"That Is a Suspicious Reaction!": Interpreting Logits Variation to Detect NLP Adversarial Attacks

Edoardo Mosca[1]  Shreyash Agarwal[1] 

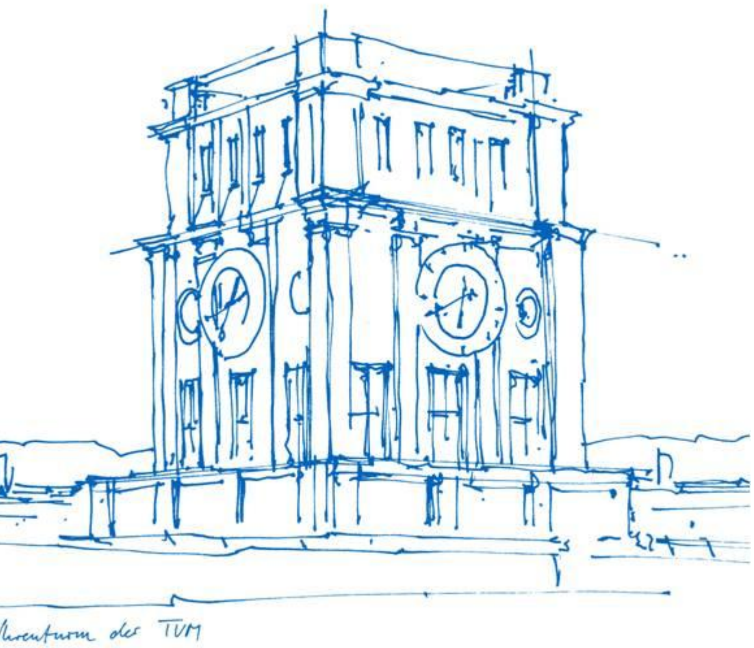
Javier Rando-Ramirez[2]  Georg Groh[1] 

[1] *Technical University of Munich, Germany*

[2] *ETH Zurich, Switzerland*

ACL 2022

22nd-27th May | Dublin



Adversarial Attacks in NLP

Original sentence

This is a **great** movie. Too bad it is not available on home video. → POSITIVE

Word-level adversarial attack

This is a **expectant** movie. Too bad it is not available on home video. → NEGATIVE

Character-level adversarial attack

This is a **greatt** movie. Too bad it is not available on home video. → NEGATIVE

Defense against Adversarial Attacks in NLP

Character-level attacks → **Spell and syntax checkers**

Word-level adversarial attacks

Robustness enhancement

Make the model inherently less likely to be fooled.

- Adversarial training
- Synonym Encoding Method
- …

Adversarial detection

Build a post-hoc system to detect potential attacks and raise alerts.

- Discriminate Perturbation
- Frequency-Guided Word Substitution (FGWS)

Our contribution

1. **Word-level Differential Reaction (WDR)**:
   Logit-based metric to capture words with suspiciously high impact in predictions.

2. WDR scores are suitable to train an **adverarial detector**
   outperforms SOTA techniques.

3. Prove such detector to have **full transferability**
   Across different datasets, attacks and target models (without retraining).

Word-Level Differential Reaction (WDR)

Model prediction: $y^* = \arg\max_y p(y|x)$

Effect of replacing a word x_i in sentence x:

$$WDR(x_i, f) = \boxed{f(x \backslash x_i)_{y^*}} - \boxed{\max_{y \neq y^*} f(x \backslash x_i)_y}$$

Logit for class y^* when removing word x_i

Highest logit for all other classes when removing word x_i

WDR and adversarial attacks

Original sentence: Neg. Review (*Class 0*)

This is absolutely the worst trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a sick joke! [...]

Removed Word x_i	Logit *Class 0*	Logit *Class 1*	WDR $WDR(x_i, f)$
∅	3.44	-3.46	**6.89**
worst	1.68	-1.75	**3.43**
sick	3.34	-3.42	**6.76**
absolutely	3.40	-3.45	**6.86**
realized	3.41	-3.47	**6.89**

Adversarial sentence: Pos. Review (*Class 1*)

This is absolutely the tough trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a silly joke! [...]

Removed Word x_i	Logit *Class 0*	Logit *Class 1*	WDR $WDR(x_i, f)$
∅	-1.85	2.17	**4.02**
tough	2.14	-1.50	**-3.64**
silly	1.38	-1.37	**-2.75**
absolutely	-0.31	0.48	**0.79**
realized	-1.07	1.36	**2.43**

$$WDR(tough, f) = -1.50 - 2.14$$

$$\boxed{WDR(x_i, f) = f(x \backslash x_i)_{y^*} - \max_{y \neq y^*} f(x \backslash x_i)_y}$$

WDR and adversarial attacks

We expect predictions for adversarial samples to **strongly depend on adversarial replacements**. This is captured by WDR!

Original sentence: Neg. Review (*Class 0*)

This is absolutely the worst trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a sick joke! [...]

Removed Word x_i	Logit *Class 0*	Logit *Class 1*	WDR $WDR(x_i, f)$
$\emptyset$	3.44	-3.46	**6.89**
worst	1.68	-1.75	**3.43**
sick	3.34	-3.42	**6.76**
absolutely	3.40	-3.45	**6.86**
realized	3.41	-3.47	**6.89**

Adversarial sentence: Pos. Review (*Class 1*)

This is absolutely the tough trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a silly joke! [...]

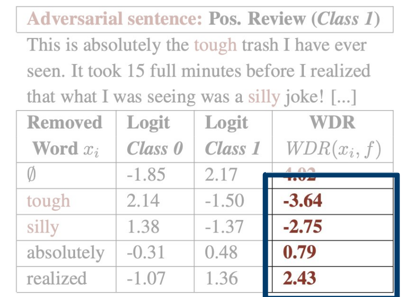
Removed Word x_i	Logit *Class 0*	Logit *Class 1*	WDR $WDR(x_i, f)$
$\emptyset$	-1.85	2.17	**4.02**
tough	2.14	-1.50	**-3.64**
silly	1.38	-1.37	**-2.75**
absolutely	-0.31	0.48	**0.79**
realized	-1.07	1.36	**2.43**

Table 1: $WDR(x_i, f)$ scores computed for an original sentence and its corresponding adversarial perturbation. Results show how when removing adversarial words such as *tough* or *silly*, the original class is recovered and the WDR becomes negative. $\emptyset$ corresponds to the prediction without any replacements

Negative WDR → change in prediction when removing a word

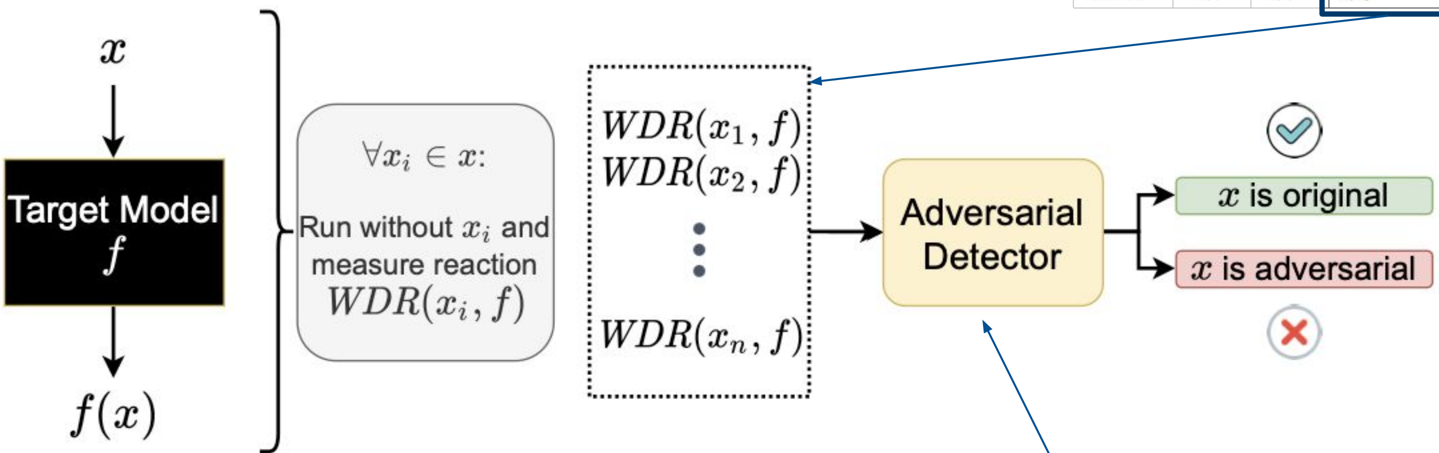
Build an adversarial detector

WDR values are sorted by ascending order to ensure a pattern shows up no matter which words were replaced.



Adversarial sentence: Pos. Review (*Class 1*)
This is absolutely the tough trash I have ever seen. It took 15 full minutes before I realized that what I was seeing was a silly joke! [...]

Removed Word x_i	Logit *Class 0*	Logit *Class 1*	WDR $WDR(x_i, f)$
$\emptyset$	-1.85	2.17	4.02
tough	2.14	-1.50	**-3.64**
silly	1.38	-1.37	**-2.75**
absolutely	-0.31	0.48	**0.79**
realized	-1.07	1.36	**2.43**



Training Procedure

1. Generate Adversarial Samples
2. Computer WDR scores for all samples
3. Train the adversarial detector

Experimental setup

To evaluate our defense, we define a set of different datasets, target models and adversarial attacks.

Datasets

- IMDb
- Rotten tomatoes Movie Reviews
- Yelp Polarity
- AG News

Target models

- DistilBERT
- BERT
- CNN
- LSTM

Adv. attacks

- PWWS
- IGA
- BAE
- TextFooler

Training a detector model

Training setup IMDb / DistilBERT / PWWS

Model	F1-Score	Adv. Recall
XGBoost	**92.4**	95.2
AdaBoost	91.8	**96.0**
LightGBM	92.0	93.7
SVM	92.0	94.8
Random Forest	91.5	93.7
Perceptron NN	90.4	88.1

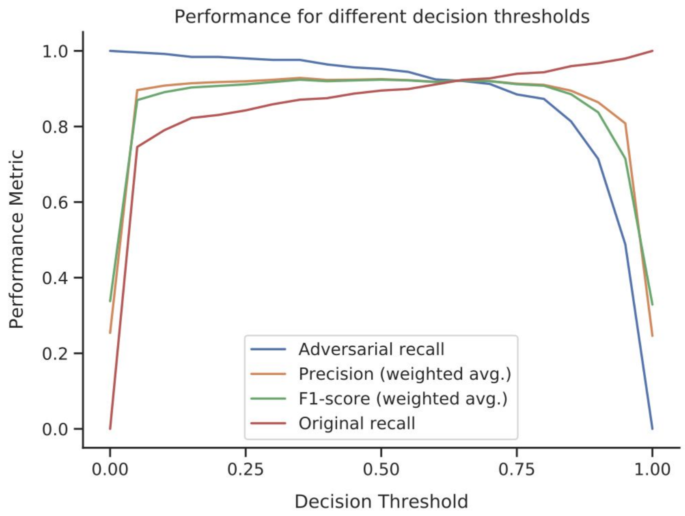
Table 2: Performance comparison of different detector architectures on IMDb adversarial attacks generated with PWWS and targeting a DistilBERT transformer.

Training a detector model

Training setup IMDb / DistilBERT / PWWS

Model	F1-Score	Adv. Recall
XGBoost	**92.4**	95.2
AdaBoost	91.8	**96.0**
LightGBM	92.0	93.7
SVM	92.0	94.8
Random Forest	91.5	93.7
Perceptron NN	90.4	88.1

Table 2: Performance comparison of different detector architectures on IMDb adversarial attacks generated with PWWS and targeting a DistilBERT transformer.



DT	Precision	F1	Adv. Recall	Orig. Recall
0.5	92.5	92.4	95.2	89.5
0.4	92.3	92.0	96.4	87.5
0.3	92.4	91.8	97.6	85.9
0.15	91.5	90.3	**98.4**	82.3

Table 4: Performance comparison using different *Decision Thresholds* (DT) for our XGBoost classifier on the configuration (IMDb, DistilBERT, PWWS). The used default value is 0.5.

Evaluating generalization

Training Config.

Configuration			***WDR*** (Ours)		***FGWS*** (Mozes et al., 2021)	
Model	**Dataset**	**Attack**	**F1-Score**	**Adv. Recall**	**F1-Score**	**Adv. Recall**
DistilBERT	IMDb	PWWS	**92.1 ± 0.5**	94.2 ± 1.1	89.5	82.7
LSTM	IMDb	PWWS	**84.1 ± 3.4**	86.8 ± 8.5	80.0	69.6
CNN	IMDb	PWWS	84.3 ± 3.1	90.0 ± 6.2	**86.3**	79.6
BERT	IMDb	PWWS	**92.4 ± 0.7**	92.5 ± 1.8	89.8	82.7
DistilBERT	AG News	PWWS	**93.1 ± 0.6**	96.1 ± 2.2	89.5	84.6
DistilBERT	RTMR	PWWS	74.1 ± 3.1	85.1 ± 8.6	**78.9**	67.8
DistilBERT	IMDb	TextFooler	**94.2 ± 0.8**	97.3 ± 0.9	86.0	77.6
DistilBERT	IMDb	IGA	**88.5 ± 0.9**	95.5 ± 1.3	83.8	74.8
DistilBERT	IMDb	BAE	**88.0 ± 0.9**	96.3 ± 1.0	65.6	50.2
DistilBERT	RTMR	IGA	**70.4 ± 5.5**	90.2 ± 6.9	68.1	55.2
DistilBERT	RTMR	BAE	**68.5 ± 4.3**	82.2 ± 9.0	29.4	18.5
DistilBERT	AG News	BAE	**81.0 ± 4.3**	95.4 ± 3.8	55.8	44.0
BERT	YELP	PWWS	89.4 ± 0.6	85.3 ± 1.7	**91.2**	85.6
BERT	YELP	TextFooler	**95.9 ± 0.3**	97.5 ± 0.6	90.5	84.2

(a) Performance results for detector trained on (DistilBERT, IMDb, PWWS).

8.89 pp. better on average !!

Understanding the adversarial detector

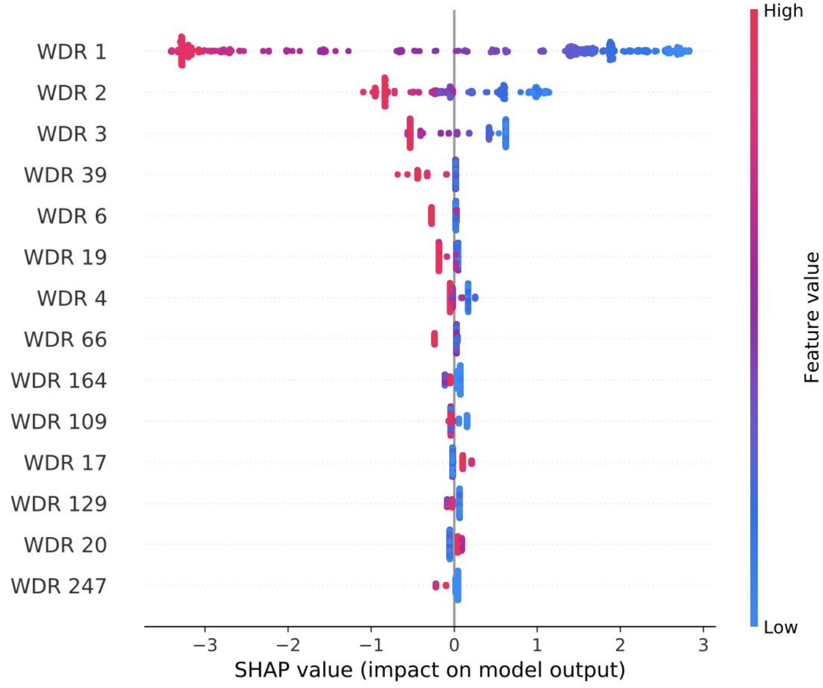


Figure 3: WDR scores with the highest impact (SHAP value) on the detector's prediction. Please recall that the WDR scores are sorted by magnitude. For instance, WDR 1 is the first and largest WDR score.

The first 3 WDR are the most relevant for the detector.

Being negative correlates with being adversarial.

Takeaways and Future Work

- WDR is extremely good for identifying adversarial examples
- Our pipeline is model-, #classes-, and detector-agnostic
- Out-of-the-box transferability works like a charm

- Expensive to compute (many forward passes needed)

- Would it work against character-level attacks?
- Is it resilient to adaptive attacks?

Thank you!!

Edoardo
Mosca

Shreyash
Agarwal

Javier
Rando-Ramirez

Georg
Groh