INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

HYDERABAD

EMNLP 2021

Punta Cana, Dominican Republic
7th – 11th November

A Strong Baseline for Query Efficient Attacks in a Black Box Setting

Rishabh Maheshwary*, Saket Maheshwary* and Vikram Pudi
International Institute of Information Technology, Hyderabad, India

Table of Contents

1. Adversarial Examples and its importance
2. Adversarial attacks in NLP
3. Our proposed attack method
4. Experiments and analysis
5. Conclusion and future work

Adversarial Examples

1. Adversarial examples are carefully crafted inputs to Deep Neural Networks (DNNs) that an attacker has intentionally designed to cause the model to make a mistake.

2. These inputs are generated by adding small perturbation to the original input such that perturbations are imperceptible to humans but they deceive DNNs.

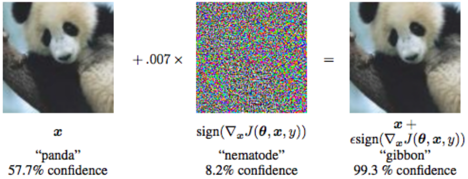
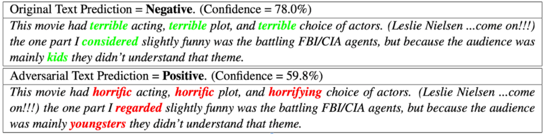

+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} +$
$\epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure: Adversarial Example in image domain



Original Text Prediction = **Negative**. (Confidence = 78.0%)

*This movie had **terrible** acting, **terrible** plot, and **terrible** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **considered** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **kids** they didn't understand that theme.*

Adversarial Text Prediction = **Positive**. (Confidence = 59.8%)

*This movie had **horrific** acting, **horrific** plot, and **horrifying** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **regarded** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **youngsters** they didn't understand that theme.*

Figure: Adversarial Example in text domain

Importance of Adversarial Examples

1. Neural Networks are used in a variety of real world applications.

2. The existence of adversarial examples pose a threat to the security of neural networks deployed in real world.

3. Hence it is essential to study the impact of adversarial examples on the decisions made by neural models in a real-world scenario, where we can query the neural models a limited number of times.

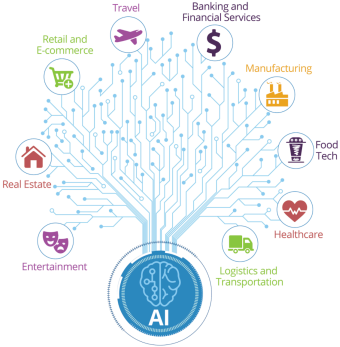


Figure: Applications of neural models

Table of Contents

1. Adversarial Examples and its importance
2. Adversarial Attack methods in NLP
3. Our proposed attack method
4. Experiments and analysis
5. Conclusion and future work

Adversarial Attacks in NLP

✓ **White Box**: Attacks requiring access to target model parameters and gradients.

✓ **Black Box Score**: Requires only the confidence scores of the target model.

✓ **Black Box Decision**: Require only the top predicted label of the target model.

In this paper we focus on the **score based black box setting** where we have access only to the **confidence scores of the target model**. We do not have access to the architecture, parameters and training data of the target model.
Also, we introduce **word level perturbations** to generate adversarial attacks in the above defined setting.

Components of an Attack method

1. Search Space

A set of constrains and transformations for each input word.

- WordNet
- HowNet + POS consistency
- Counter-fitted Embeddings + POS consistency
- Counter-fitted Embeddings + Language Modelling + POS consistency

2. Search Method

Search algorithm to find adversarial examples in the search space.

- Genetic algorithm (GA) based attack (Alzantot et al.,2018)
- Particle swarm optimization (PSO) (Zang et al.,2020)
- Probability weighted word saliency (PWWS) (Ren et al.,2019)
- TextFooler (Jin et al.,2018)

Drawbacks of Existing Methods

1. PSO and GA use combinatorial optimization algorithms as search methods which are extremely slow and take massive amount of queries to attack.

2. PWWS and TextFooler use word ranking methods which are either inefficient or suffers from a low attack success rate.

3. Also, word ranking methods often delete a word or replace it with <UNK> token which often modify the semantics of input while ranking.

4. Further, prior methods evaluate their methods only on a single search space and do not maintain a consistent search space while comparing their method with other methods.

Table of Contents

1. Adversarial Examples and its importance
2. Adversarial Attack methods in NLP
3. Our proposed attack method
4. Experiments and analysis
5. Conclusion and future work

Proposed Search Method

Our search method uses generates adversarial examples using a two step process

1. Word Ranking

Scores each word based upon **(1) how important it is for classification** and **(2) how its replacement can impact the decision of the target model.**

2. Word Substitution

It generates the final adversarial example for the input text by substituting the words with their synonyms in the order retrieved by the word ranking step.

Word Ranking — Attention based Scoring

- To identify important words for classification we use attention to score each word.
- The input is passed through a pre-trained attention model to get attention scores of each word. The scores are computed using Hierarchical Attention Network and Decompose Attention Model for text classification and entailment tasks respectively.
- Also, unlike prior methods, we do not rank each word by removing it from the input (or replacing it with a UNK token), preventing us from altering semantics of the input.

Attention Example

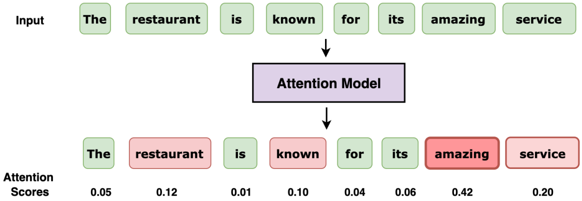


Figure: Attention step

This method is very efficient as instead of querying the target model every time to score each word, this step scores all words together in a single pass thus, significantly reducing the number of queries required.

Word Ranking — LSH based Scoring

LSH is a technique used for finding nearest neighbours in high dimensional spaces. It takes an input, a vector x and computes its hash $h(x)$ such that similar vectors gets the same hash with high probability and dissimilar ones do not. LSH differs from cryptographic hash methods as it aims to maximize the collisions of similar items.

- It assigns high scores to words whose replacement will highly influence the decision of the target model.
- Each word is replaced with every synonym from the search space and the generated perturbed text inputs are encoded using a sentence encoder.
- Then LSH is used to map similar perturbed text vectors to same bucket.
- An input is sampled from each bucket and queried to target model. The maximum change in confidence score of the original label among all the queried inputs is the score assigned to each word. This process is repeated for all input words.

LSH Example

LSH maps highly similar perturbed inputs (which will impact the target model almost equally) to the same bucket. This drastically reduces the number of queries required for ranking each word and results in a highly effective and efficient ranking method.

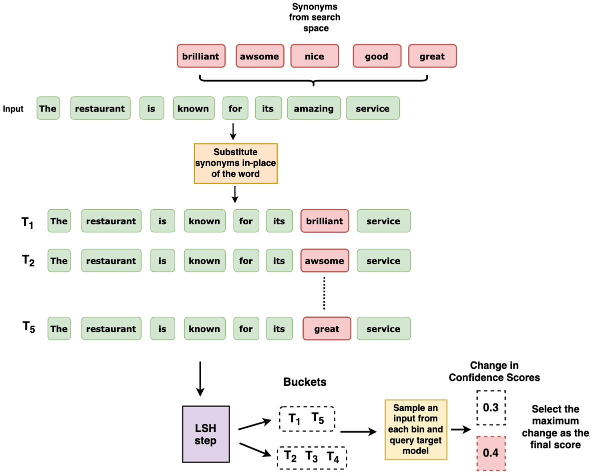
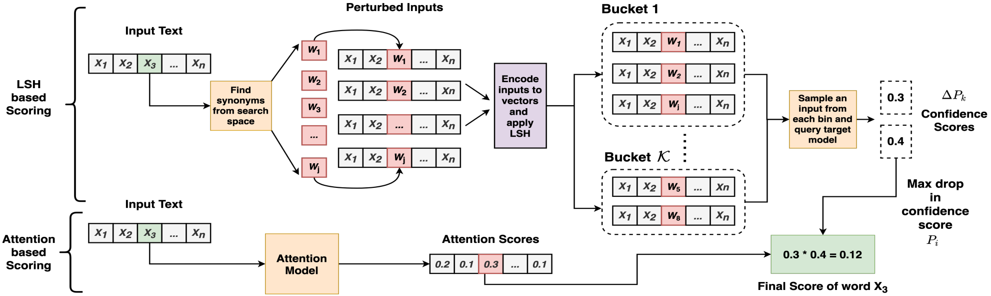


Figure: LSH step

Word Ranking — Final score calculation

After obtaining the attention scores and the scores from synonym words for each index (calculated using LSH), we multiply the two to get the final score for each word. All the words are sorted in descending order based upon the score.



LSH based Scoring
Input Text
x_1 x_2 x_3 ... x_n
Find synonyms from search space
Perturbed Inputs
w_1 w_2 w_3 ... w_j
x_1 x_2 w_1 ... x_n
x_1 x_2 w_2 ... x_n
x_1 x_2 ... ... x_n
x_1 x_2 w_j ... x_n
Encode inputs to vectors and apply LSH
Bucket 1
x_1 x_2 w_1 ... x_n
x_1 x_2 w_2 ... x_n
x_1 x_2 w_j ... x_n
Bucket $\mathcal{K}$
x_1 x_2 w_5 ... x_n
x_1 x_2 w_8 ... x_n
Sample an input from each bin and query target model
0.3
0.4
ΔP_k
Confidence Scores
Max drop in confidence score
P_i
Attention based Scoring
Input Text
x_1 x_2 x_3 ... x_n
Attention Model
Attention Scores
0.2 0.1 0.3 ... 0.1
0.3 * 0.4 = 0.12
Final Score of word X_3

Figure: Scoring of each input word using attention mechanism and Locality Sensitive Hashing (LSH).

Word Substitution

- It generates the final adversarial example for the input text by perturbing the words in the order retrieved by the word ranking step.
- Each word is replaces with all synonyms from the search space.
- It filters perturbed texts which do not satisfy search space constraints.
- Then each perturbed text input is passed to the target model and the input which alters the prediction of the target model is selected as the final adversarial example.
- In case the prediction is not altered the perturbed input which causes the maximum change in the confidence score of the target class is selected as the best adversarial example and the process is repeated for the remaining ranked words.

Word Substitution — Example

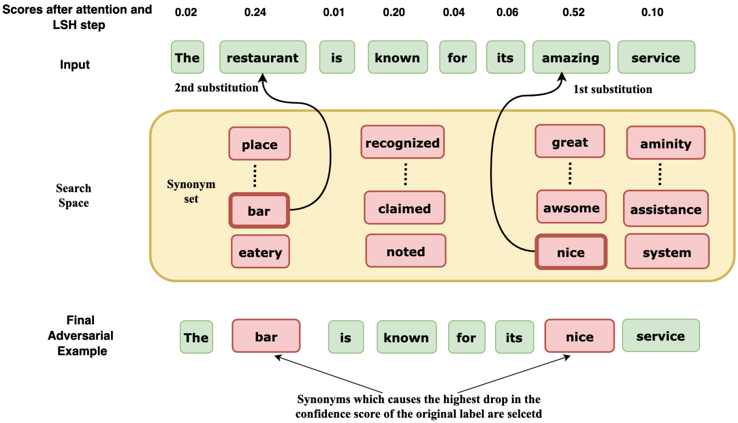


Table of Contents

1. Adversarial Examples and its importance
2. Adversarial Attack methods in NLP
3. Our proposed attack method
4. Experiments and analysis
5. Conclusion and future work

Experiments - Tasks and Baselines

2 NLP Tasks

- Classification (IMDB, Yelp)
- NLI (MNLI)

4 Search Spaces

- WordNet
- HowNet + POS consistency
- Counter-fitted Embeddings + POS consistency
- Counter-fitted Embeddings + Language Modelling + POS consistency

2 SOTA Models

- BERT
- Word-LSTM

4 Baselines

- Genetic algorithm (GA)
- Particle swarm optimization (PSO)
- Probability weighted word saliency. (PWWS)
- TextFooler (TF)

Experiments - Metrics

- **Attack success rate** — Ratio of successful attacks to total number of attacks.
- **Number of queries** — Average number of queries required to attack.
- **Perturbation rate** — Percentage of words substituted in an input.
- **Grammatical error rate** — Average grammatical error increase rate.
- **Human Evaluation** — Qualitative analysis of generated adversarial examples.

Results

Model	Attack	IMDB		Yelp		MNLI	
		Qrs	Suc%	Qrs	Suc%	Qrs	Suc%
BERT	PSO	81350.6	**99.0**	73306.6	**93.2**	4678.5	**57.97**
	Ours	**737**	97.4	**554.2**	91.6	**97.2**	56.1
LSTM	PSO	52008.7	**99.5**	43671.7	**95.4**	2763.3	**67.8**
	Ours	**438.1**	**99.5**	**357.6**	94.75	**79.8**	66.4

(a) Comparison with PSO.

Model	Attack	IMDB		Yelp		MNLI	
		Qrs	Suc%	Qrs	Suc%	Qrs	Suc%
BERT	Gen	7944.8	66.3	6078.1	**85.0**	1546.8	**83.8**
	Ours	**378.6**	**71.1**	**273.7**	84.4	**43.4**	81.9
LSTM	Gen	3606.9	97.2	5003.4	**96.0**	894.5	**87.8**
	Ours	**224**	**98.5**	**140.7**	95.4	**39.9**	86.4

(b) Comparison with Genetic Attack.

Model	Attack	IMDB		Yelp		MNLI	
		Qrs	Suc%	Qrs	Suc%	Qrs	Suc%
BERT	PWWS	1583.9	**97.5**	1013.7	**93.8**	190	**96.8**
	Ours	**562.9**	96.4	**366.2**	92.6	**66.1**	95.1
LSTM	PWWS	1429.2	**100.0**	900.0	**99.1**	160.2	**98.8**
	Ours	**473.8**	**100.0**	**236.3**	**99.1**	**60.1**	98.1

(c) Comparison with PWWS.

Model	Attack	IMDB		Yelp		MNLI	
		Qrs	Suc%	Qrs	Suc%	Qrs	Suc%
BERT	TF	1130.4	**98.8**	809.9	**94.6**	113	85.9
	Ours	**750**	98.4	**545.5**	93.2	**100**	**86.2**
LSTM	TF	544	**100.0**	449.4	**100.0**	105	95.9
	Ours	**330**	**100.0**	**323.7**	**100.0**	**88**	**96.2**

(d) Comparison with TextFooler (TF).

Table: Result comparison. Succ% is the attack success rate and Qrs is the average query count. Note as each baseline uses a different search space, our method will yield different results when comparing with each baseline.

Results - Contd.

Model	Attack	IMDB		Yelp		MNLI	
		Pert%	I%	Pert%	I%	Pert%	I%
BERT	PSO	4.5	0.20	10.8	0.30	8.0	3.5
	Ours	**4.2**	**0.10**	**7.8**	**0.15**	**7.1**	**3.3**
LSTM	PSO	2.2	0.15	7.7	0.27	**6.7**	**1.27**
	Ours	**2.0**	**0.11**	**4.9**	**0.15**	6.8	1.3

(a) Comparison with PSO.

Model	Attack	IMDB		Yelp		MNLI	
		Pert%	I%	Pert%	I%	Pert%	I%
BERT	Gen	**6.5**	1.04	11.6	1.5	**8.7**	**1.9**
	Ours	6.7	**1.02**	**10.5**	**1.49**	9.2	2.1
LSTM	Gen	4.1	0.62	8.6	1.3	7.7	2.5
	Ours	**3.19**	**0.56**	**6.2**	**1.05**	**8.2**	**2.1**

(b) Comparison with Genetic Attack.

Model	Attack	IMDB		Yelp		MNLI	
		Pert%	I%	Pert%	I%	Pert%	I%
BERT	PWWS	**5.2**	**0.74**	**7.3**	**1.5**	**7.1**	1.71
	Ours	7.5	0.9	9.9	1.9	9.6	**1.48**
LSTM	PWWS	2.3	**0.3**	**4.8**	**1.29**	**6.6**	**1.5**
	Ours	**1.9**	0.4	5.5	**1.29**	7.8	2.1

(c) Comparison with PWWS.

Model	Attack	IMDB		Yelp		MNLI	
		Pert%	I%	Pert%	I%	Pert%	I%
BERT	TF	9.0	1.21	5.2	**1.1**	11.6	**1.23**
	Ours	**6.9**	**0.9**	**6.6**	1.2	**11.4**	1.41
LSTM	TF	**2.2**	2.3	5.7	2.06	**9.8**	1.7
	Ours	2.4	**1.5**	**5.3**	**1.5**	10.1	**1.4**

(d) Comparison with TextFooler (TF).

Table: Result comparison. Pert% is the perturbation and I% is the average grammatical error increase.

Query Analysis

Comparison of our method with the baselines under a fixed query budget

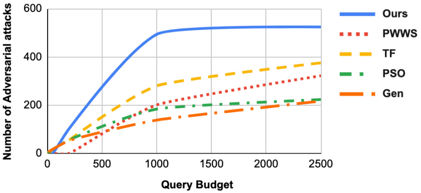


Figure: Adversarial attacks on BERT-IMDB

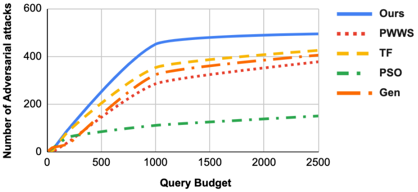


Figure: Adversarial attacks on Yelp-IMDB

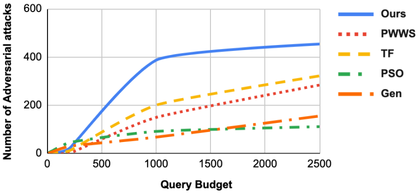


Figure: Adversarial attacks on BERT-Yelp

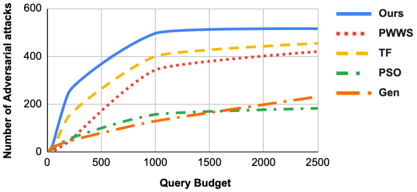


Figure: Adversarial attacks on LSTM-Yelp

Ablation study and Additional Analysis

Dataset	Random			Only Attention			Only LSH			Both LSH and Attention		
	Suc%	Pert%	Qrs	Suc%	Pert%	Qrs	Suc%	Pert%	Qrs	Suc%	Pert%	Qrs
IMDB	90.5	13.3	507.9	94.0	9.3	851.3	95.3	8.0	694.9	**96.4**	**7.5**	**562.9**
Yelp	87.3	15.0	305.9	91.0	11.0	550.0	90.2	10.2	475.2	**92.6**	**9.8**	**366.2**
MNLI	88.8	14.3	60.1	92.4	11.7	121.2	94.3	10.1	100.1	**95.1**	**9.6**	**66.1**

Table: Ablation Study of attention mechanism and LSH on WordNet search space.

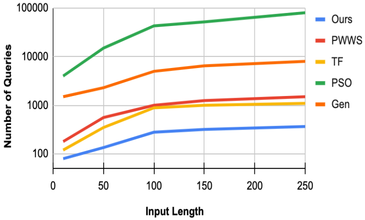


Figure: Queries taken vs number of words in input

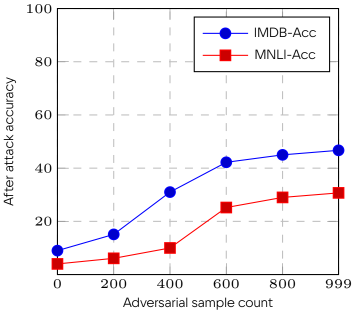
Transfer	Accurracy	IMDB	MNLI
BERT → LSTM	Original	90.9	85.0
	Transferred	**72.9**	**60.6**
LSTM → BERT	Original	88.0	70.1
	Transferred	**67.7**	**62.1**

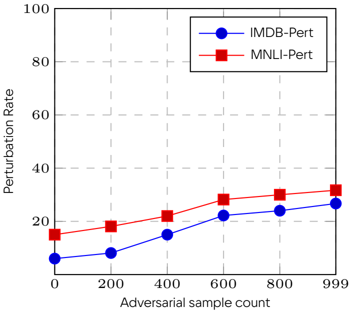
Table: Transferability analysis

Adversarial Learning

Target models becomes robust as more adversarial samples are augmented.

We generated adversarial samples on the original training set of the target models and then augmented the original training set with the generated adversarial examples. As the count of augmented adversarial examples increases the models become difficult to attack.





Human Evaluation

- We sampled 25% of original instances and their corresponding adversarial examples generated on BERT for IMDB and MNLI datasets on WordNet search space.
- We asked 3 human judges to evaluate each sample based upon the following criteria:
  - **Classification result**: Assign classification labels to generated adversarial examples.
  - **Semantic Similarity**: Assign a score of 0, 0.5 or 1 based on how well the adversarial examples were able to retain the meaning of their original counterparts.
  - **Grammatical Correctness**: Assign a score the range 1 to 5 for grammatical correctness of each adversarial example.

Evaluation criteria	IMDB	MNLI
Classification result	94%	91%
Grammatical Correctness	4.32	4.12
Semantic Similarity	0.92	0.88

Table: Demonstrates scores given by judges (scores are averaged)

Some Generated Adversarial Examples

Examples	Prediction
The movie has an excellent screenplay (the situation is credible, the action has pace), first-class [fantabulous] direction and acting (especially the 3 leading actors but the others as well -including the mobster, who does not seem to be a professional actor). I wish [want] the movie, the director and the actors success.	Positive → Negative
Local-international gathering [assembly] spot [stain] since the 1940s. One of the coolest pubs on the planet. Make new friends from all over the world, with some of the best [skilful] regional and imported beer selections in town.	Postive → Negative
It's weird, wonderful, and not neccessarily [definitely] for kids.	Negative → Positive.
Premise: If we travel for 90 minutes, we could arrive [reach] arrive at larger ski resorts. **Hypothesis**: Larger ski resorts are 90 minutes away.	Entailment → Neutral
Premise: Basically [Crucially], to sell myself. **Hypothesis**: Selling myself is a very important thing.	Contradict → Neutral

Table: The actual word is highlighted green and substituted word is in square brackets colored red.

Table of Contents

1. Adversarial Examples and its importance
2. Adversarial Attack methods in NLP
3. Our proposed attack method
4. Experiments and analysis
5. Conclusion and future work

Conclusion

- We proposed a query efficient attack that generates plausible adversarial examples on text classification and entailment tasks.
- Extensive experiments across *three* search spaces and *four* baselines shows that our attack generates high quality adversarial examples with significantly lesser queries.
- Further, we demonstrated that our attack has a much higher success rate in a limited query setting, thus making it extremely useful for real world applications.

Future Work

- The existing word level scoring methods can be extended to sentence level.
- Also, the attention scoring model used can be trained on different datasets to observe how the success rate and the query efficiency gets affected.
- New attack methods can rely on transferability approaches or Reinforcement learning methods to craft attacks which may further reduce the query count.
- Furthermore, existing attack methods can be evaluated against various defense methods to compare the effectiveness of different search methods.

Important Links

- Paper
  — https://arxiv.org/abs/2109.04775
- Code
  — https:/github.com/rishabhmaheshwary/query-attack
- Slides and Poster
  — https://drive.google.com/drive/folders/1HM97Xy7U5-A6UDLAbgbGhqpHsTqTjsRJ?usp=sharing
- Textattack implementation
  — github.com/RishabhMaheshwary/TextAttack/tree/query-attack.

In case of any questions open an issue at the above Git repository or reach out to us at rishabh.maheshwary@research.iiit.ac.in or rf.rishabh@gmail.com.