Exploration of Adversarial Attacks on Legal-related Text Datasets

Ronaldo Canizales
Department of Computer Science
Colorado State University
Fort Collins, CO 80521
rcanizal@colostate.edu

Sarthak Bharadwaj
Department of Computer Science
Colorado State University
Fort Collins, CO 80521
sarthak7@colostate.edu

Abstract

Natural Language Processing (NLP) faces a significant challenge in the form of adversarial examples, wherein seemingly insignificant changes to input data can lead to models producing inaccurate predictions. We look forward to attacking the European Court of Human Rights Dataset using the TextAttack Python framework, which offers a modular approach for systematically producing adversarial examples on NLP models through state-of-the-art black box attacks. We perturbed the text in the dataset's facts column using various attack models within the TextAttack framework, showing that adversarial manipulation can affect assessments of an individual's guilt or innocence. We successfully change the models' classifications by utilizing six distinct attack techniques, demonstrating how susceptible NLP models are to malicious adversarial attacks.

1 Introduction

Machine learning (ML) models are pervasive in today's technology landscape and are being used in a wide variety of applications across all fields, from economics to health sciences, engineering, and humanities, like law. Like any other technology created by humankind, the possibility that it fails exists. This is a problem when such models are used in safety-critical scenarios like real-time energy and natural disaster monitoring. Not only random unlucky corner cases can fool ML models, but also carefully hand-crafted adversarial examples can be created for many input modalities like images, sound, and text as seen in Figure 1; which is currently a very active area of research.

In this work, we experimented with adversarial attacks on the European Court of Human Rights (ECTHR) violations legal dataset [1] to explore how small perturbations in the input can cause misbehavior in models, which can affect people's lives on a large scale. These attacks revealed weaknesses in models, which in the future can be used to strengthen models' resilience and as a guideline for more robust models. We investigate how different adversarial strategies work and how NLP models react to these perturbations using the TextAttack framework [2]. Through testing these approaches, we gained a deeper understanding of the broader effects of adversarial attacks on the legal data set.

2 Methodology

Our work can be divided into three steps. First, we pre-processed the ECTHR dataset. Second, we trained two models on binary and multi-class classification of the data; we look forward to obtaining the highest validation accuracy possible. Finally, we executed several black-box attacks on test data and gathered statistics about their behavior.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

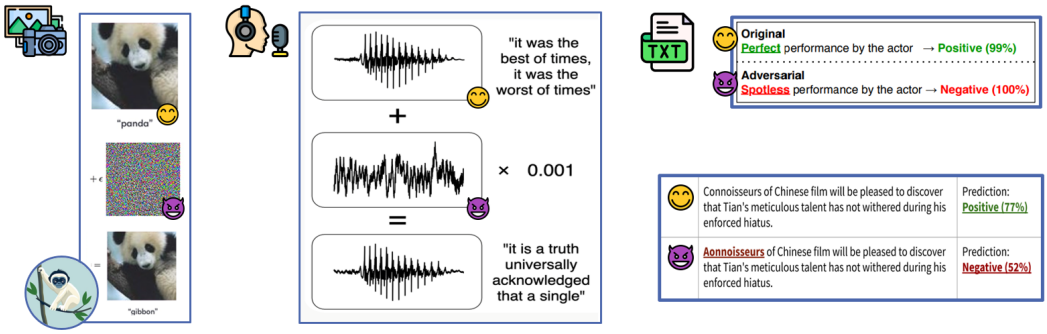

Figure 1: Adversarial attacks in various modalities aimed to fool ML models by misclassifying inputs. Left: adversarial noise added to a panda's image. Center: adversarial noise added to an audio file. Right: small changes, such as one word or one character, performed in two movie reviews.

2.1 Dataset pre-processing

We selected the European Court of Human Rights (ECTHR) dataset [1] as our target dataset because it contains rich semantic information related to violations of the European Convention on Human Rights [3]. The dataset was initially designed to experiment with systematic judgment prediction and rationale extraction. For each case, understanding a case as a unique dispute between two entities (such as individuals, companies, or countries) is resolved by a court using the concrete facts of the case (in a nutshell, a decision made by a judge or judges), the dataset contains: (a) a list of "facts" in the form of paragraphs, (b) who the "applicants" and "defendants" are, (c) a list of the allegedly violated articles are, (d) a list of the actual violated articles if any, (e) and other fields related to which specific facts led to the decision made by the judges.

For this work purposes, only the fields "facts" and "violated articles" are relevant. The former is concatenated into a single big string per case; the latter is pruned to only those cases where a single article was violated, as taking into account multiple simultaneous violated articles was difficult and we considered out of the scope of this work. Lastly, we obtained two simplified datasets. One designed for binary classification of "there was a violation" or "there was no violation" (empty list). The other dataset contains eight classes: "no violation" and seven specific articles and protocols from the European Convention on Human Rights [3], specifically:

	No article violation.
A2	Right to life.
A3	Prohibition of torture.
A5	Right to liberty and security.
A8	Right to respect for private and family life.
A10	Freedom of expression.
A13	Right to an effective remedy.
P1	Protection of property.

We selected this specific list of violations because they are the most frequent in the dataset. Other article violations are rare (like A12: Right to marry) or appear only in conjunction with others and not in isolation.

2.2 Training of base models

We leveraged a Google pretrained model called "gnews-swivel" [4], which maps from text to 20-dimensional embedding vectors. This model was trained on the English Google News 130GB corpus; its vocabulary contains 20,000 tokens and 1 out of vocabulary bucket for unknown tokens. It was created using Swivel matrix factorization method. This model receives a 1-D tensor of strings as input and preprocesses it by splitting on spaces. It is publicly available for its use in TensorFlow for arbitrary length text embeddings and also as a Keras layer for incremental model building.

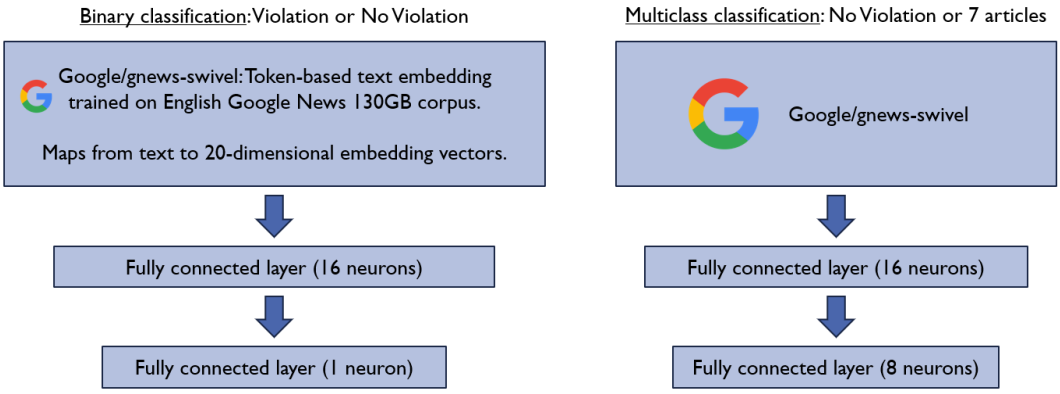

Figure 2: Models' architecture. Both models' first layer leverage pre-trained google-gnews-swivel model [4]. On top of that, two additional fully connected layers are trained with (left) binary entropy loss and (right) categorical entropy loss, respectively.

We implemented our models with two additional fully connected layers, the first one contains 16 neurons and the second one varies depending on binary (1 neuron) or multi-class (8 neurons) classification, as detailed in Figure 2. We fine-tuned this model using 9,000 training and 1,000 validation cases from the EDTHR dataset up to obtaining the following accuracies:

- **Binary classification model:** 94.9% training and 85.2% validation accuracy.
- **Multi-class classification model:** 79% training and 60% validation accuracy.

2.3 Getting ready to attack: Attacks setup and strategy

As summarized on Figure 3, we aim to use adversarial attacks in both our classifier models to change the articles under which an individual is judged by manipulating the input data. The binary model classifies a person as guilty if they are booked under any article of the European Convention of Human Rights; if no article is mentioned, the model classifies them as not guilty. We also plan to attack our multi-class classifier, transforming a person's status either by removing their violations, adding a violation, or changing their guilt from violation of one article to another.

Six different attacks were selected to be used to manipulate the details of the case (model's input):

- **PWWS [5]**: Words are prioritized for a synonym-swap transformation based on a combination of their saliency score and maximum word-swap effectiveness.
- **Pruthi [6]**: This attack focuses on a small number of character-level changes that simulate common typos. It combines swapping neighboring characters, deleting characters, inserting characters, and swapping characters for adjacent keys on a QWERTY keyboard.
- **DeepWordBug [7]**: Simple character-level transformations are applied to the highest-ranked tokens to minimize the edit distance of the perturbation while changing the original classification.
- **TextBugger [8]**: Generates adversarial text with computational complexity sub-linear to the text length.
- **InputReduction [9]**: Deletes words until the model misclassifies.
- **CLARE [10]**: Uses greedy search with replace, merge, and insertion transformations that leverage a pretrained language model.

When attacking a model, there are several possible outcomes; detailed as follows:

- **Attack succeeded:** modified input is missclassified by the model, untargeted attack.
- **Attack failed:** modified input is still classified correctly by the model.
- **Attack skipped:** original input is missclassified by the model; no action taken.
- **Attack timed-out:** after one hour of executing an attack, no output is obtained.

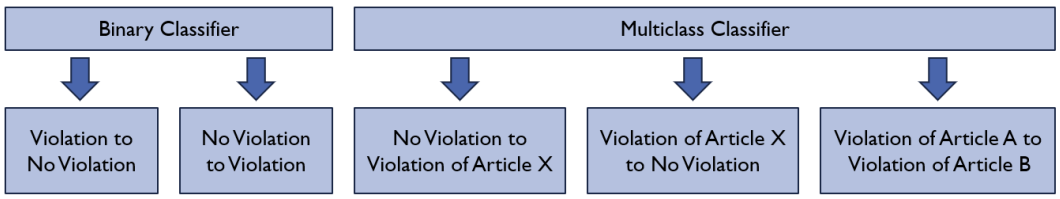

Figure 3: Methodology, detail of desired attacks. Note: even though our code searched for untargeted missclassifications, we looked forward obtaining at least one example of all combinations of targeted missclassifications. For mode details, visit Appendices A and B.

3 Experimental Results

We compare the performance of five attack techniques on a binary classifier in the table: PWWS [5], Pruthi2019 [6], DeepWordBug [7], Text Bugger [8], and InputReduction [9]. With the model's initial accuracy and accuracy under attack, the table displays the number of successful, unsuccessful, and skipped attacks. We included metrics such as average perturbed words, average number of queries, and attack success rate. In contrast to Text Bugger,PWWS Ren2019 has the lower perturbation (5.75%) with a bit higher queries, whereas PWWS and InputReduction both achieved 100% attack success rates. According to results in the table we come to conclusions that PWWS Ren2019 performed the best achieving 100% accuracy with less perturbations in words.

Table 1: Results on Binary Classifier

Attack	PWWSRen'19	Pruthi'19	DeepWordBug Gao'18	TextBugger Li'18	InputReduct Feng'18
Successful attacks	8	0	1	7	8
Failed attacks	0	8	7	1	0
Skipped attacks	2	2	2	2	2
Original accuracy	80%	80%	80%	80%	80%
Accuracy under attack	0%	80%	70%	10%	0%
Attack success rate	100%	0%	12.5%	87.5%	100%
Avg. perturbed words	5.75%	-	1.25%	89.31%	35.94%
Avg. number of queries	5,882	22,025	774	2,024	1,906

In the table, we evaluate the effectiveness of four attack methods on a multiclass classifier: InputReduction [9], Text Bugger [8], Pruthi2019 [6], and PWWS [5].The table shows the number of successful, failed, and skipped attacks along with the model's initial accuracy and the accuracy that was attacked. Compared to Text Bugger, InputReduction required a few more queries but produced a lower perturbation (37.03%). A 100 percent attack success rate was attained by both Text Bugger and InputReduction, totally destroying the model's performance to 0 percent accuracy. Even though PWWS Ren2019 needed the most queries, it showed a balance between effectiveness (83% attack success rate) and minimal perturbation (1.93%).

Table 2: Results on Multi-class Classifier

Attack	PWWSRen'19	Pruthi'19	TextBugger Li'18	InputReduction Feng'18
Successful attacks	5	0	4	4
Failed attacks	1	4	0	0
Skipped attacks	4	6	6	6
Original accuracy	60%	-	40%	40%
Accuracy under attack	10%	-	0%	0%
Attack success rate	83%	0%	100%	100%
Avg. perturbed words	1.93%	-	69.18%	37.03%
Avg. number of queries	19,732	22,466	1,473	1,939

3.1 Experimental setup

All training and attacking occurred in a single computer which specs are: System76 Serval WS, 96.0GB RAM memory, Processor Intel Core i9 (32 cores), Ubuntu 22.04, and NVIDIA RTX 4070.

4 Discussion and Future Work

The trade-off between the average number of perturbed words and attack success rate draws attention to a crucial adversarial text attack challenge: striking a balance between subtlety and effectiveness. Attacks like PWWS and InputReduction were able to fool the model with high success rates—often 100 percent by altering the text only slightly, maintaining the meaning of the text. Other approaches, like TextBugger, on the other hand, necessitated more extensive perturbations, which, although successful, ran the risk of changing the semantics of the text and becoming obvious. Real-world applicability depends on this trade-off; in order to preserve human interpretability and credibility, successful attacks must balance increasing attack success rates with reducing the amount of text alteration.

The study can be expanded for future research to incorporate a wider variety of datasets in order to assess the classifier's resilience across various domains. Targeted attacks might also be investigated to see how they affect particular classifier classes. Testing with more recent models may reveal information about how vulnerable contemporary architectures are. To improve the classifier's resilience and lessen adversarial vulnerabilities, it would also be beneficial to put defense mechanisms against these attacks into place.

5 Conclusion

Our primary contribution is the investigation of adversarial attacks on the dataset of the European Court of Human Rights in order to identify important weaknesses in models used for natural language processing. For both binary and multi-class classifications, we showed that even small changes to textual inputs could significantly change model predictions. We demonstrated the vulnerability of NLP models to adversarial manipulations by experimenting with six attack techniques using the TextAttack framework [2], especially in high-stakes domains such as legal decision-making.

This study emphasizes how critical it is to address adversarial robustness in NLP systems. Our results highlight the fact that these systems are still susceptible to manipulation in the absence of appropriate safeguards, which could have dire repercussions in practical applications. In addition to offering a methodology that can be applied to other datasets and domains for additional research, our work establishes the foundation for creating stronger, more resilient models by recognizing and evaluating these flaws.

References

[1] Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases, 2021.

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

[3] Council of Europe. Convention for the protection of human rights and fundamental freedoms, 1950.

[4] Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. Swivel: Improving embeddings by noticing what's missing. *CoRR*, abs/1602.02215, 2016.

[5] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency, 2019.

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

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

[8] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*, NDSS 2019. Internet Society, 2019.

[9] Shi Feng, Eric Wallace, Mohit Iyyer, Pedro Rodriguez, Alvin Grissom II, and Jordan L. Boyd-Graber. Right answer for the wrong reason: Discovery and mitigation. *CoRR*, 2018.

[10] Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. Contextualized perturbation for textual adversarial attack. *CoRR*, abs/2009.07502, 2020.

Appendix A: Results on Binary Classifier

Adversarial attacks consist on purposefully introduce "perturbations," or little modifications, into the text data such that the model is confused by the altered text. Though they aren't noticeable enough to modify the text's meaning significantly for a human reader.

Figure 4: In a case originally dictated as a violation by the European Council on Humans Rights, which our model correctly classifies as violation with 95% confidence, an adversarial attack is able to change it to "no violation" after some synonyms and similar-looking changes to the input text.

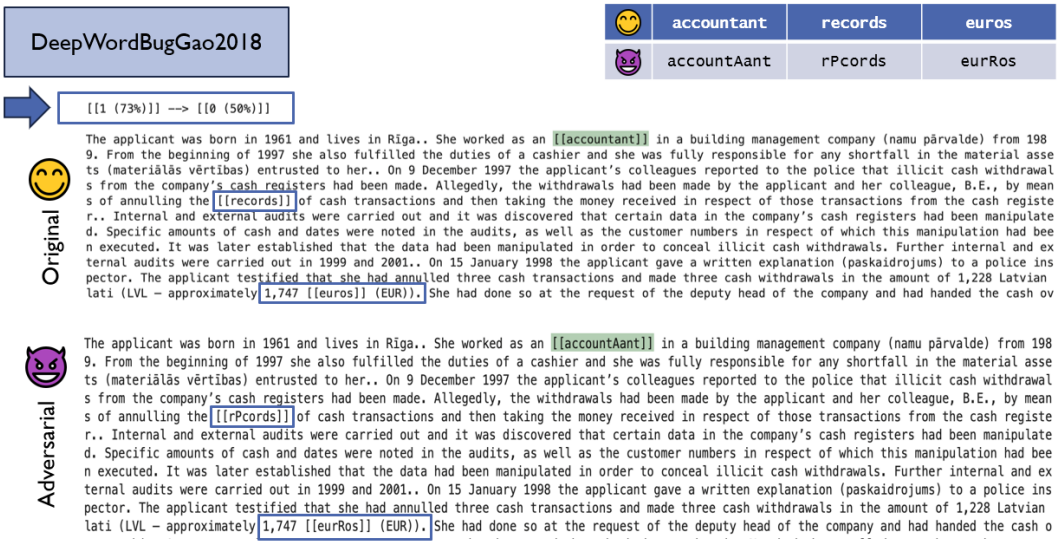

Figure 5: Similarly to the previous example, an individual originally found guilty is missclasified as innocent by performing minor character-level changes to the description of the case details.

Appendix B: Results on Multi-class Classifier

Examples of missclassifications achieved on the multi-class model performed by various attacks:

- An individual originally found guilty of violating the "protection of property" protocol, is missclassified as being guilty of the "right to liberty and security" article of the European Convention on Human Rights. This is achieved using the Text Bugger attack by substituting some words by their synonyms and adding minor character-level changes.
- An individual originally found guilty of violating the same protocol, "protection of property," is missclassified as innocent (no violation) by performing the same kind of changes in their case description. Such as owned to possesion, 1954 to 1594, land to earth, land to lad, plot to conspiracy, and co-owners to coowners.

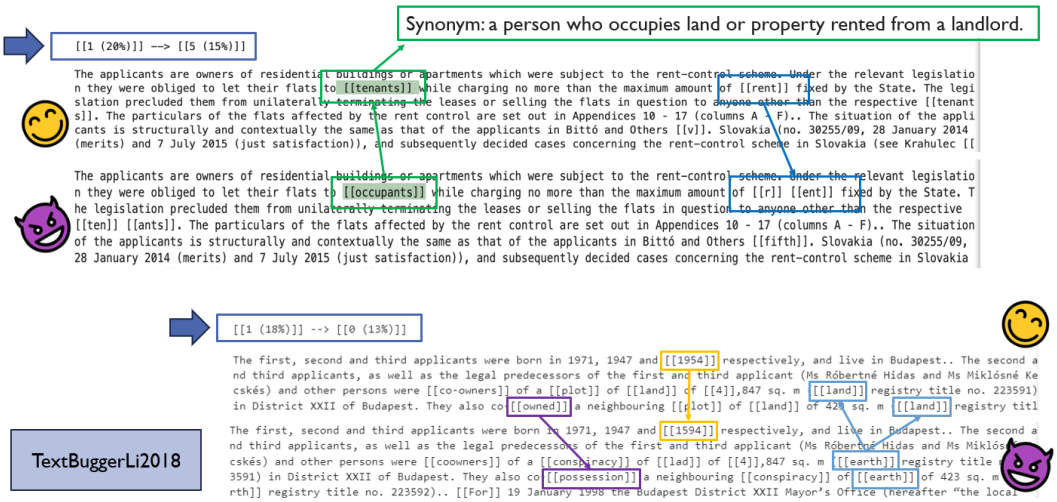

Figure 6: Examples of the Text Bugger attack successfully applied to two ECTHR cases.

- An individual originally found innocent (no violation occurred), is missclassified as being guilty of violating the "right to liberty and security" article of the European Convention on Human Rights. Similarly to the previous example, some apparently minor changes in the text causes the model to fail. Some synonyms are company/party, and necessarily/inevitably.

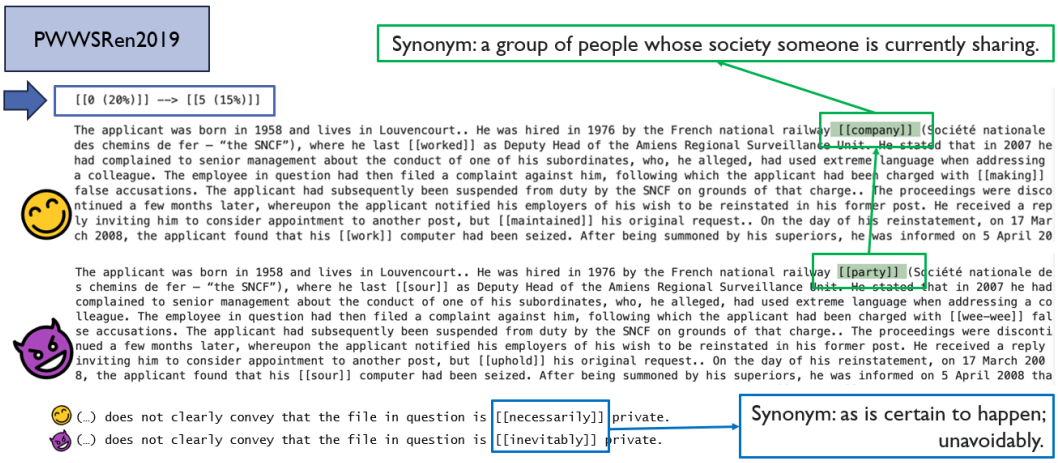

Figure 7: An example of the PWWWS attack successfully applied to an ECTHR case.