Improving the Quality of Adversarial Examples via Contrastive Learning and Pretraining

LYU2106 Final Year Project Term 2 Presentation

Yung-chieh Huang (1155120711)

Agenda

- Introduction
- Objective
- A recap of last term
- Contribution of this term
- Methodology
- Baselines
- Experiments
- Conclusion

Introduction – Adversarial Attack

- Adversarial attack is an approach to test the robustness of machine learning models, by intentionally applying perturbations to make the models misclassify.
- To ensure security in real-life applications.

Introduction – Adversarial Attack for Text

- Adversarial examples are generated by attack models, by replacing words in a sentence.
- A well-crafted adversarial example should have minimum perturbations and preserve the structure and characteristics of the original.

Perfect performance by the actor	Positive (99%)
Spotless performance by the actor	Negative (100%)

Objective

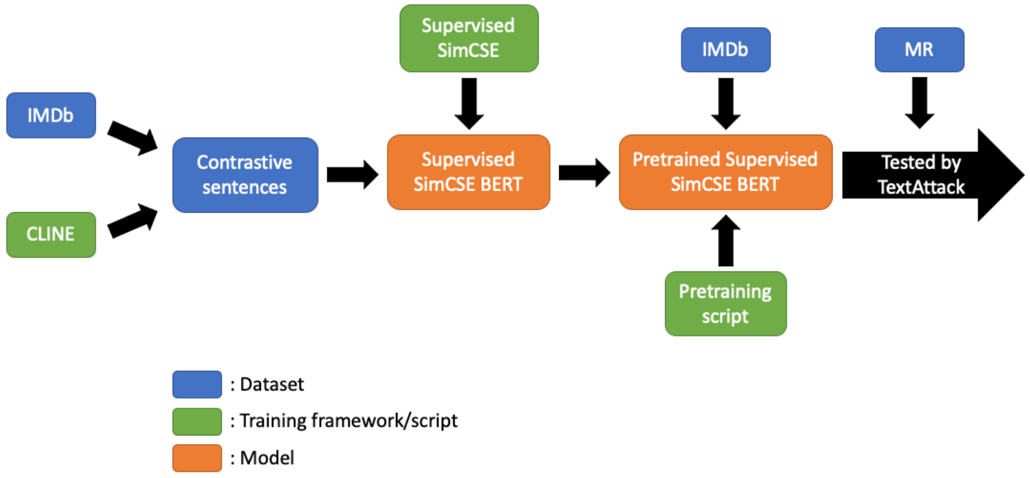
- Generate examples to be free from opposite semantic or out-of-context replacements and maintain fluency.
- Higher successful attack rate and lower perturbation than baseline attack models.

Original sentence	no amount of good intentions is able to overcome the triv-iality of the story	Negative (100%)
Adversarial example	no amount of good intentions is able to overcome the beauty of the story	Positive (99%)

Original sentence	watching spirited away is like watching an eastern imagi-nation explode	Positive (99%)
Adversarial example	watching spirited away is like watching an eastern maga-zine explode	Negative (100%)

Recap – Conclusion from last term

- Pretrain on domain-specific datasets to generate a domain-specific attack model to avoid out-of-context replacements.
- Contrastive learning can distinguish synonyms and antonyms in the embedding space, which helps avoid opposite semantic replacements.



Recap – Conclusion from last term

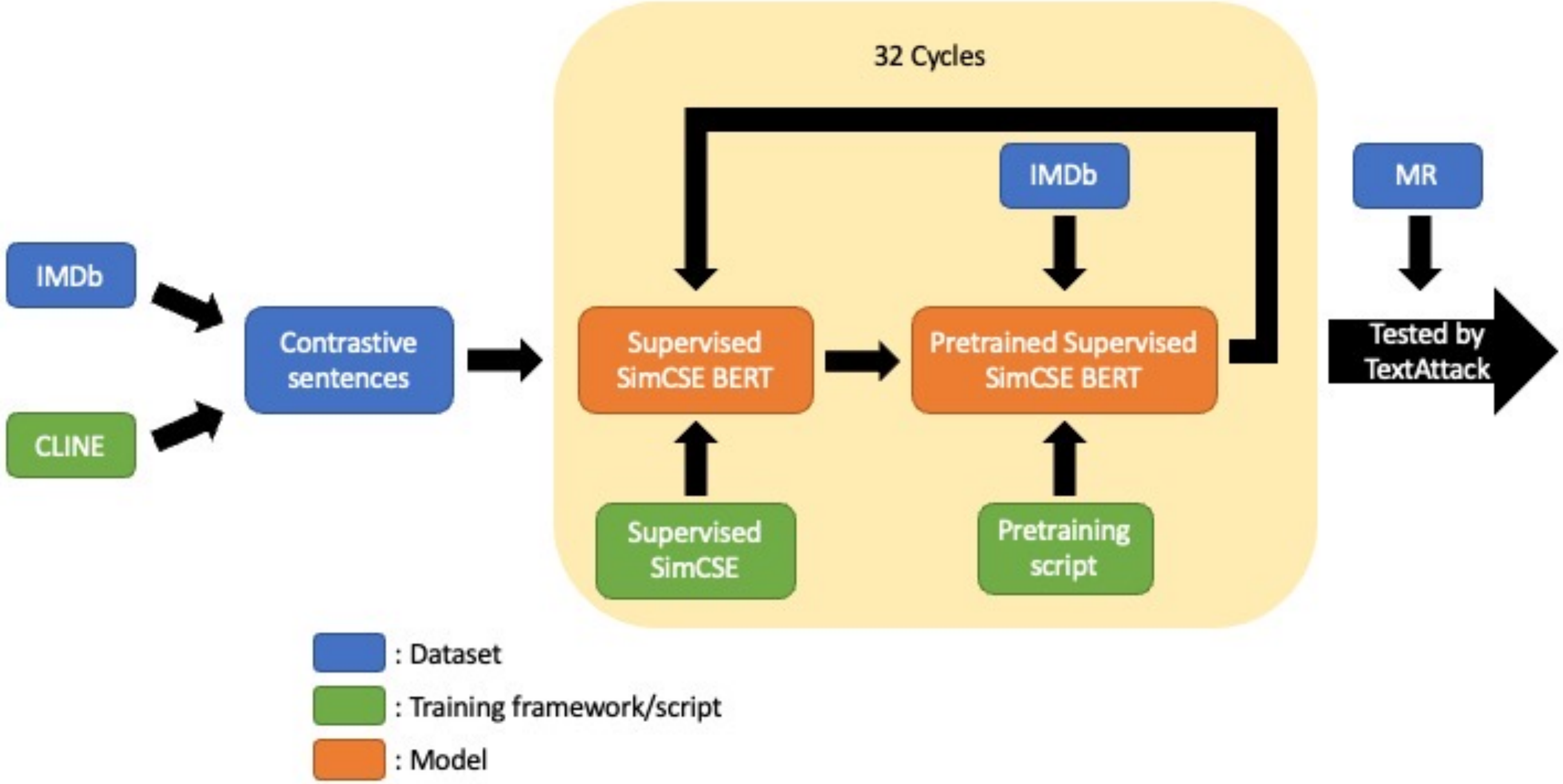
Dataset: MR		
	BAE	Ours
Number of successful attacks	473	**475**
Number of failed attacks	365	**363**
Number of skipped attacks	162	162
Original accuracy	83.8%	83.8%
Accuracy under attack	36.5%	**36.3%**
Attack success rate	56.44%	**56.68%**
Average perturbed word %	13.91%	**13.37%**
Average number of words per input	18.64	18.64
Average number of queries	63.49	**63.19**

Dataset: MR						
	BAE	Ours (50,000)	Ours (25,000)	Ours (5,000)	Ours (2,500)	Ours (0)
Number of successful attacks	473	471	473	487	**501**	411
Number of failed attacks	365	367	365	351	**337**	427
Number of skipped attacks	162	162	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	36.5%	36.7%	36.5%	35.1%	**33.7%**	42.7%
Attack success rate	56.44%	56.21%	56.44%	58.11%	**59.79%**	49.05%
Average perturbed word %	13.91%	13.19%	**13.13%**	13.58%	13.17%	14.85%
Average number of words per input	18.64	18.64	18.64	18.64	18.64	18.64
Average number of queries	63.49	64.27	64.05	64.01	62.96	54.93

Contribution of this term

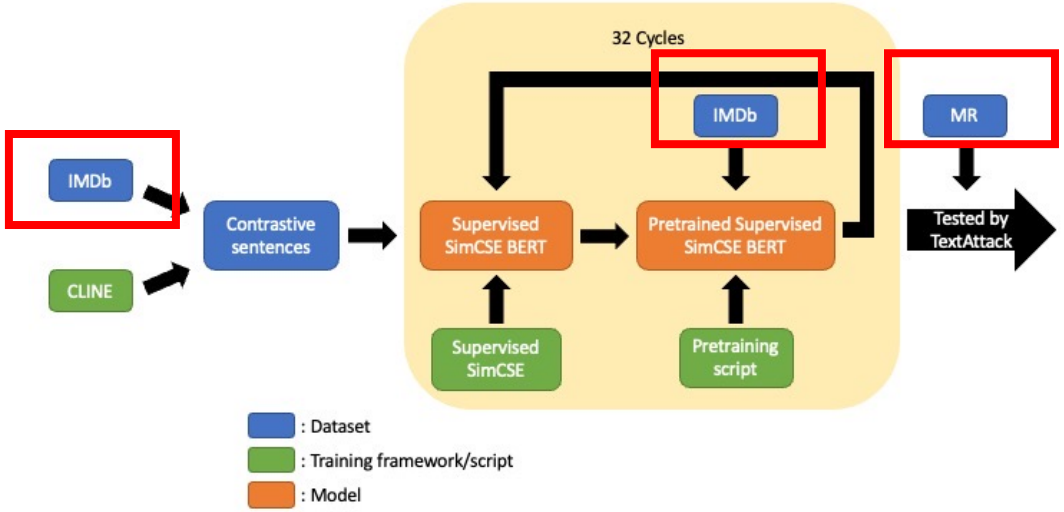
- We create our own contrastive sentence pairs to improve the performance of contrastive learning.
- We are the first to propose an iterative training method to combine contrastive learning and pretraining.
- This iterative training method balances the quality of generated adversarial examples and the goal to increase the attack success rate well.
- It largely improves the overall attack performance.

Methodology



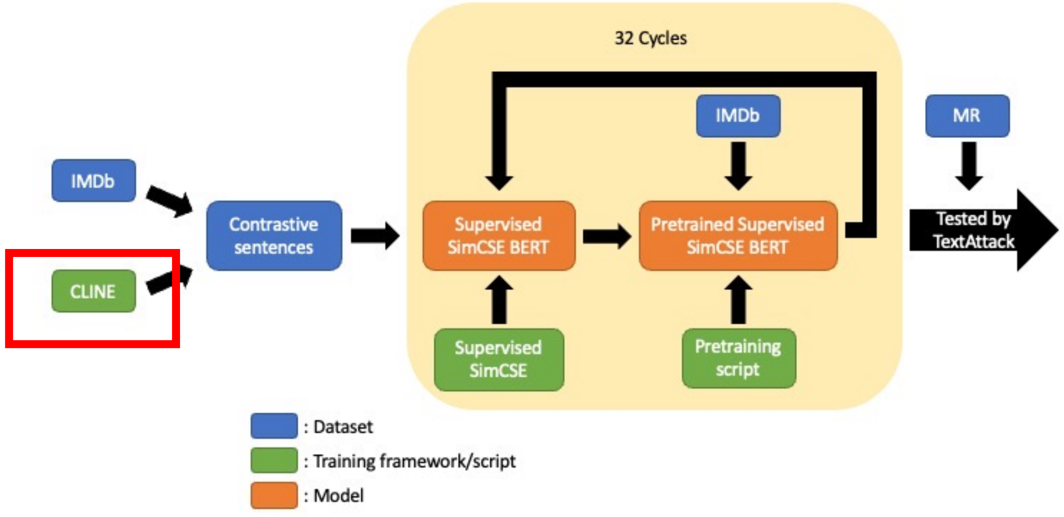
Methodology - Datasets

- IMDb (Mass et al. 2011): 25,000 highly polar movie reviews for training, 25,000 for testing, and additional 50,000 unlabeled data.
- MR (Pang and L. Lee 2005): 5,331 positive and 5,331 negative reviews from Rotten Tomatoes.



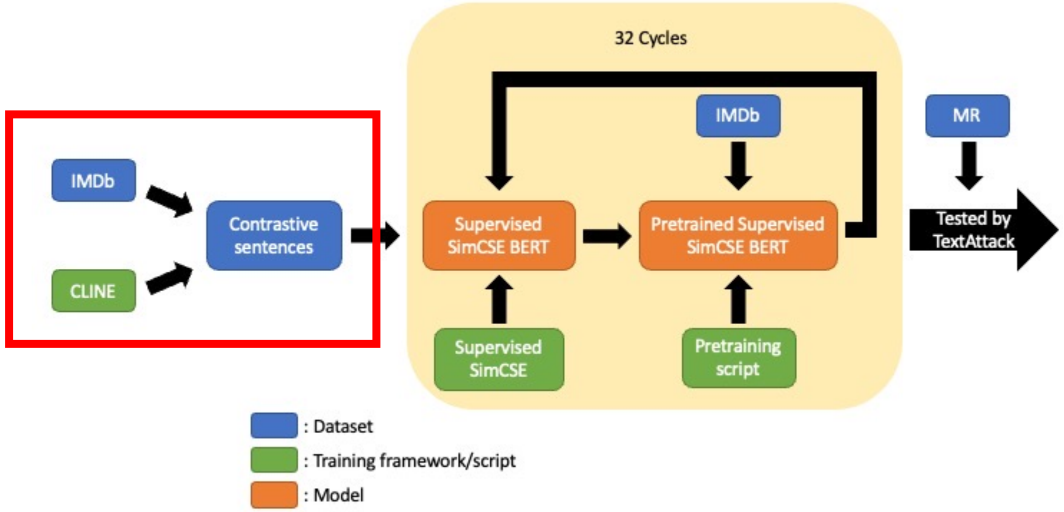
Methodology – CLINE (Wang et al. 2021)

- Generates semantically similar sentences by replacing words with synonyms.
- Generates semantically opposite sentences by replacing words with antonyms or random words.



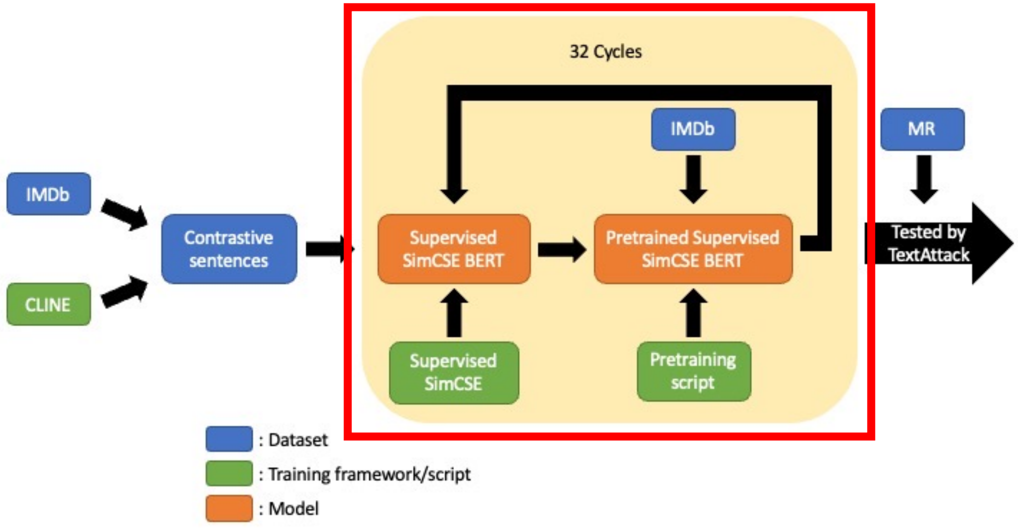
Methodology – CLINE data augmentation

- Create our own contrastive sentence pairs of different replace ratios:
  - 0.05
  - 0.1
  - 0.2
  - 0.4
  - 0.5



Methodology – Iterative training

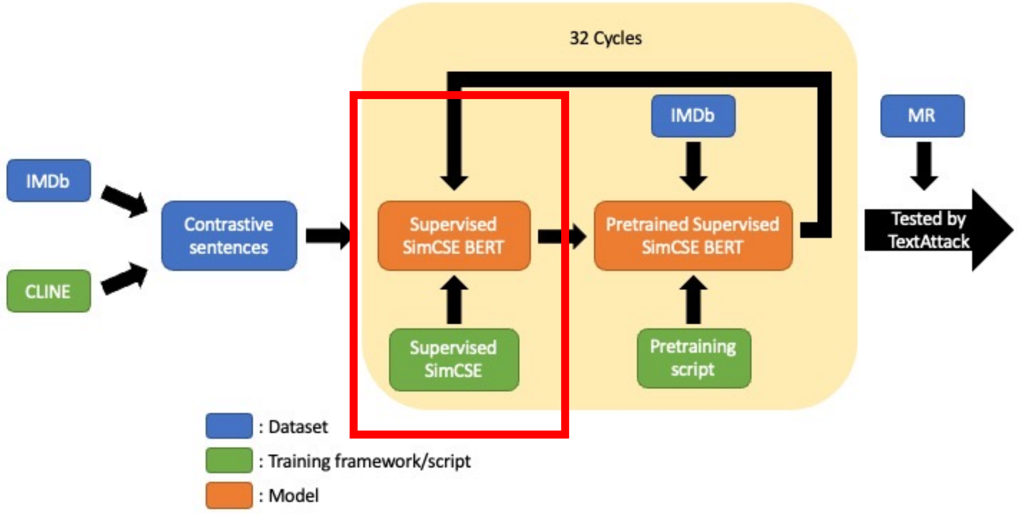
- Equally divide the training process into 32 cycles.
- In each cycle:
  - 125,000/32 contrastive sentence pairs.
  - Pretrain 2,500/32 steps.



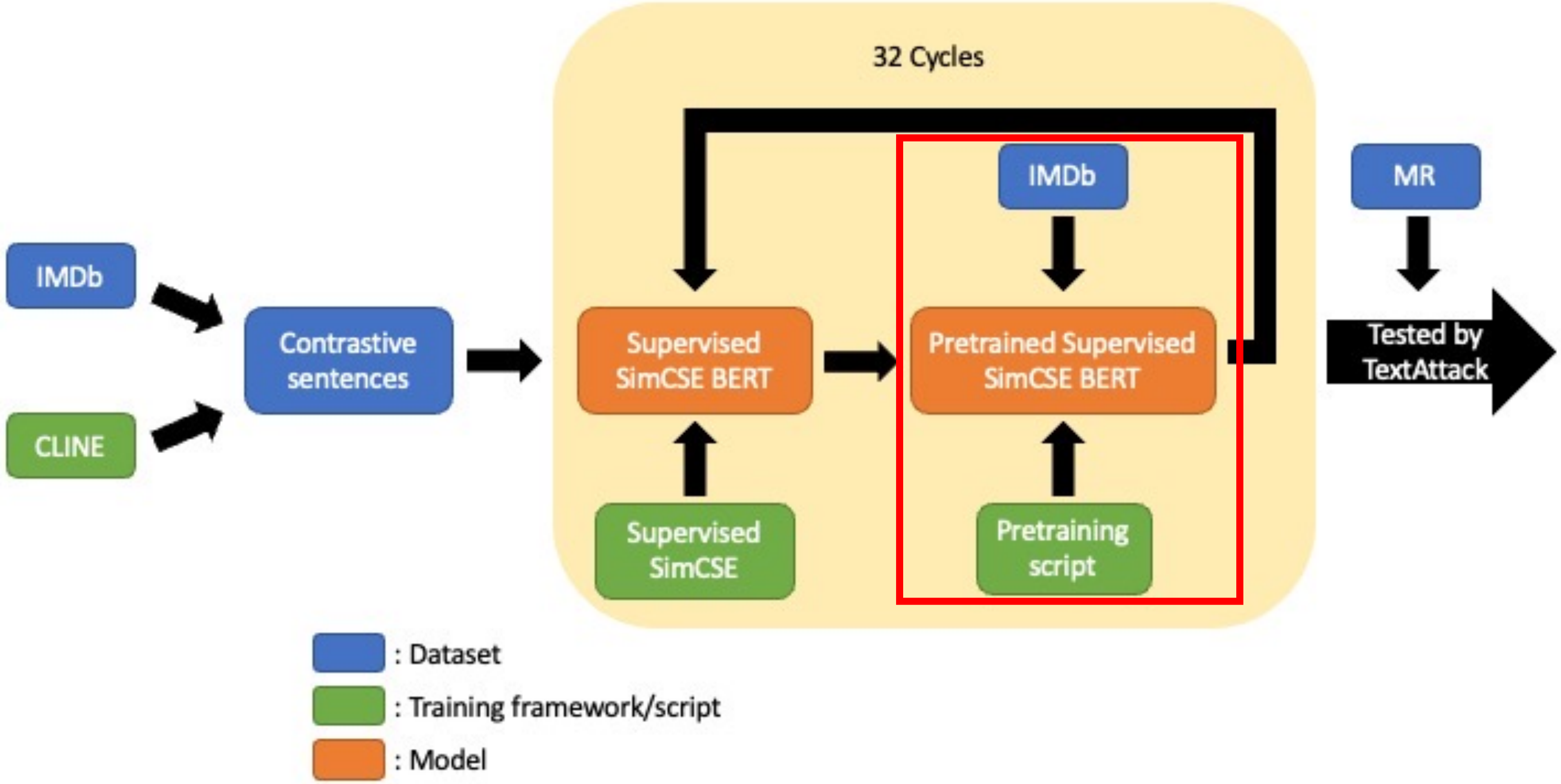
Methodology – SimCSE (T. Gao, Yao, and Chen 2021)

- Pulls semantically close neighbors together and pushes apart non-neighbors.
- The training objective is defined by:

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}$$

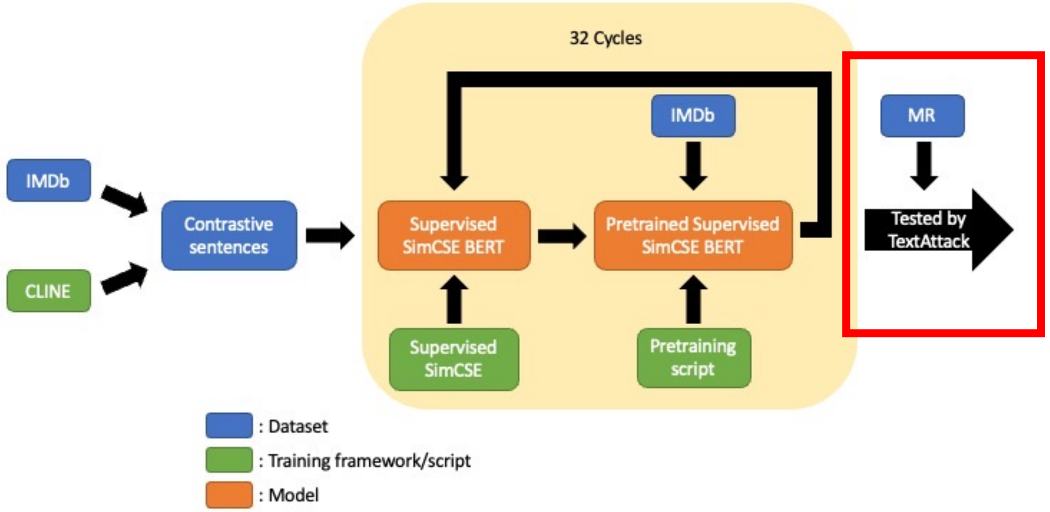


Methodology - Pretraining



Methodology – TextAttack (Morris et al. 2020)

- A framework to evaluate different NLP attacks.
- Generates adversarial examples from a given dataset using an attack recipe and attack a victim model.



Baselines

- BAE (Garg and Ramakrishnan 2020): Inserts/Replaces tokens using BERT MLM.
- PWWS (Ren et al. 2019): Uses word saliency and classification probability to determine the word replacing order. Applies the synonym replacement strategy greedily to each word in that order.
- TextFooler (Jin et al. 2020): A strong and commonly used baseline. Uses multiple rule-based strategies.

Experiments – CLINE data augmentation

- Evaluate with different replace ratios.

Dataset: MR					
	Ours (0.05)	Ours (0.05 + 0.1)	Ours (0.05 + 0.1 + 0.2)	Ours (0.05 + 0.1 + 0.2 + 0.4)	Ours (0.05 + 0.1 + 0.2 + 0.4 + 0.5)
Number of successful attacks	500	504	499	505	**508**
Number of failed attacks	338	334	339	333	**330**
Number of skipped attacks	162	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	33.8%	33.4%	33.9%	33.3%	**33%**
Attack success rate	59.67%	60.14%	59.55%	60.26%	**60.62%**
Average perturbed word %	13.69%	13.45%	13.37%	13.22%	**13.18%**
Average number of words per input	18.64	18.64	18.64	18.64	18.64
Average number of queries	63.57	64.42	63.22	62.3	62.58

Experiments – CLINE data augmentation

Dataset: MR				
	BAE	Ours (pre-training only)	Ours (con-trastive pretrain 2,500)	Ours (0.05 + 0.1 + 0.2 + 0.4 + 0.5)
Number of successful attacks	473	475	501	**508**
Number of failed attacks	365	363	337	**330**
Number of skipped attacks	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	36.5%	36.3%	33.7%	**33.0%**
Attack success rate	56.44%	56.68%	59.79%	**60.62%**
Average perturbed word %	13.91%	13.37%	**13.17%**	13.18%
Average number of words per input	18.64	18.64	18.64	18.64
Average number of queries	63.49	63.19	62.96	62.58

Experiments – Iterative training

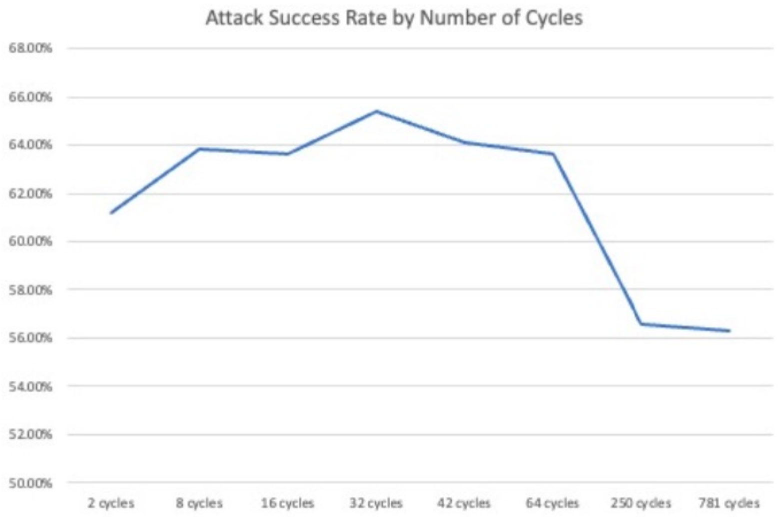
- Evaluate with different number of cycles.

Dataset: MR				
	Ours (2 cycles)	Ours (8 cycles)	Ours (16 cycles)	Ours (32 cycles)
Number of successful attacks	513	535	533	**548**
Number of failed attacks	325	303	305	**290**
Number of skipped attacks	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	32.5%	30.3%	30.5%	**29.0%**
Attack success rate	61.22%	63.84%	63.6%	**65.39%**
Average perturbed word %	13.4%	12.34%	12.01%	**11.83%**
Average number of words per input	18.64	18.64	18.64	18.64
Average number of queries	62.91	59.98	59.14	57.68

Dataset: MR					
	Ours (32 cycles)	Ours (42 cycles)	Ours (64 cycles)	Ours (250 cycles)	Ours (781 cycles)
Number of successful attacks	**548**	537	533	474	472
Number of failed attacks	**290**	301	305	364	366
Number of skipped attacks	162	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	**29.0%**	30.1%	30.5%	36.4%	36.6%
Attack success rate	**65.39%**	64.08%	63.6%	56.56%	56.32%
Average perturbed word %	**11.83%**	12.21%	12.22%	12.97%	13.31%
Average number of words per input	18.64	18.64	18.64	18.64	18.64
Average number of queries	57.68	58.0	54.43	36.65	37.43

Experiments – Iterative training

- An attack model is under-fitted without iterative training.
- Any more than 32 cycles will shows signs of over-fitting.
- Our method can reduce the negative effect of excessive pretraining on contrastive learning.



Experiments – Iterative training

Dataset: MR				
	Ours (pre-training only)	Ours (con-trastive pretrain 2,500)	Ours (0.05 + 0.1 + 0.2 + 0.4 + 0.5)	Ours (32 cycles)
Number of successful attacks	475	501	508	**548**
Number of failed attacks	363	337	330	**290**
Number of skipped attacks	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	36.3%	33.7%	33.0	**29.0%**
Attack success rate	56.68%	59.79%	60.62	**65.39%**
Average perturbed word %	13.37%	13.17%	13.18%	**11.83%**
Average number of words per input	18.64	18.64	18.64	18.64
Average number of queries	63.19	62.96	62.58	**57.68**

Dataset: MR				
	BAE	PWWS	TextFooler	Ours (32 cycles)
Number of successful attacks	473	434	531	**548**
Number of failed attacks	365	404	307	**290**
Number of skipped attacks	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	36.5%	40.4%	30.7%	**29.0%**
Attack success rate	56.44%	51.79%	63.37%	**65.39%**
Average perturbed word %	13.91%	16.0%	20.78%	**11.83%**
Average number of words per input	18.64	18.64	18.64	18.64
Average number of queries	63.49	62.44	58.36	57.68

Experiments – Iterative training

Original sentence	one of the funnier movies in town.	Positive (94%)
BAE	one of the funnier locations in town.	Negative (97%)
PWWS	matchless of the funnier movies in town.	Negative (100%)
TextFooler	one of the funnier kino in town.	Negative (88%)
Ours (32 cycles)	one of the funnier scenes in town.	Negative (99%)

Experiments – Batch-sorted sentence pairs

- Create 16 nonidentical sentence pairs for each sentence and sort them together.

Dataset: MR		
	Ours (32 cycles)	Ours (32 cycles + batch-sorted)
Number of successful attacks	**548**	543
Number of failed attacks	**290**	295
Number of skipped attacks	162	162
Original accuracy	83.8%	83.8%
Accuracy under attack	**36.3%**	29.5%
Attack success rate	**65.39%**	64.8%
Average perturbed word %	11.83%	**11.65%**
Average number of words per input	18.64	18.64
Average number of queries	57.68	56.23

Experiments – Merged contrastive and pretraining

- Add the auxiliary MLM (masked language modelling) function to the SimCSE loss:

$$l = l_{contrastive} + \lambda \times l_{MLM}$$

- Evaluate with different MLM weights.

Dataset: MR		
	SimCSE MLM weight = 0.02	SimCSE MLM weight = 0.1
Number of successful attacks	426	398
Number of failed attacks	412	440
Number of skipped attacks	162	162
Original accuracy	83.8%	83.8%
Accuracy under attack	41.2%	44.0%
Attack success rate	50.84%	47.49%
Average perturbed word %	13.82%	14.15%
Average number of words per input	18.64	18.64
Average number of queries	54.01	57.74

Experiments – Merged contrastive and pretraining

- Modify the training script so that MLM only reads the original sentence.

Dataset: MR				
	SimCSE no MLM	SimCSE MLM weight=0.02	SimCSE MLM weight=0.1	SimCSE MLM weight=1
Number of successful attacks	**411**	379	410	385
Number of failed attacks	**427**	459	428	453
Number of skipped attacks	162	162	162	162
Original accuracy	83.8%	83.8%	83.8%	83.8%
Accuracy under attack	**42.7%**	45.9%	42.8%	45.3%
Attack success rate	**49.05%**	45.23%	48.93%	45.94%
Average perturbed word %	14.85%	14.47%	**14.0%**	14.23%
Average number of words per input	18.64	18.64	18.64	18.64
Average number of queries	54.93	53.78	54.49	52.9

Experiments – Merged contrastive and pretraining

- Apply gradient accumulation to eliminate over-fitting.

Dataset: MR		
	SimCSE no MLM	SimCSE MLM weight=0.1 Gradient Accumulation=100
Number of successful attacks	**411**	391
Number of failed attacks	**427**	447
Number of skipped attacks	162	162
Original accuracy	83.8%	83.8%
Accuracy under attack	**42.7%**	44.7%
Attack success rate	**49.05%**	46.66%
Average perturbed word %	14.85%	**14.52%**
Average number of words per input	18.64	18.64
Average number of queries	54.93	59.7

Experiments – Merged contrastive and pretraining

- Use separate datasets for contrastive learning and MLM.

Dataset: MR		
	SimCSE no MLM	SimCSE MLM weight=0.1 Gradient Accumulation=10
Number of successful attacks	**411**	392
Number of failed attacks	**427**	446
Number of skipped attacks	162	162
Original accuracy	83.8%	83.8%
Accuracy under attack	**42.7%**	44.6%
Attack success rate	**49.05%**	46.78%
Average perturbed word %	14.85%	**14.82%**
Average number of words per input	18.64	18.64
Average number of queries	54.93	61.21

Experiments – Merged contrastive and pretraining

- MLM affects SimCSE’s ability to learn a good representation.
- Merging the two is like cutting the process into countless mini-cycles, which can cause over-fitting.
- The iterative training remains to be our best training method.

Conclusion

- Out-of-context replacements exist because attack models are too general. We make the model domain-specific by pretraining on task-related datasets.
- Opposite semantic replacements are caused by the embedding space of language models, so we alter the embedding space by doing contrastive learning.
- Data augmentation to increase the data diversity.
- Apply the iterative training method to maximize the efficacy.

Thank you