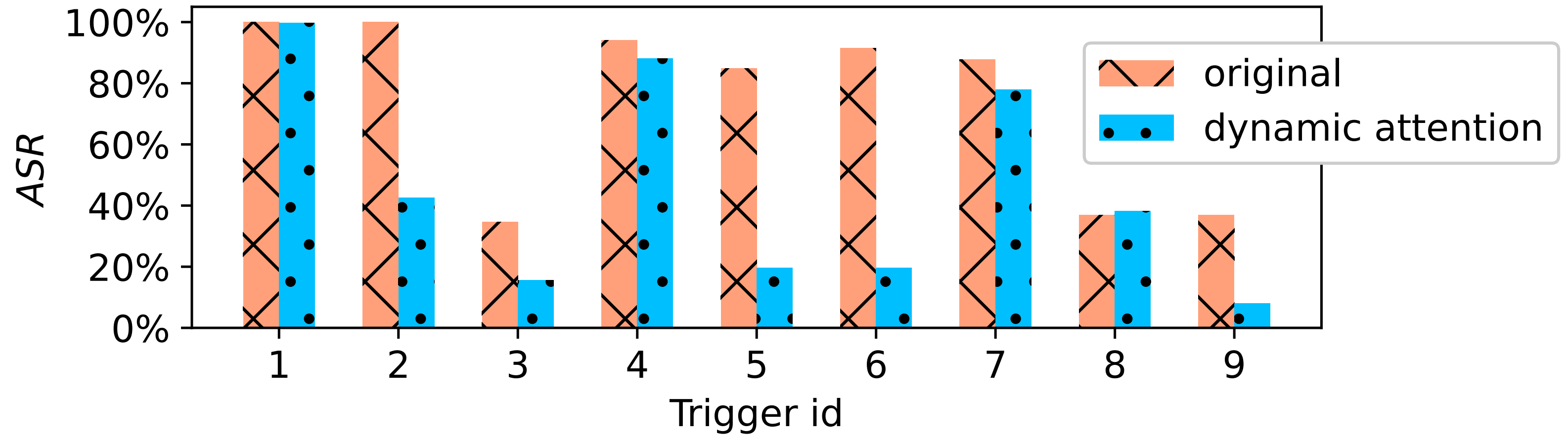


Takeaway

1. A suitable range of m can be determined without setting a smaller upper bound or a larger lower bound.
2. This sensitivity analysis result of text generation task is consistent with previous choice of keeping the top few tokens unchanged and weakening later tokens.

Backdoor Attacks





Takeaway

1. Dynamic attention can effectively find these attentive triggers injected by traditional backdoor attacks like BadNets and eliminate their backdoor influence;
2. Backdoor attacks which associate triggered texts with target hidden representations like POR, are more elusive and harder to defend.

Adaptive Attacks

1 $$\frac{|\mathcal{T}_g \cap \mathcal{T}_o|}{|\mathcal{T}_g \cup \mathcal{T}_o|} > 0.8$$

2 $$\sigma(A_s) < 1.5$$

		TextFooler			
		ASR_{SL}	ASR_{ST}	ASR_{DL}	ASR_{DT}
	dynamic attention	47.53%	34.24%	52.90%	22.22%
Fine-tuning	adaptive 1	29.46%	37.47%	30.11%	23.33%
	adaptive 2	6.88%	55.21%	9.72%	44.44%

Takeaway

1. The two adaptive attacks yield slightly higher transfer ASR on the fine-tuned model;
2. To achieve higher transfer ASR, they drastically decrease the local ASR, which lead to less successfully attacked texts without adaptive attack.

Adaptive Attacks

1 $$\frac{|\mathcal{T}_g \cap \mathcal{T}_o|}{|\mathcal{T}_g \cup \mathcal{T}_o|} > 0.8$$

2 $$\sigma(A_s) < 1.5$$

		TextFooler			
		ASR_{SL}	ASR_{ST}	ASR_{DL}	ASR_{DT}
Fine-tuning	dynamic attention	47.53%	34.24%	52.90%	22.22%
	adaptive 1	29.46%	$\boxed{37.47\%}$	30.11%	$\boxed{23.33\%}$
	adaptive 2	6.88%	$\boxed{55.21\%}$	9.72%	$\boxed{44.44\%}$

Takeaway

1. The two adaptive attacks yield slightly higher transfer ASR on the fine-tuned model;
2. To achieve higher transfer ASR, they drastically decrease the local ASR, which lead to less successfully attacked texts without adaptive attack.

Adaptive Attacks

1 $\frac{|\mathcal{T}_g \cap \mathcal{T}_o|}{|\mathcal{T}_g \cup \mathcal{T}_o|} > 0.8$

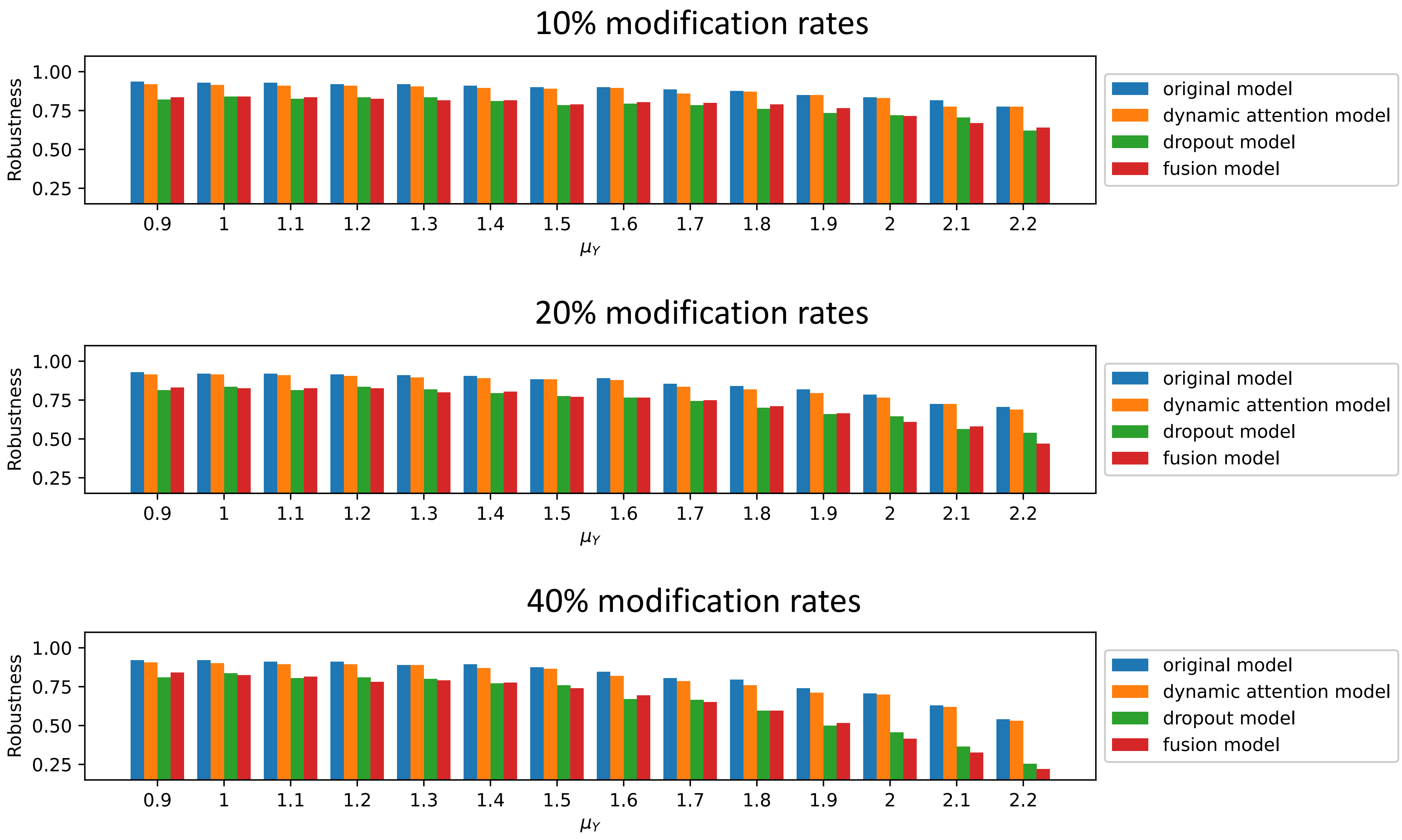
2 $\sigma(A_s) < 1.5$

		TextFooler			
		ASR_{SL}	ASR_{ST}	ASR_{DL}	ASR_{DT}
Fine-tuning	dynamic attention	47.53%	34.24%	52.90%	22.22%
	adaptive 1	29.46%	37.47%	30.11%	23.33%
	adaptive 2	6.88%	55.21%	9.72%	44.44%

Takeaway

1. The two adaptive attacks yield slightly higher transfer ASR on the fine-tuned model;
2. To achieve higher transfer ASR, they drastically decrease the local ASR, which lead to less successfully attacked texts without adaptive attack.

Robustness Analysis



Takeaway

1. Dynamic attention model can preserve 98% of the original model's robustness space;
2. Dropout and fusion models can only preserve 83% of the original robustness.

Conclusions

Dynamic Attention: the first dynamic modeling tailored for transformer-based models that can improve model's robustness;

1. **Dynamic attention serves as a supplementary to existing robustness-enhancement methods instead of an alternative;**
2. **Dynamic attention is effective in mitigating adversarial evasion attacks in classification and generation tasks and can attenuate the effects of backdoor trigger in backdoor model;**
3. **Dynamic attention preserves the robustness space of the original model and maintains more stability in repeated predictions.**