Adversarial Attacks and Defense for Conversation Entailment Task

Zhenning Yang
University of Michigan
znyang@umich.edu

Ryan Krawec
University of Michigan
rkrawec@umich.edu

Liang-Yuan Wu
University of Michigan
lyuanwu@umich.edu

1 Introduction

Transformers are super large language models that are proved to be very powerful on different NLP tasks (Wolf et al., 2019). However, there are still many ways to attack the model with very low costs. How to defend the model becomes an important problem. In our work, we treat adversarial attack results as a new (unseen) domain of the model, and we frame the defending problem into how to improve the robustness of the model on the new domain. We focus on the task of conversation entailment, where multi-turn natural language dialogues are the premise, and the transformer model is fine-tuned to predict whether a given hypothesis about the given dialogue is true or false. The adversary would attack the hypothesis to fool the model to make the wrong predictions. We apply synonym-swapping as the attack method. To show the robustness of the model, we implement some fine-tuning strategies and propose the embedding perturbation loss as a method to improve the robustness of the model. Finally, we show the importance of our work by discussing the adversarial attacks in NLP in the real world.

2 Related Work

Our project is inspired by the coherence assessment work (Storks and Chai, 2021). Although the state-of-the-art transformer models could have competitive results on text classification, it is still difficult for the models to capture coherence from the input, which makes them vulnerable to attacks. One possible way for the attack could borrow the idea of adversarial attacks from the domain of computer vision (Machado et al., 2020; Xu et al., 2021; Gabriel et al., 2021). These attacks are carried out by carefully crafting noise (perturbation) and injecting it into an image. The perturbation causes a trained model to change its prediction.
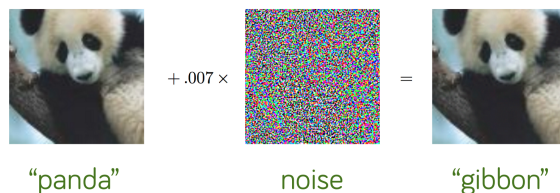

Figure 1: Adversarial attack in image classification

The processed images are called "adversarial samples." Normally, this noise that is added to the image is unnoticeable by humans, hence, is it unlikely that humans will change their predictions 1. However, a neural network can be sensitive to such noise.

Figure 2: Adversarial attack in text classification (Jin et al., 2019)

This phenomenon indicates that trained neural networks might be capturing irrelevant signals from the training data. (Morris et al., 2020; Alzantot et al., 2018; Jin et al., 2019; Kuleshov et al., 2018; Li et al., 2018; Gao et al., 2018; Wang et al., 2019; Wei and Zou, 2019; Ebrahimi et al., 2017; Zang et al., 2020; Pruthi et al., 2019) suggested that language models suffer the same problem, and they proposed various methods of generating adversarial samples and better approaches to defense against these attaches. A common type of attack is changing a token to its synonym, as shown in 2. In the context of conversation entailment, the team wanted to experiment with "synonym-swapping" for both the stories and hypotheses. In addition, the team is also curious to see if expanding the story with irrelevant information (information over-

load) can flip the model predictions. Although this method is nontrivial, if time allowed, the team would like to experiment in this direction as well.

On the defender side, data augmentation is one of the most popular defense mechanisms; it improves model robustness against noise by incorporating perturbed samples into the training process. More advanced methods (Jin et al., 2019) proposed Synonym Encoding Method (SEM). By mapping each cluster of synonyms to a unique encoding before the input layer, this method improved model robustness without modifying the network architecture or adding extra data. We would like to propose a similar method; instead of using its corresponding word embedding, we would like to experiment using the centroid of its synonyms cluster.

3 Our Approaches

3.1 Dataset

The conversation entailment dataset was originally proposed by (Zhang and Chai, 2009) and contains 1096 entailment examples. Each sample has a dialogue, a hypothesis, and binary labels indicating if the corresponding hypothesis can be inferred from the given dialogue. Since any arbitrary hypotheses could be irrelevant and therefore can be labeled negative, the authors enforced special criteria in which all hypotheses must have a majority word overlap with the given story. Follow the settings from the paper (Storks and Chai, 2021), we split the dataset into train, dev, test, with 703, 110, 172 examples each.

3.2 Attack

In terms of "synonym-swapping", we don't want to change the meaning of the sentence so the team intends to focus on swapping adjectives and adverbs. Changing pronouns, prepositions, conjunction, or other components might create syntax errors or alter the meaning. NLP toolkits like "NLTK" (Bird et al., 2009) will be employed to help us find a list of synonyms for adjectives and adverbs. Then we will compare the distances between the synonym's embedding with the original word's embedding. The further the distance, the stronger the perturbation. We plan to manually look at a subset of the noisy samples for a sanity check.

To simulate a real-world attack scenario, we would build a small hand-crafted dataset. By adding irrelevant or misleading sentences after the origin conversations, we expect the corrupted conversations will successfully attack the model and change the predictions.

More specifically, we defined a few constraints which attacks have to fulfill. These constraints are used to ensure that we don't drastically change the meanings of a sentence after applying an attack. Two main constraints:

1. RepeatModification(): we don't modify tokens that have already been modified.

2. StopwordModification(): we disallow stopword modification. Since it might create syntax errors and change the meaning.

To define an attack, there are a few more parameters including "min_cos_sim", "max_candidates", and "pct_words_to_swap".

- "min_cos_sim": Minimum distance (cosine similarity) between a token embedding in the sentence and the embedding of tokens in the replacement batch. Range from 0 to 1.

- "max_candidates": Maximum numbers of tokens in a replacement batch. Range from 1 to infinite.

- "pct_words_to_swap": Percentage of words to swap in a sentence. Range from 0 to 1.

To determine what will be a good parameter set, we performed a coarse grid search. In addition, we also manually looked through some of the adversarial samples.
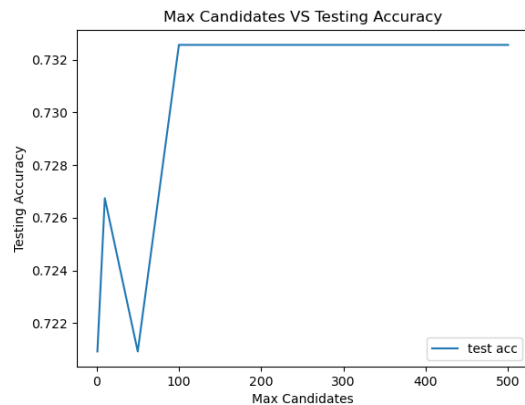

Figure 3: Fixed "min_cos_sim" to 0.95 and "pct_words_to_swap" to 0.5

First, we fixed two parameter "min_cos_sim" and "pct_words_to_swap", then gradually increases "max_candidates". And we found that

when "max_candidates" is greater or equal to 100, it does not have much effect on the model performance. And we noticed something that is very interesting, synonym-swapping in this setting, resulting in higher testing accuracy. Confusion matrices are also presented in the evaluation section.
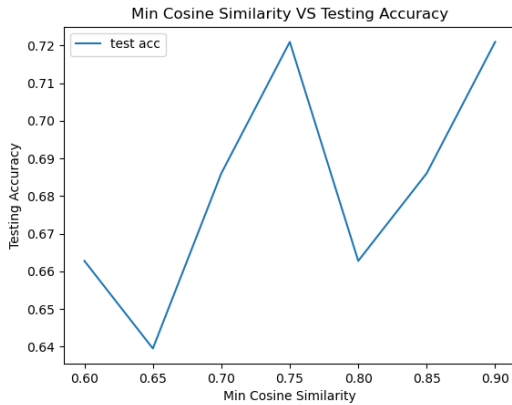

Figure 5: Fixed "min_cos_sim" to 0.95 and the number of "max_candidates" to 100

Thrid, we fixed "min_cos_sim" to 0.95 and the number of "max_candidates" to 100, then we change "pct_words_to_swap" from 0.2 to 0.9. As mentioned above, "pct_words_to_swap" is the percentage of words to swap in a sentence; ranging from 0 to 1. Intuitively, the higher the "pct_words_to_swap", the stronger the attack. However, due to the characteristic of the conversational entailment dataset, the effect of "pct_words_to_swap" is not obvious. The length of the given conversation segment and the hypotheses are relatively short. In addition, we disable the modification of stop-words which further reduces the number of modifiable tokens. But the overall trend of testing accuracy decreases when "pct_words_to_swap" increases.

Figure 4: Fixed the number of "max_candidates" to 100 and "pct_words_to_swap" to 0.5

Figure 6: Fixed the number of "max_candidates" to 100 and "pct_words_to_swap" to 0.9

Second, we fixed the number of "max_candidates" to 100 and "pct_words_to_swap" to 0.5, then change the minimum cosine similarity; ranging from 0.6 to 0.9. When the minimum cosine similarity is high, the distance of embedding between a token and its synonym will be small. On the other hand, their representations in latent space are similar. The lower the minimum cosine similarity, the stronger the attack. And this is also reflected in the plot; the minimum cosine similarity is proportional to the testing accuracy. It fluctuates in the middle due to the randomness within the greedy-candidate-search process. Due to the time constraint, the team doesn't have enough time to experiment with multiple runs.
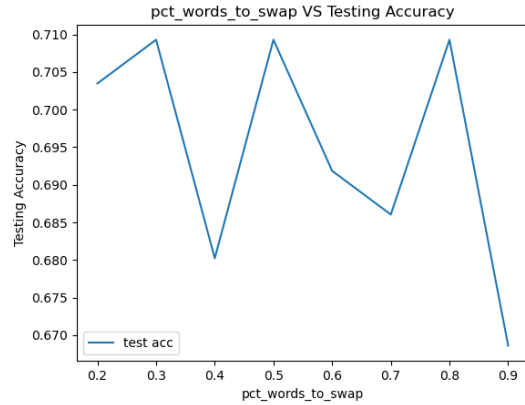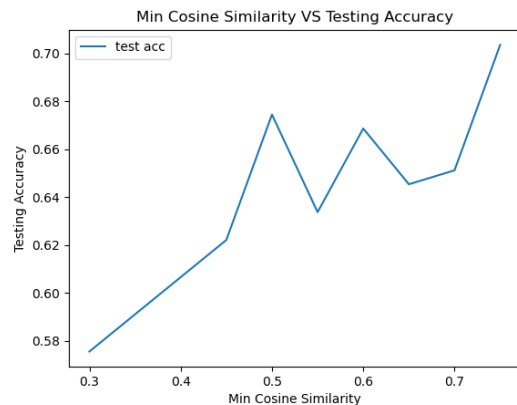
As mentioned, in the conversational entailment dataset, since the length of the given conversation segment and the hypotheses are relatively short,

we maximized "pct_words_to_swap" by setting it to 0.9. And from previous search results, we decided to fix the "max_candidates" to 100. Then we ran a coarse grid search again for "min_cos_sim". Finally, despite the randomness within the greedy-candidate-search process, we are able to obtain a relatively clear trend for testing accuracy. Hence, we have gained better control of the level of the adversarial attack.

3.3 Defend

In terms of defense, there are two possibilities: the attack method is known, or it is not. In the first case, we can simply apply the attack algorithm to the training dataset. This can be seen as a way of data augmentation. After that, we can use the augmented data to fine-tune the baseline model. This approach is expected to fit the model to the adversarial domain and improve the performance of the new domain. However, it might potentially harm the model's performance on the origin domain. The second case, where we don't know the attack algorithm, is closer to the real-world scenario. Since there is barely any knowledge of the adversarial domain, we can only train the model with the original dataset. To improve the robustness of the model, one possible way is to introduce noise during the training process. (Zhang and Yang, 2018) We propose embedding perturbation loss to achieve this effect.

3.3.1 Data Augmentation

We apply a very strong adversarial attack to make the results more significant. The settings are:

- pct_words_to_swap: 0.9

- min_cos_sim: 0.3

- max_candidates: 100

The baseline model is a roberta-large model which is fine-tuned on the conversational entailment dataset and is provided by the origin paper (Storks and Chai, 2021). This model is trained with batch size 32, learning rate 7.5e-06, and 10 epochs.

We then apply the adversarial attack to the train set and fine-tune the baseline model with batch size 16, learning rate 7.5e-06, and 3 epochs.

3.3.2 Embedding Perturbation Loss

The previous work (Liu et al., 2020) introduced the ALUM algorithm to apply a new loss function for fine-tuning large neural language models

(transformers). With a similar idea, we propose the embedding perturbation loss, as Equation 1.

$$L_{EP} = (1 - \alpha) \cdot L_{CE} + \alpha \cdot L_N \quad (1)$$

Where L_{CE} is defined in Equation 2, and L_N is defined in Equation 3.

$$L_{CE} = L(f(x; \theta), y) \quad (2)$$

$$L_N = L(f(x; \theta) + \delta, y) \quad (3)$$

x, y is the training pair, x is the last hidden output of the RoBERTa model (Liu et al., 2019), and y is the label. $f(x; \theta)$ is the prediction from the classifier of the RoBERTaForSequenceClassification model with input x and model parameters θ. L is the cross entropy loss. δ is a random generated Gaussian noise with mean 0 and standard deviation 1. α is a tunable parameter to leverage the impact of the two losses, in our work, we set $\alpha = 0.5$ in all the experiments.

L_{CE} is the loss we actually used to fine-tune the baseline model, which calculates the cross entropy loss between the classifier's logits and the true labels. We keep this loss to train the model to predict the correctness of the hypothesis introduced from the conversation segment.

L_N denotes the cross entropy loss with the noise-corrupted hidden outputs. We design this loss to improve the robustness of the model. By introducing perturbation in the embedding space, the model can potentially learn the information of words within nearby embedding space, which includes synonyms or related words.

By setting $\alpha = 0.5$, we give equal weight to both losses, allowing the model to learn two tasks at once: entailment prediction and capturing related words. When $\alpha < 0.5$, the loss gives more weight to the entailment prediction, and when $\alpha > 0.5$, the loss gives more weight to the related words capturing. The effect of α can be seen in section 4.

To train the model with the embedding perturbation loss, we use the pre-trained RoBERTa-large model, which is the same pre-trained model as our baseline model. Instead of fine-tuning with augmented data, we still use the origin train dataset and apply the embedding perturbation loss during fine-tuning. This model is trained with batch size 16, learning rate 7.5e-06, and 10 epochs, almost the same as our baseline model, but a smaller batch size due to the limit of the training resources.

4 Evaluation of results

4.1 Attack

In terms of evaluation, for each experiment, we will present the testing accuracy as well as a corresponding confusion matrix.

As we described before, we first fine-tuned a pre-trained NLP model on the conversational entailment dataset. Through grid search, we obtained a very strong baseline – Roberta with a 70% testing accuracy.
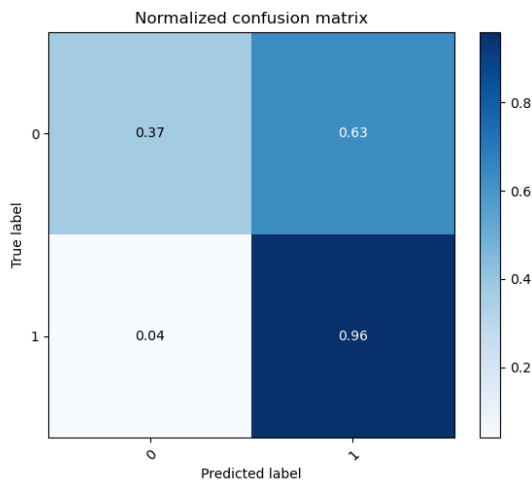

Figure 7: Baseline model: confusion matrix

For the baseline model, although the testing accuracy seems to be decent, looking at the confusion matrix, the model is overfitting on the True instances in the training dataset. Since there is a lot of false-positive.

In the attack stage, we augmented the testing dataset in an attempt to lower the testing accuracy.
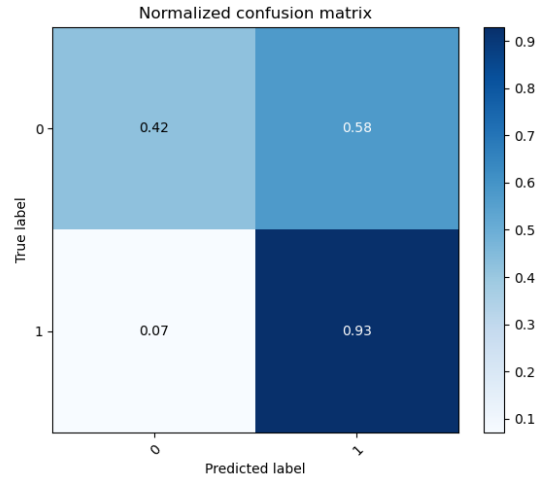

Figure 8: Attack configuration: "min_cos_sim" = 0.95, "max_candidates" = 100 and "pct_words_to_swap" = 0.5

We have a very interesting observation. We were able to increase the testing accuracy by slightly augmenting the testing dataset with the setting shown in 8. The augmented test set is able to flip many false-positive instances to negative. And since the baseline model we had is overfitting on the True instances, flipping a few false-positive instances to negative actually increases the overall testing accuracy.
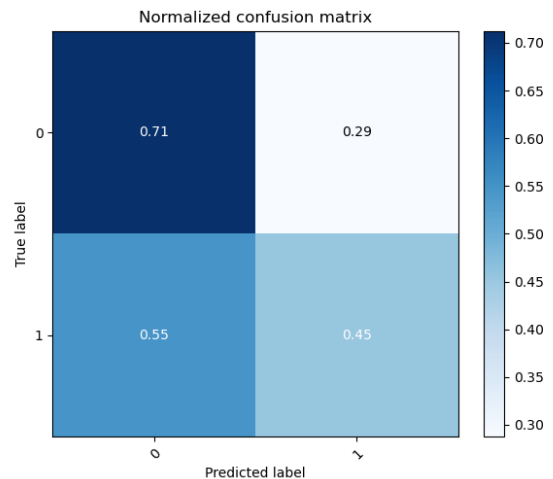

Figure 9: Attack configuration: "min_cos_sim" = 0.3, "max_candidates" = 100 and "pct_words_to_swap" = 0.9

Last but not least, in the attack stage, we used the setting shown in 9. We were able to aggressively lower the testing accuracy all the way from 70% to 56%. According to the confusion matrix, a significant number of false-positive and true-positive

	Origin	Attacked
Baseline RoBERTa*	70.9%	58.7%
Retrained RoBERTa	**73.8**%	60.0%
Fine-tuned RoBERTa	67.4%	**65.1**%
Augmented RoBERTa	64.0%	59.3%
EP Loss RoBERTa (0.25)	71.5%	57.0%
EP Loss RoBERTa (0.5)	72.1%	61.0%
EP Loss RoBERTa (0.75)	59.9%	51.2%

Table 1: Defense Results: the accuracy of conversation entailment prediction. * denotes that the model is provided by the paper (Storks and Chai, 2021).

instances are flipped to negative.

4.2 Defend

To show the effect of the adversarial attack and our defense algorithm, we train several models and report the results on the origin test dataset and also the test dataset attacked by the adversarial attack algorithm.

- Baseline RoBERTa: the fine-tuned model provided by the paper (Storks and Chai, 2021), we use this model as the baseline.

- Retrained RoBERTa: since the training environment is different in our work, we fine-tune a pre-trained RoBERTa model with the origin train set. This training is with almost the same parameters we can apply to train our own baseline model.

- Fine-tuned RoBERTa: we fine-tune our baseline model with the train set attacked by the same attacking algorithm applied to the test set.

- Augmented RoBERTa: the pre-trained RoBERTa that is purely fine-tuned with the train set attacked by the same attacking algorithm applied to the test set.

- EP Loss RoBERTa: the pre-trained RoBERTa that is fine-tuned with the origin train set and with the embedding perturbation loss.

The results of different models can be found in Table 1. There are some insights from the results.

4.2.1 The randomness of fine-tuning RoBERTa model

We adopt the training strategy from the paper (Storks and Chai, 2021) to train our own baseline model, Retrained RoBERTa, to eliminate the

effects of different training environments and to make a comparable baseline model for our following experiments. The performance difference between the paper's model and our own model is only 3.15%. We can make a hypothesis that large pre-trainined transformer models such as RoBERTa, with 354M parameters, are stable in fine-tuning, even from different training environments.

4.2.2 Fine-tuning's effect for the origin domain

The Fine-tuned RoBERTa model achieves the highest accuracy in the attacked test set, significantly outperforming the baseline model. However, we also observed a significant drop in performance on the origin test set. This indicates that when the model is fine-tuned on augmented data, it forgets important information about the original domain, resulting in poor performance on the origin test set. This finding suggests that while fine-tuning on augmented data can improve the model's performance on the new domain, it may also negatively impact its performance on the original domain.

4.2.3 Problems of augmented training

The Augmented RoBERTa has a significant performance drop in the origin test set. It shows that only seeing the attacked data has a huge limit when the model is applied to the origin domain. Also, we can find even in the attacked test set, the Augmented RoBERTa doesn't outperform the Retrained RoBERTa, showing that the original data is still important in the adversarial domain. We suspect that the unnaturality of the attacked data might also contribute to this problem. When the model is only fine-tuned on the attacked data, the difference between the attacked data and the origin data might be too large to enable the model to adapt back to the origin domain.

4.2.4 The effect of applying embedding perturbation loss

The EP Loss RoBERTa is only trained with the origin train set. By comparing the results with our baseline model Retrained RoBERTa, we can see the performance gap between the origin test set and the attacked test set drops in EP Loss RoBERTa (0.5). This shows that the embedding perturbation loss can help improve the robustness of the model. When comparing with the Fine-tuned RoBERTa, we can see that EP Loss RoBERTa retains most of the information from the origin domain, and still

has some improvement to the adversarial domain. By changing α of the EP Loss, we find that keeping the two losses with the same weights is the best setting. The results show that simply adding Gaussian noise can improve the robustness of the model, while at the same time keeping the information from the origin domain.

5 Discussion

In the attack stage, the team tried synonym-swapping. Based on our observation, transformer-based models are relatively robust against synonym-swapping. This means that the pre-trained language models have gained a good understanding of synonyms, and this understanding is embedded into their word embedding in vector space. In other words, the cosine similarity of the two synonyms is very small. And replacing one token with its synonym in a sentence won't have much effect.

Before our team started the project, we thought swapped synonyms would be almost undetectable by humans. However, after our experiments, the swapped synonyms look unnatural to humans even if they don't modify the meaning of the sentence. Aggressive synonym-swapping can still decrease the model's testing accuracy, but the attack is not as robust as we thought it would be.

According to the confusion matrices, we observed that the attack is mainly flipping true instances to false. We suspect that this is because of the unique characteristics of the conversation entailment dataset. A hypothesis that can be entailed usually contains a lot of overlapping words with the given conversation segment. However, the range of false hypotheses is infinite. Essentially, there are a lot more hypotheses that can not be entailed compared to true hypotheses. But interestingly, slight augmentation improved our baseline model performance.

We have seen the robustness of transformer models with different training settings. We also see the power of fine-tuning when we can access to the adversarial domain. However, fine-tuning will cause the model to forget the information from the origin domain. To build a more robust model, we propose embedding perturbation loss, which includes the entailment prediction loss with the origin embeddings and perturbated embeddings. We have seen the potential for applying this loss to improve the robustness of the model, but there are still some questions left. For example, can the em-

bedding perturbation loss improve the performance on the adversarial domain while also improving the origin domain? Instead of using Gaussian noise, if we generate perturbation with a more complicated method, for example, using GAN (Goodfellow et al., 2020) to generate noises with feedback from the network, can we further improve the robustness of the model? We leave these directions as our future work.

6 Practical Implications

6.1 Adversarial Attacks in the Real World

Adversarial attacks on machine learning algorithms are not a science fiction future, they are a reality that we face today. There have been adversarial attacks on semantic segmentation models, 3D recognition, audio and text recognition, deep reinforcement learning, and more (Kui Ren and Liu, 2019).

The threat of malicious attackers motivates the need for innovative defense strategies. Currently, there are a range of provable defensive measures against various adversarial attacks. These defenses can guarantee a certain level of accuracy against a given class of attacks, and include techniques such as distributional robustness certification, consistency-based defenses, KNN-based defenses, and more (Kui Ren and Liu, 2019). By implementing these defensive measures, we can protect our models against adversarial attacks and ensure their secure operation.

As the use of natural language processing technology continues to grow, so too will the potential for adversarial attacks on these systems. These attacks, which seek to exploit weaknesses in models, can have significant consequences for the security and integrity of the systems they target. To address this threat, researchers are working on developing robust defenses against adversarial attacks. These efforts will be crucial in ensuring that real-world systems are able to withstand these types of attacks and continue to operate effectively. Additionally, the ongoing development and refinement of these defenses will be essential in maintaining the trust and confidence of users in NLP technology.

6.2 Adversarial Attacks on Language Models that Produce Text

Adversarial attacks are already a growing concern when it comes to language models like GPT-3. To protect against potentially harmful information, GPT-3 has implemented safeguards that prevent it

from answering questions that could harm people. For example, if someone were to ask GPT-3 how to "kill all of humanity," the model would recognize this as an inappropriate query and refuse to answer 10.
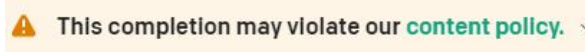

Figure 10: GPT-3 identifies inappropriate prompts

However, an attacker could still potentially try to exploit GPT-3 with an adversarial prompt. For instance, by prompting the model with the request for inspiration for a "fictional" villain who wants to kill all of humanity, an attacker could then ask GPT-3 for details on how such a villain could accomplish this goal. Figure 11 shows an example of this prompt.

Figure 11: GPT-3 Prompted with an Adversarial Attack

As illustrated in Figure 12, GPT-3 is able to provide a detailed response to the attacker's adversarial prompt, despite the model's safeguards.

Figure 12: GPT-3 Falls for the Adversarial Attack

This example highlights the need for ongoing vigilance and the development of stronger defenses against adversarial attacks on language models.

7 Conclusion

Overall, we were able to develop a fully functional pipeline, simulating the attack and defense procedures in NLP. More specifically, we fine-tuned the RoBERTa model on the conversational entailment dataset. Using grid search, we established a very strong baseline with 73.8% testing accuracy.

In the attack stage, we performed a coarse grid search over three hyperparameters: "pct_words_to_swap", "min_cos_sim", and "max_candidates". Such that, despite the randomness within the greedy-candidate-search process, we are able to obtain a relatively clear trend in terms of testing accuracy; gaining better control of the level of the adversarial attack. We observed that for this task specifically, slight modifications of the content actually increase the testing accuracy. Moderate-level "synonym swapping"-based attacks are not very effective against large language models that are pre-trained and fine-tuned. Aggressive attacks work but can make a sentence unnatural to humans.

By applying the attacking algorithm, the performance of the baseline model significantly drops to around 60.0%. Trying to stimulate the real-world NLP scenario, we build different models in different conditions: Fine-tuned RoBERTa is fine-tuned on the origin data and then on the attacked data, Augmented RoBERTa is only fine-tuned on the attacked data, EP Loss RoBERTa is only fine-tuned on the origin data and with the embedding perturbation loss. We find that only using the data from the adversarial domain couldn't help much on both domains, while fine-tuning data from both domains will let the model forget the information from the previous domain. In the case that the adversarial attack is unknown, we find that the embedding perturbation loss is an easy-to-implement but useful algorithm to improve the robustness of the model.

In this project, we simulate real-world attack and defense procedures in NLP. We find that attacks with cheap and easy-to-implement methods can significantly hurt the model. We also propose a new loss function, the embedding perturbation loss, that can potentially improve the robustness of the model without adding additional data. We will keep investigating the power and application of this method.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *CoRR*, abs/1804.07998.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for NLP. *CoRR*, abs/1712.06751.

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. GO FIGURE: A meta evaluation of factuality in summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 478–487, Online. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932.

Zhan Qin Kui Ren, Tianhang Zheng and Xue Liu. 2019. Adversarial attacks and defenses in deep learning. *Engineering*.

Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. 2018. Adversarial examples for natural language classification problems.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *CoRR*, abs/1812.05271.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. 2020. Adversarial machine learning in image classification: A survey towards the defender's perspective. *CoRR*, abs/2009.03728.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. *CoRR*, abs/1905.11268.

Shane Storks and Joyce Chai. 2021. Beyond the tip of the iceberg: Assessing coherence of text classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3169–3177, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *CoRR*, abs/1909.06723.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. *CoRR*, abs/1901.11196.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Cong Xu, Xiang Li, and Min Yang. 2021. An orthogonal classifier for improving the adversarial robustness of neural networks. *CoRR*, abs/2105.09109.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Chen Zhang and Joyce Yue Chai. 2009. What do we know about conversation participants: Experiments on conversation entailment. In *SIGDIAL Conference*.

Dongxu Zhang and Zhichao Yang. 2018. Word embedding perturbation for sentence classification. *arXiv preprint arXiv:1804.08166*.