Certified Robustness Under Bounded Levenshtein Distance

Elias Abad Rocamora, Grigorios G. Chrysos and Volkan Cevher

ICLR, Singapore 2025

Adversarial Attacks in Sentence Classification

Elections in Spain: Spanish citizens choose their next president this sunday.

Jordan scored a triple-double in today's match against Celtics.

Tesla stock went up 20% after Elon Musk's last tweet.

PyTorch 3.0 just released, is there any future for TensorFlow?

BERT

textattack/bert-base-uncased-ag-news

World	Sports	Business	Sci/Tech
1	0	0	0
0	**1**	0	0
0.008	0	**0.966**	0.026
0	0	0.001	**0.999**

EPFL

Adversarial Attacks in Sentence Classification

E2lections in Spain: Spanish citizens choose their next president this sunday.

Jordan scored a triple-double in todaK's match against Celtics.

Tesla stock went up 20% after Elon Musk's last twee\t.

PyTorch 3.0 just released, is there any future for Te sorFlow?

BERT

textattack/bert-base-uncased-ag-news

World	Sports	Business	Sci/Tech
0.014	0	0	**0.986**
0.538	0.462	0	0
0	0	0.023	**0.977**
0.003	**0.997**	0	0

Adversarial Attacks in Sentence Classification

E2lections in Spain: Spanish citizens choose their next president this sunday.

Jordan scored a triple-double in todaK's match against Celtics.

Tesla stock went up 20% after Elon Musk's last twee\t.

PyTorch 3.0 just released, is there any future for Te sorFlow?

BERT

textattack/bert-base-uncased-ag-news

World	Sports	Business	Sci/Tech
0.014	0	0	**0.986**
0.538	0.462	0	0
0	0	0.023	**0.977**
0.003	**0.997**	0	0

How can we obtain certifiably robust models? 🤔

EPFL

Robustness Verification

- The worst case margin needs to be positive:

$$g_{y,\hat{y}}(\boldsymbol{S}') = f_{\boldsymbol{\theta}}(\boldsymbol{S}')_y - \max_{\hat{y}\neq y} f_{\boldsymbol{\theta}}(\boldsymbol{S}')_{\hat{y}} > 0, \quad \forall \boldsymbol{S}' : d_{\text{Lev}}(\boldsymbol{S}, \boldsymbol{S}') \leq k\,,$$

where $\boldsymbol{f_\theta}$ is our classifier, $(\boldsymbol{S}, y)$ the original sentence and its label, d_{Lev} the Levenshtein distance and k our perturbation budget.

Robustness Verification

- The worst case margin needs to be positive:

$$g_{y,\hat{y}}(\boldsymbol{S}') = f_{\boldsymbol{\theta}}(\boldsymbol{S}')_y - \max_{\hat{y}\neq y} f_{\boldsymbol{\theta}}(\boldsymbol{S}')_{\hat{y}} > 0, \quad \forall \boldsymbol{S}' : d_{\mathrm{Lev}}(\boldsymbol{S}, \boldsymbol{S}') \leq k\,,$$

where $\boldsymbol{f_\theta}$ is our classifier, $(\boldsymbol{S}, y)$ the original sentence and its label, d_{Lev} the Levenshtein distance and k our perturbation budget.

- **How can we check this condition?**

Brute Force: An illustrative example

$\boldsymbol{S} =$ "Elections in Spain: Spanish citizens choose their next president this Sunday."

$y = 1$ ("World")

- **Can we just try all of the sentences?**

$$\arg\max_j f(\boldsymbol{S}')_j = 1 \qquad \forall \boldsymbol{S}' \in \left\{ \begin{array}{c} \text{"Elec\textbf{!}tions in Spai\textbf{T}: Spanish} \cdots \text{"} \\ \text{"\textbf{3}lections in Spain:\textbf{4}Spanish} \cdots \text{"} \\ \vdots \\ \text{"Elections\textbf{.}in Sain: Spanish} \cdots \text{"} \end{array} \right\}$$

Existing approaches

- **The number of sentences to test grows exponentially with k** 😵‍💫

$$\left|\left\{\begin{array}{c}\text{"Elec}\textbf{!}\text{tions in Spai}\textbf{T}\text{: Spanish}\cdots\text{"}\\ \text{"}\textbf{3}\text{lections in Spain:}\textbf{4}\text{Spanish}\cdots\text{"}\\ \vdots\\ \text{"Elections.in Sain: Spanish}\cdots\text{"}\end{array}\right\}\right| = \left|\{\boldsymbol{S}' : d_{\mathrm{Lev}}(\boldsymbol{S}, \boldsymbol{S}') \leq k\}\right| = 💥$$

- Existing approaches also require a large number of forward passes:

Huang et al., Achieving verified robustness to symbol substitutions via interval bound propagation. *EMNLP 2019*.

Huang et al., Edit distance robustness certificates for sequence classifiers via randomized deletion. *NeurIPS 2023*.

Our idea

- **Lipschitzness + large enough margin ⇒ Robustness** 🤩

$$\left.\begin{array}{c} |g_{y,\hat{y}}(\boldsymbol{S}) - g_{y,\hat{y}}(\boldsymbol{S}')| \leq G \cdot d_{\text{Lev}}(\boldsymbol{S}, \boldsymbol{S}') \\ + \\ g_{y,\hat{y}}(\boldsymbol{S}) > k \cdot G \end{array}\right\} \Rightarrow g_{y,\hat{y}}(\boldsymbol{S}') > 0 \;\; \forall \boldsymbol{S}' : d_{\text{Lev}}(\boldsymbol{S}, \boldsymbol{S}') \leq k$$

where $g_{y,\hat{y}}(\boldsymbol{S}) = f(\boldsymbol{S})_y - f(\boldsymbol{S})_{\hat{y}}$.

- **This paper:** We are able to compute G for convolutional classifiers.

Some results

Table: **Verified accuracy in AG-News:** IBP, proposed by (Huang et al., 2019)

p	k	Acc.(%)	Charmer Adv. Acc.(%)	Charmer Time(s)	BruteF Ver.(%)	BruteF Time(s)	IBP Ver.(%)	IBP Time(s)	LipsLev Ver.(%)	LipsLev Time(s)
∞	1	65.23	47.90	5.70	47.87	16.15	27.77	16.76	32.33	0.0015
	2		32.97	5.70	OOT		✗		11.60	0.0015
1	1	69.63	54.47	5.43	54.43	15.33	18.93	17.56	34.50	0.00140
	2		37.77	5.43	OOT		✗		12.53	0.00140
2	1	74.80	62.20	7.32	62.07	29.12	29.10	31.54	38.80	0.00970
	2		46.47	7.32	OOT		✗		13.93	0.00970

- Lipslev is the only method able to verify for $k > 1$.
- Lipslev is 4 orders of magnitude faster than other methods.

Final remarks

- **Conclusion** 🤩
  - ▷ **LipsLev:** We introduce Lipschitz certification in NLP.
  - ▷ First to verify Levenshtein constraints in a **single forward pass!**
- **Some interesting challenges** 🤔
  - ▷ **Tokenizers:** Can we compute the Lips. constant of tokenizers?
  - ▷ **Transformers:** Can we compute the Lips. constant of Self-Attention?
  - ▷ **Scalability:** Can we scale the approach to larger models?

Thank you!

- **Check our paper:** More details, datasets and ablations!

Thanks for your attention!

contact: `elias.abadrocamora@epfl.ch`

EPFL