Formalizing Robustness against Character-level Perturbations for Neural Network Language Models

Authors: **Zhongkui Ma**[1], Xinguo Feng[1], Zihan Wang[1], Shuofeng Liu[1], Mengyao Ma[1], Hao Guan[1], Mark Huasong Meng[2]

[1]The University of Queensland

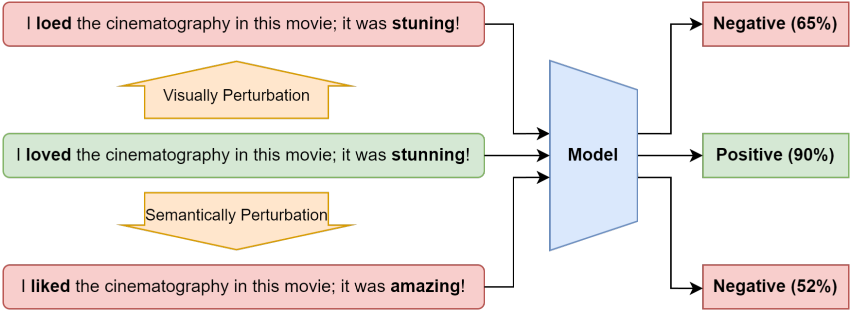
[2]National University of Singapore

November, 2023

Natural Language Processing with Neural Networks

Neural Networks revolutionize the performance of natural language processing (NLP).

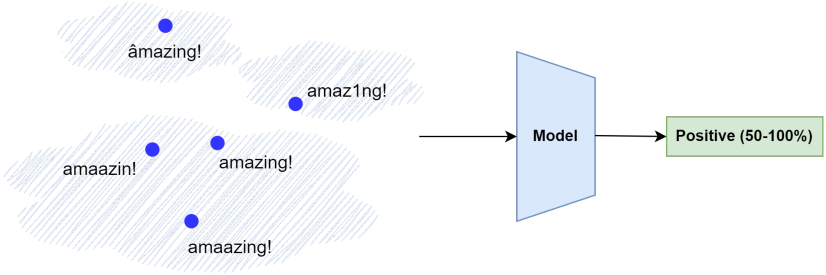
However, adversarial samples pose a considerable risk to their reliability and accuracy.



THE UNIVERSITY OF QUEENSLAND AUSTRALIA

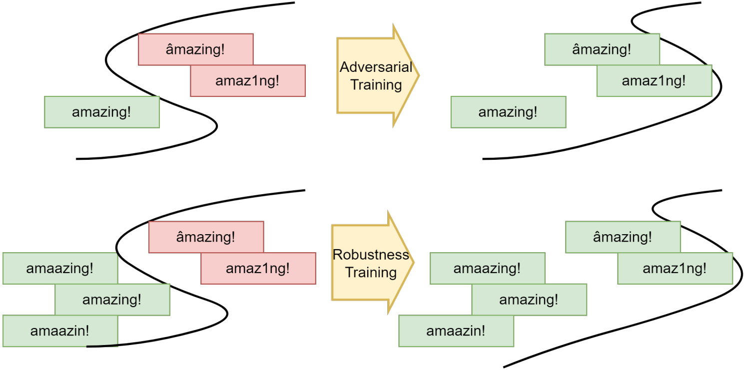
What is robustness?

Robustness is a property that ensures the model produces the same result no matter whether the sample is with or without perturbation.



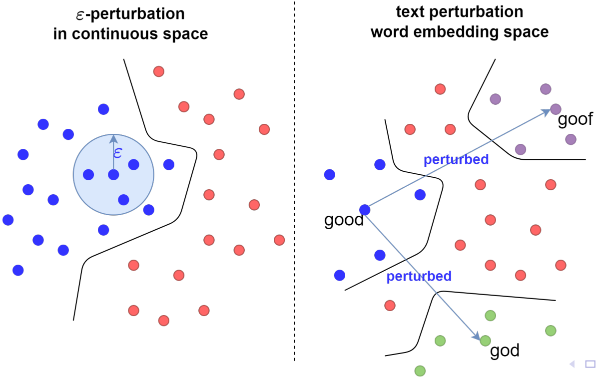
How to enhance robustness?

- Adversarial training: train model with adversarial samples that mislead the model
    - Expensive because of the search for adversarial samples
- Robustness training: train models with perturbed samples
    - Cheap because of the simple generation of perturbed samples



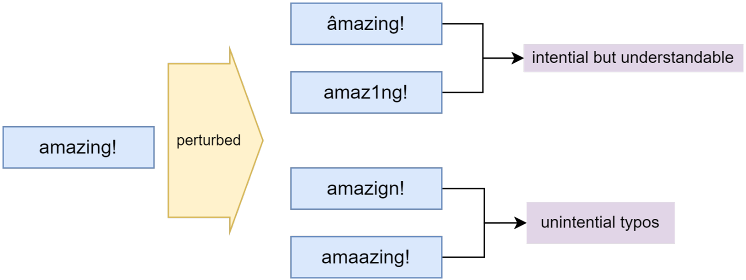
How to generate perturbed samples?

- Continuous data (image, audio signal)
  - Tiny numerical change like $x \pm \epsilon$
- Discrete data (text)
  - character-level, e.g., typos or mistakes, keyboard typos, inserting special characters, ...
  - word-level, e.g., synonyms, POS perturbation, domain-specific perturbation, ...



Difficulties of generating perturbed texts?

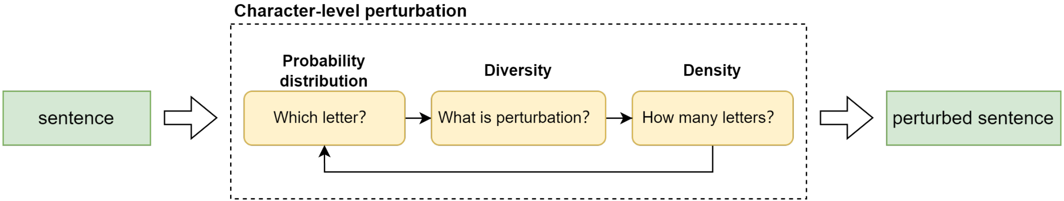
- Practical meaning
  - It should happen in the real world, e.g., typos, letters swap, repeated letters, ...
- Semantic meaning
  - It should be understandable by humans
  - It should maintain certain linguistic properties, e.g., grammar, semantics, style, ...



THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Our work - PdD

In this work, we propose our approach **PdD** to customize character-level perturbation for text data and aim to enhance the robustness of language models by robustness training.



Unified Formalization of Perturbation

Perturbation Metrics

Which item of input vector is perturbed?

Definition (Probability Distribution)

The probability distribution P of a perturbation for a given input vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ refers to the distribution that governs the probability $P(i)$ of i-th element x_i $(1 \leq i \leq n)$ in the input vector being perturbed.

Uniform Distribution

`Hēllo âll, gree\u2029tings to Brisbanē, and ä warm welcomu to Australiâ!`

Normal Distribution

Hello all, greetings tô Bris\u2029bänē, aud ä wârm welco~~m~~e to Australia!

Formalization

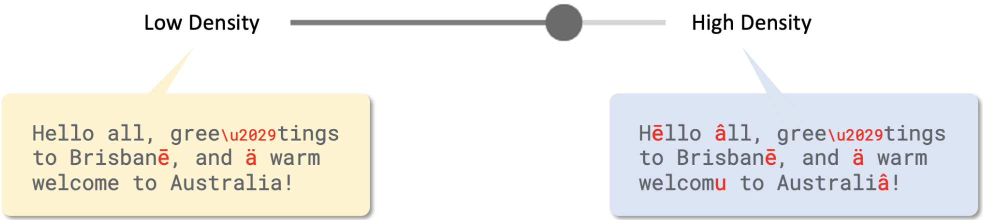
Unified Formalization of Perturbation

Perturbation Metrics

How many items of the input vector are perturbed?

Definition (Density)

The density d $(0 \leq d \leq 1)$ of perturbation refers to the percentage of perturbed elements in the given input vector.



Formalization

Unified Formalization of Perturbation

Perturbation Metrics

How do the items of input vector are perturbed?

Definition (Diversity)

The diversity $D = \{(x_i, D_i) | 1 \leq i \leq n\}$ is a set of sets that contains all pairs (x_i, D_i), where $D_i = \{x_i', x_i'', \cdots\}$ is the set of all possible candidate elements that can be used to perturb x_i.

Low Diversity

Hēllo âll, greetings to Brisbanē, and ä warm welcome to Australiâ!

High Diversity

Hēllo âll, gree\u2029tings to Brisbanē, and ä warm welcomu to Australiâ!

Unified Formalization of Perturbation

Example 1: (ϵ-perturbation)

- Probability Distribution: Uniform distribution
- Density: The percentage of items to perturbed
- Diversity: $[x_i - \epsilon, x_i + \epsilon]$

Example 2: (Character-level perturbation)

- Probability Distribution: The probability of a character being perturbed
- Density: The proportion of characters subject to perturbation.
- Diversity: A specified set used to be the perturbed character

Definition (Language Model)

A language model f is a function that takes a sequence of words $\boldsymbol{x} \in \Sigma^*$ as input and outputs a sequence of words $\boldsymbol{y} \in \Sigma^*$, where Σ^* is its finite word set.

Definition (Robustness of Language Model)

Robustness is the property that, given a language model f and a input $\boldsymbol{x}_0$ and its perturbed values set $U_{\boldsymbol{x}_0}$, the resulting output set $f(U_{\boldsymbol{x}_0})$ satisfies being a subset of the predefined set $U_{\boldsymbol{y}_0} \subseteq \Sigma^*$, i.e.

$$\forall x' \in U_{\boldsymbol{x}_o}, y' = f(x') \implies y' \in U_{\boldsymbol{y}_0}$$

Formalization

Character-level Operation

Replacement For each element x_i in the input vector $\boldsymbol{x}$, we define a finite discrete set of candidates D_i. The set D_i comprises k candidate characters, denoted as c_{ij}, where $1 \leq j \leq k$. Each c_{ij} represents a possible substitution from the candidate set D_i.

This movie was exciting → This movie ŵas exciting

The cast is good → The cäst is good

Highly addictive → Highly addįctive

Formalization

Character-level Operation

Deletion For deletion, we set $(x_i, D_i) = (x_i, \texttt{[EMP]})$, where [EMP] represents an empty character.

This movie ~~w~~as exciting → This movie as exciting

The c~~a~~st is good → The cst is good

Highly add~~i~~ctive → Highly addctive

Formalization

Character-level Operation

Insertion For insertion, we set $(x_i, D_i) = (x_i, x_i c_{ij}, c_{ij} x_i | 1 \leq j \leq k, k \in \mathbb{Z})$, where c_{ij} is defined as in the replacement operation. Here, $x_i c_{ij}$ or $c_{ij} x_i$ represents the concatenation of the original character x_i and the inserted character c_{ij}.

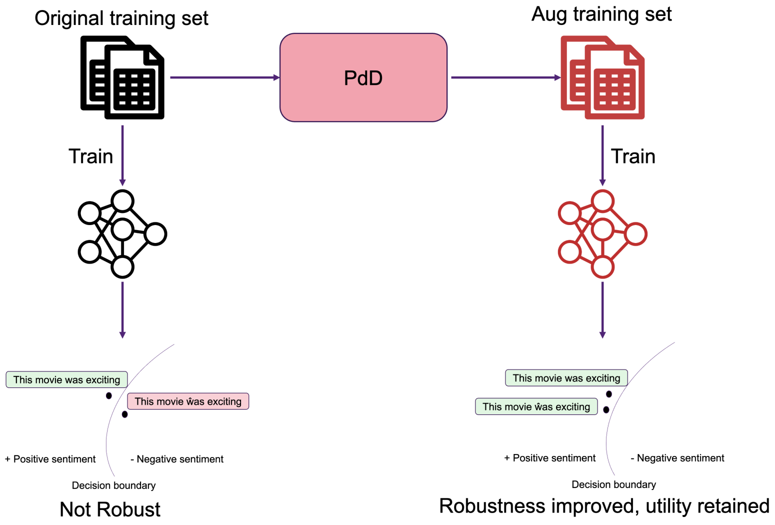
This movie was exciting → This movie wass exciting

The cast is good → The caast is good

Highly addictive → Highly addiictive

Approach

PdD Workflow.



Evaluation

Research Questions

- **RQ1**: Does the character-level perturbation affect the performance of language models? Are the effects of different perturbations different?
- **RQ2**: Does the robustness training enhance the model's performance on perturbed samples?
- **RQ3**: Does the robustness sacrifice the model's performance on original/clean samples?

Experimental Settings

Original Dataset & Augmented Dataset

- Clean Dataset: the original dataset without perturbation
- Augmented Dataset: the original dataset with perturbed samples

Original Model & Augmented Model

- Clean Model: trained with the dataset without perturbed samples
- Augmented Model: trained with the dataset with perturbed samples

	Original Dataset	Augmented Dataset
Original Model		RQ1
Augmented Model	RQ3	RQ2

Evaluation

Experimental Settings

Perturbation Settings

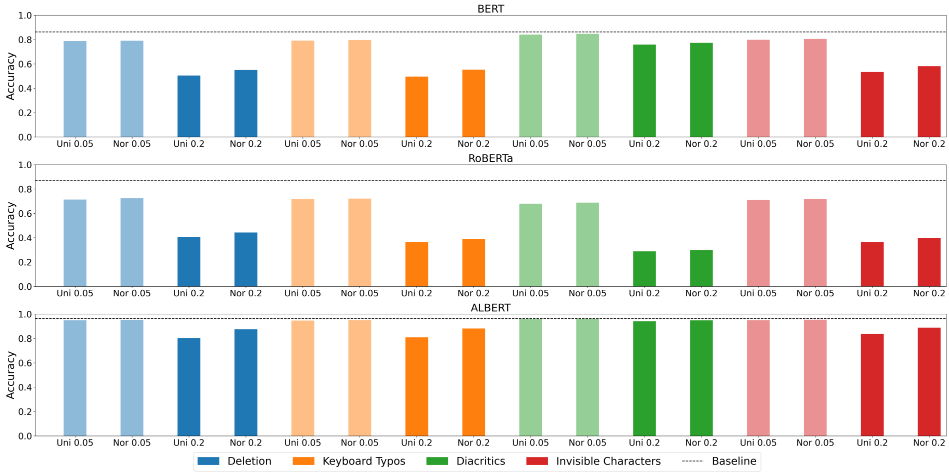
- **Probability distribution**: uniform distribution, normal distribution
- **Density**: low level (5%), high level (20%)
- **Diversity**:
  - **Deletion**: replacing a character with an empty character.
  - **Keyboard typos**: mistakenly pressing adjacent keys (8 neighboring keys).
  - **Diacritics**: replace a character with a set of 5 different diacritics.
  - **Invisible characters** insert one character that is not detectable by human eyes; from a pool of 48 invisible characters.

Models, Datasets, and Tasks

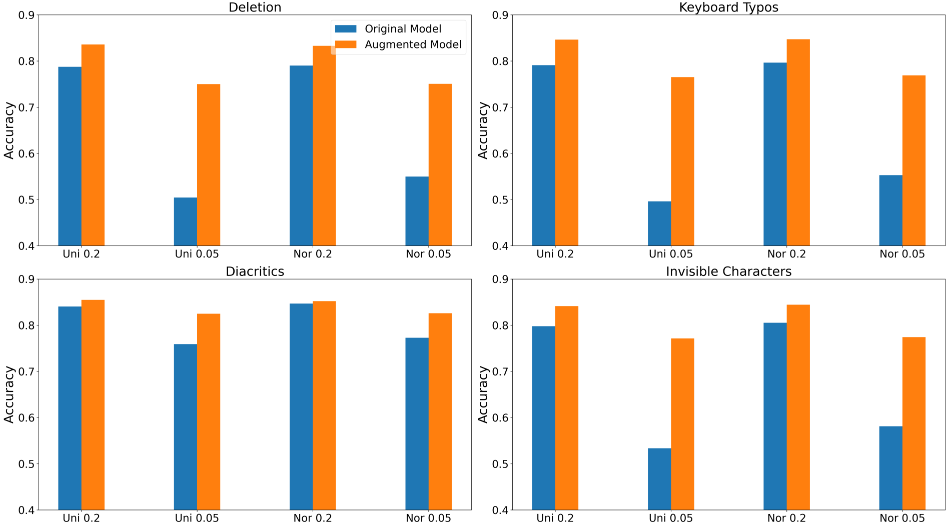
Model	Dataset	Task	#Class
BERT	Rotten Tomatoes	Sentiment Analysis	2
RoBERTa	SNIL	Natural Language Inference	3
ALBERT	E-commerce	Text Classification	4

Evaluation

RQ1: Effects of different perturbations (probability distributions (**Uni**form, **Nor**mal), densities (**0.05**, **0.2**), and four types of diversity) on original models.

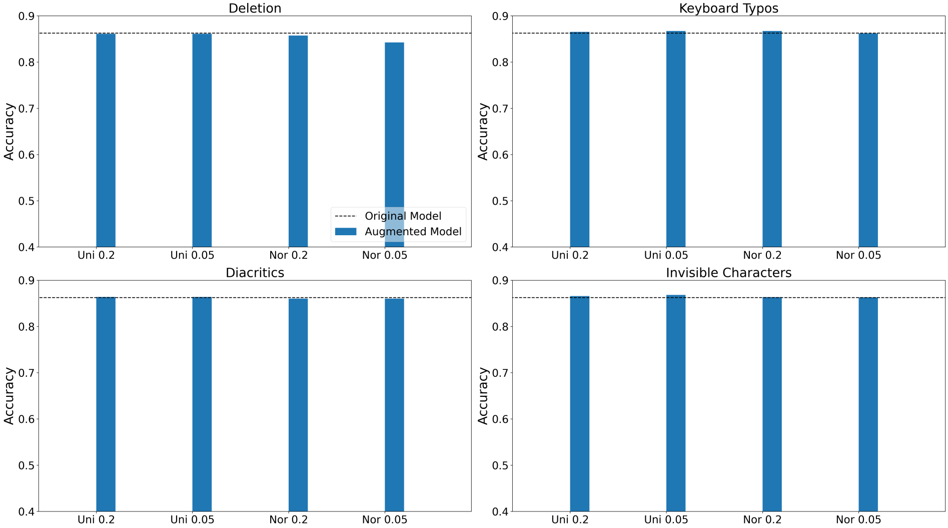


RQ2: Comparison of performance on perturbed datasets between the original model and the augmented model (BERT Model).



Evaluation

RQ3: Performance on the original datasets of the augmented model (BERT Model).



Conclusion

- We formalize the perturbation of text data and propose three metrics: probability distribution, density, and diversity.
- We propose a perturbation generation algorithm, PdD configurable by the three metrics.
- We implement robustness training on various typical language models using PdD. The results demonstrate that the generated perturbed datasets are beneficial for enhancing the robustness against specified character-level perturbations.

Thank you