

Exploiting Natural Language Services: A Polarity Based Black-box Attack

Fatma GUMUS, M. Fatih AMASYALI

Frontiers of Computer Science, DOI: [10.1007/s11704-021-0198-y](https://doi.org/10.1007/s11704-021-0198-y)

Problems & Ideas

- Problem of exploiting NLP services in black-box setting:
 - Querying the victim model to find the important words to manipulate potentially cause a big request-response overhead.
- Ideas: Replace words by their class polarity scores calculated on an external dataset.

0. Set the perturbation parameters.

Source and target class determine the attack direction. Min. target probability controls the minimum required attack confidence, distortion range sets the maximum perturbation per swapped word, and max. num. of swaps regulates the number of swaps allowed.

$$0 < \alpha \leq 5.0, 0 < \theta < 1.0, \delta > 0$$

1. Preprocess the original sample.

-Lower case, fix contractions, and remove punctuations. Filter the words by PoS tag.
 -Random shuffle the filtered words. Run the perturbation scheme for each word in the shuffled order.

2. Start the search for the word that is to be replaced: **best**.

2.1. Search the polarity table.

-Filter the available words for swap by PoS tag.
 -Calculate the word distortion by the difference between the original and swap polarity scores. Order ascending by the distortion.

-Mark the candidates that are within the required distortion range.

2.2. Query the black box.

Select one among the previous sample and the new adversarial samples that returns the most confidence towards the target class.

3. If the max. num. of swaps are made, return the selected adversarial sample. If the minimum target probability is not reached and there are more words marked for swap, go to Step 2, otherwise, return the selected adversarial sample.

Perturbation Settings			
Source Class	(+)	Target Class	(-)
Min. target prob. (θ)	0.51	Distortion range (α)	1.5
Max. num. of swaps (δ)	100	Allow swap for PoS tags	NN,VBZ,JJS,NNS

TEXT	the	album	has	one	of	the	best	lyrics
POS	DT	NN	VBZ	CD	IN	DT	JJS	NNS
Swap Words		album	has				best	lyrics
Shuffled Words		best	album	lyrics	has			

	w_t	POS	Polarity Score $s_{w_t}(+ -)$	Distortion $(s_{best} - s_{w_t})$
Original	best	JJS	0.99	
	finest	JJS	1.49	-0.5
Candidates	greatest	JJS	0.83	0.16
	latest	JJS	0.16	0.83
Out of Range	worst	JJS	-2.72	3.71

Previous	the	album	has	one	of	the	best	lyrics	P(- Previous)=0.25
Adv1	the	album	has	one	of	the	FINEST	lyrics	P(- Adv1)=0.16
Adv2	the	album	has	one	of	the	GREATEST	lyrics	P(- Adv2)=0.24
Adv3	the	album	has	one	of	the	LATEST	lyrics	P(- Adv3)=0.37
Adv4	the	album	has	one	of	the	WORST	lyrics	P(- Adv4)=0.69

Victim Model

Figure 1. Example on one iteration of our proposed attack.

Main Contributions

- We perform black-box word perturbations without an oracle, or a trained generator.
- Classifiers were fooled by a small number of word replacements.
- The generated samples retain semantic similarity with the originals based on embedding space.
- We evaluate a true black-box attack on IBM Watson Natural Language Understanding (NLU) service to further confirm its applicability on real case scenarios.

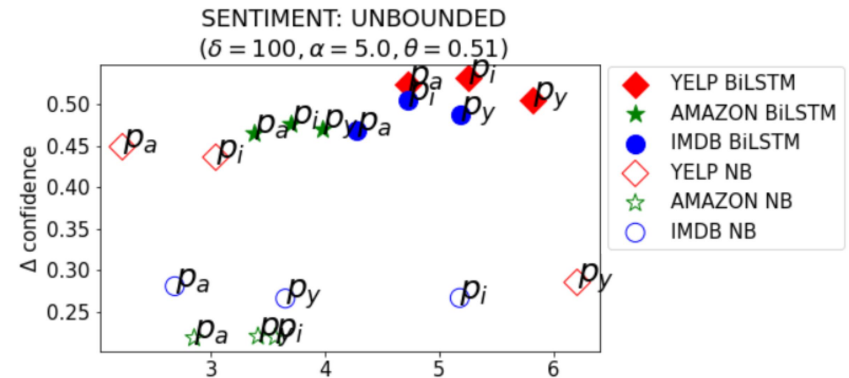


Figure 2. Number of Word replacements (x axis) and the change of probability score on false class (y-axis) on the Sentiment Task.

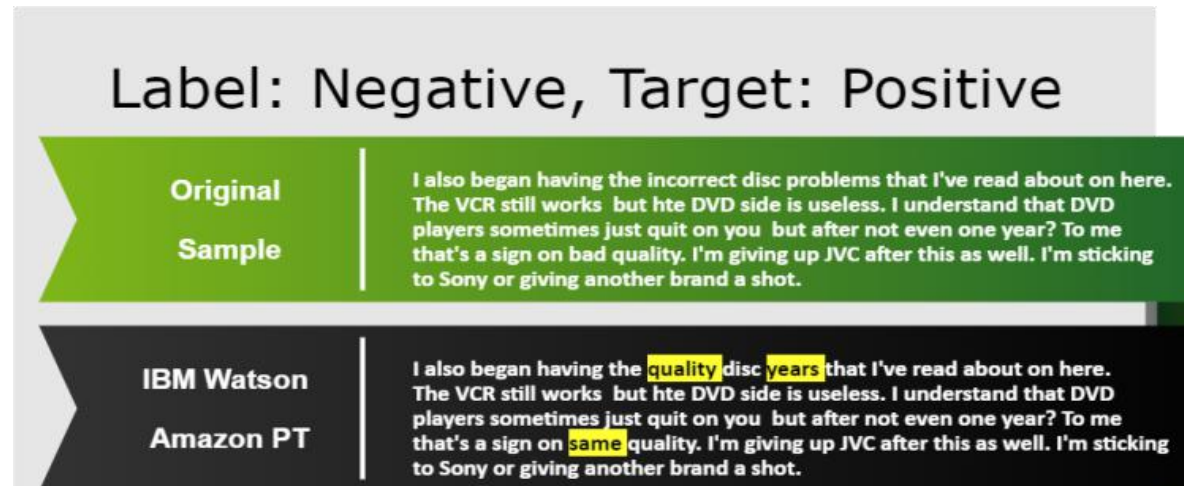


Figure 3. The original and the corrupted sample using Amazon polarity table on IBM Watson NLU victim.