AdversNLP: A Practical Guide to Assessing NLP Robustness Against Text Adversarial Attacks

Othmane Belmoukadam [1] **Jiri De Jonghe** [1] **Naim Sassine** [1] **Ben Hover** [1] **Amir Krifa** [1] **Joëlle Van Damme** [1] **Maher Mkadem** [1] **Patrice Latinne** [1]

Abstract

The emergence of powerful language models in Natural Language Processing (NLP) has sparked a wave of excitement for their potential to revolutionize decision-making. However, this excitement should be tempered by their vulnerability to adversarial attacks, which are carefully perturbed inputs able to fool the model into inaccurate decisions. In this paper, we present *AdversNLP*, a practical framework to assess the robustness of NLP applications against text-based adversaries. Our framework combines and extends upon the technical capabilities of established NLP adversarial attacking tools (i.e. TextAttack) and tailors an audit guide to navigate the landscape of threats to NLP applications. *AdversNLP* illustrates best practices and vulnerabilities through customized attacking recipes, and presenting evaluation metrics in the form of Key Performance Indicators (KPIs). Our study demonstrates the severity of the threat posed by adversarial attacks and the need for more initiatives bridging the gap between research contributions and industrial applications.

1. Introduction

NLP and Large Language Models (LLMs) have gained increasing attention in recent years due to their breakthroughs in numerous applications. Today, generative AI and its pioneers (e.g., GPT-3/4, T5...) started the age of AI (stated Bill Gates [1]), transforming the way we interact with machines and each other. The latter models were trained on massive amounts of text data, allowing them to learn the complexities of language and make accurate predictions about the meaning of text (Brown et al., 2020; Raffel et al., 2019).

In financial services, NLP and language models have seen widespread adoption for tasks such as risk analysis, fraud detection and automation of critical back-office operations. Recently, Bloomberg released BloombergGPT, a new large-scale generative Artificial Intelligence (AI) model specifically trained on a wide range of financial data to support a diverse set of NLP tasks within the financial industry (Wu et al., 2023). However, NLP applications are highly vulnerable to adversarial attacks, i.e. perturbed input samples forcing the model to make false decisions, compromising their accuracy and trustworthiness. As such, the issue of adversarial robustness has become increasingly critical for ensuring the reliability and trustworthiness of NLP systems. To that aim, multiple contributions emerged either exploiting or mitigating text adversaries. For instance, leveraging accessible insights of the target systems (e.g., decision/score, loss-function...) in crafting efficient adversarial text samples (Li et al., 2020; Jin et al., 2020). On the other hand, multiple active and passive defenses (e.g., misspelling check, adversarial training) have been proposed to detect text attacks and limit their impact (Yoo et al., 2022; Wang et al., 2021b).

In terms of understanding and communicating the threat landscape, The ATLAS MITRE initiative provides a common language and reference point for organizations to better understand the different threats and techniques used[2]. While open-source frameworks like OpenAttack and TextAttack help testing attacking algorithms, there still is a pressing need for a standardized methodology that allows understanding of the threats, simulation of attacks and the ability to draw customer-friendly conclusions for each particular NLP application (Morris et al., 2020; Zeng et al., 2021). Inspired by the latter, we summarize our main contributions within *AdversNLP*:

- A horizontal view of the state-of-the-art text adversarial attack techniques, including toolkits and frameworks.
- A practical guide that assists in comprehending potential threats posed by adversarial attacks, automate the simulation and evaluation of personalized attacks and tests the effectiveness of shielding techniques.
- A performance evaluation using real-world cases from the financial industry, such as fake news detection and stock index classification.

[1]AI Lab. Correspondence to: Othmane Belmoukadam, Naim Sassine, Ben Hover, Amir Krifa, Maher Mkadem, Patrice Latinne <firstname.lastname@be.ey.com>, Jiri De Jonghe, Joëlle Van Damme <firstname.lastname1.lastname2@be.ey.com>.

2^{nd} AdvML Frontiers workshop at 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

By combining these contributions, *AdversNLP* can help organizations to gain a comprehensive understanding of threats to their NLP applications, simulate personalized attacking techniques and present the results in a user-friendly manner using KPIs. This enables proactive mitigation of vulnerabilities and enhances the overall robustness of NLP systems.

2. Related works

The field of text adversarial attacks has seen significant growth in recent years, with different contributions from crafting adversarial attacks to shielding techniques to mitigate their threat. In this section we provide a brief overview of the adversarial text attacks state-of-the-art.

2.1. Threat models

A plethora of attack methods have been proposed to manipulate the text input of NLP models and eventually fool them, each with their own objective function, search - and perturbation method. While the literature names multiple families, *Whitebox* and *Blackbox* threat models represent the main two and any attack can be narrowed down to one of them (Table.1 General summary).

2.1.1. Whitebox threat models

Representing the category of attacks where the attacker has complete knowledge of the underlying NLP model and its inner workings. In other words, the attacker has full access to the model's architecture, its parameters and training data. These can be used to craft adversarial examples that are most effective in fooling the model. For instance, (Ebrahimi et al., 2018) relies on flip operation, which swaps one token for another, based on the gradients of the one-hot input vectors, highlighting that character-level models are highly sensitive to adversarial perturbations. Meanwhile, (Ren et al., 2019) proposes assigning probability weights to each word in a sentence based on their contribution to the loss function and target the most important ones with perturbations leading to a derail of the model from its true prediction. TextBugger, an adversarial attack framework, when under *Whitebox* settings, finds important words by computing the Jacobian matrix of the target classifier, generates five possible substitutions and chooses the optimal one based on the change of the confidence value (Li et al., 2019).

2.1.2. Blackbox threat models

While *Whitebox* attacks assume access to the target model's architecture and parameters, *Blackbox* attacks are designed to be more practical and better represent real-world scenarios. *Blackbox* adversarial attacks are particularly challenging since the attacker does not have access to the inner workings of the model and can only leverage at most the decision and score. (Li et al., 2020) proposes BERT-ATTACK, identifying high importance words with Masked Language Model (MLMs) and using a BERT architecture to generate substitutions preserving the context of the sentence. Meanwhile, (Ren et al., 2019), introduced a greedy algorithm called Probability Weighted Word Saliency (PWWS), where word replacement order is determined by the classification probability and word saliency.

TextFooler, a two-fold adversarial attacking framework, generates semantically similar adversarial examples, that are also grammatically correct and fluently phrased. The latter technique computes an importance score per token, and then selects a suitable replacement word that has similar semantic meaning, fits within the surrounding context and forces the target model to make wrong predictions (Jin et al., 2020). On the extreme side of *Blackbox* attacks (a.k.a Blind-attacks), VIPER uses a Bernoulli distribution to decide which character to change and Character Embedding Spaces (CES) presenting visually similar substitution for multiple original characters that will be sampled once a given character is chosen for perturbation (Eger et al., 2020).

2.2. Text Adversarial Attacks Toolkits & Frameworks

In recent years, several toolkits have been developed to aid in the generation and evaluation of text adversarial attacks. We briefly mention two main examples:

2.2.1. OpenAttack

An open-source toolkit for implementing and evaluating adversarial attacks in Natural Language Processing (NLP) models (Zeng et al., 2021). It provides implementations in Python of a range of attack algorithms, as well as evaluation metrics such as attack success rate and semantic similarity. OpenAttack's architecture is based on a modular design, which allows users to customize attacking techniques and evaluation metrics. OpenAttack is flexible and extendable for various deep learning models (i.e., Transfomers [3]).

2.2.2. TextAttack

An open-source Python-based framework for generating adversarial examples in NLP. It provides a collection of pre-defined attacks and metrics to evaluate the robustness of NLP models against adversarial examples. The framework supports a wide range of NLP models, including transformers, RNNs/LSTMs and is designed to be extensible (Morris et al., 2020). TextAttack's architecture is based on a unified framework which allows users to easily compare and contrast different attack algorithms.

Table 1. Text adversarial attacks related works

References	Knowledge	Targeted	Granularity	Task	Dataset
(Belinkov & Bisk, 2018)	Blackbox	No	Char	Machine Translation (MT)	TED Corpus
(Iyyer et al., 2018)	Blackbox	No	Sentence	Sentiment/Entailment	SST/ SICK
(Jin et al., 2020)	Blackbox	No	Word	Classification/Entailement	AG's/IMDB/Yelp
(Li et al., 2020)	Blackbox	No	Sentence	Classification/NLI	AG's/FAKE/IMDB
(Eger et al., 2020)	Blackbox	No	Multiple	POS/Toxic Comments	Combilex
(Alzantot et al., 2018)	Blackbox	No	Multiple	Sentiment/Entailment	IMDB/SNLI
(Gao et al., 2018)	Blackbox	No	Char	Sentiment/Classification	AG's/AmazonReviews
(Ebrahimi et al., 2018)	Whitebox	No	Mulitple	Classification	AG
(Li et al., 2019)	Whitebox	No	Word	Sentiment/ToxiComments	IMDB/RottenTomato
(Liang et al., 2018)	Whitebox	Yes	Multiple	Classification	DBpedia/MR
(Cheng et al., 2020)	Whitebox	Yes	Word	MT/Summarization	DUC/WMT'16
(Samanta & Mehta, 2017)	Whitebox	No	Word	Sentiment/GenderDetection	IMDB/ Twitter
(Sato et al., 2018)	Whitebox	No	Word	Sequence Labeling	DBpedia/FCEpublic

2.2.3. ATLAS MITRE

While the previous tools provide a useful starting point for generating and evaluating adversarial examples, frameworks such as ATLAS MITRE offer a more comprehensive and structured best practices for testing and evaluating AI models. The Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK[4]) framework is a publicly-available knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base, created by the MITRE corporation, is used as a foundation for the study, development of specific threat models and methodologies in the cybersecurity field. The ATLAS MITRE matrix specifically targets adversarial attacks on machine learning systems. It is intended to help researchers and practitioners better understand the various types of adversarial attacks,ntheir objectives and to develop effective defense mechanisms against these attacks.

The ATLAS MITRE matrix consists of two axes: tactics and techniques. The tactics axis contains the high-level goals of an attacker, such as ML model access, discovery, collection, and ML attack staging. The techniques axis contains specific methods that attackers may use to accomplish their goals, such as ML model inference API access or leveraging ML-enabled product or services (model access), identify models artifacts or documentation and profiling outputs (discovery), information repositories to mine valuable information or data from local systems (collection), and *White/Blackbox* attacks (ML attack staging).

Although TextAttack and OpenAttack are powerful toolkits for generating adversarial examples, it can be challenging for non-experts to use effectively. Additionally, the ATLAS MITRE framework is a valuable resource for assessing cybersecurity risks, but it is not specifically tailored to NLP applications. To address these limitations, we propose *AdversNLP*, an audit framework that combines and extends upon the technical capabilities of previous toolkits, while tailoring an assessment questionnaire focusing specifically on NLP applications and assessment of adversarial robustness.

3. AdversNLP

With the increasing deployment of NLP applications in various domains, including finance, healthcare and security, the damage of text adversarial attacks can be enormous. Moreover, organisations still lack tangible and less technical tools to asses, understand and mitigate the vulnerabilities of their NLP systems. To that aim, we present *AdversNLP*, a user-friendly framework with the following core contributions: (i) A practical audit guide, describing the landscape of adversarial threats an NLP system might face, (ii) a UI automating customized attacking recipes and presenting robustness KPIs.

3.1. Audit guide

The audit guide can be summarized in six steps, aiming to identify risks, generate adversarial attacks, evaluate the model and provide shielding techniques assessment. The initial three steps are performed offline to understand vulnerabilities, while the latter three steps form an iterative process of generating and evaluating adversarial inputs, applying shielding methods, and repeating for enhanced robustness. After simulating and evaluating customized attacks, a dashboard with various Key Performance Indicators (KPIs) is available, providing insights into the model's robustness performance.

3.1.1. Define the Scope

The first step towards auditing or assessing the robustness of NLP applications is to define the scope and the specific assets to be evaluated:

- What is the purpose of the NLP application and target domain?
- What are the input and output formats of the NLP application?
- What are the potential impact and consequences of successful adversarial attacks on the NLP application?
- Who are the users of the NLP application, and what are their expectations and requirements?
- What are the legal and ethical implications of the NLP application and its potential vulnerabilities?

Answering these questions helps understanding the target system purpose and the potential attack surface.

3.1.2. Identify accessible attack assets

This section puts in perspective the assets that can be leveraged by an attacker to craft efficient adversarial examples. This includes determining the various data sources, such as training, consumer or scored by the NLP application, as well as any state of the art architectures used to build the model in question (e.g., BERT, T5 ...).

- What types of access do you have to the NLP model (e.g., source code, API, hosted service)?
- What type of data can be used to test/valuate/probe the model (e.g., publicly available datasets, custom datasets)?
- What are the resources used to build the application (e.g., fine-tuning datasets, architectures)?

By identifying the accessible attack assets, it becomes easier to determine the range of possible attack vectors and to select appropriate attack methods.

3.1.3. Select attacks to simulate

This step involves choosing a set of adversarial attacks that are relevant to the application being tested, and can effectively simulate potential attack scenarios. The selection should be based on the scope of the application and the accessible assets identified in the previous steps.

- What kind of attacks to simulate: *Blackbox/Whitebox*? Targeted or Untargeted?

3.1.4. identify target samples

This step focuses on the selection of the dataset used to craft adversarial examples. If the dataset to attack is already available, it can be used directly. However, if no dataset provided, one can probe the model for a representative dataset. The goal is to obtain a diverse set of examples that can effectively evaluate the robustness of the NLP application.

3.1.5. Evaluate results

The evaluation of the generated attacks is two-fold: one part is assessing the attack success, the other part is the fluency of adversarial examples.

Model performances: Starting with the standard evaluation metrics (e.g., accuracy, precision/recall) up to success rate, processing time and number of queries.

Fluency: Measuring the fluency of generated samples, using metrics such as the word modification rate, semantic similarity and grammatical errors.

3.1.6. Shielding

Based on the analysis of the previously mentioned results, *AdversNLP* will eventually provide recommendations to improve the robustness of NLP applications (feature under developement, see Future works). Here are some examples of shielding techniques that can be suggested:

- **Adversarial Training:** Augmenting the training data with adversarial examples to force the model to learn semantics of adversarial perturbations (Jin et al., 2020; Li et al., 2019).
- **Input Sanitization:** Pre-processing the input data to remove potential adversarial triggers (e.g., special characters, mislead words...) (Pruthi et al., 2019; Zhou et al., 2019).
- **Adversary Detection:** Training a binary classifier to serve as a filter for adversaries (Yin et al., 2022).

3.2. AdversNLP architecture

AdversNLP is designed to be hosted on the Azure cloud platform and incorporates various Azure services to achieve scalability, reliability, and seamless integration [8]. The solution consists of a Streamlit-based front-end for user interaction, with a back-end leveraging libraries like TextAttack and OpenAttack for adversarial NLP capabilities (Yoo et al., 2022; Zeng et al., 2021). The entire application is containerized using Docker and deployed through Azure App Services and Azure Kubernetes Service (AKS). Access to the application is secured and managed by Azure Active Directory and Azure Application Gateway. The NLP models supported by *AdversNLP* can be stored on Azure Machine Learning, while Azure Blob Storage and Azure SQL Database handle storage and database requirements.

In Figure.1, we dive into the functional architecture of *AdversNLP*, and highlight main components of its integration flow within the Azure Cloud Platform:

- **(Front/Back)-end:** The front-end of the AdversNLP application is built using Streamlit, a popular Python library for creating interactive web applications for data

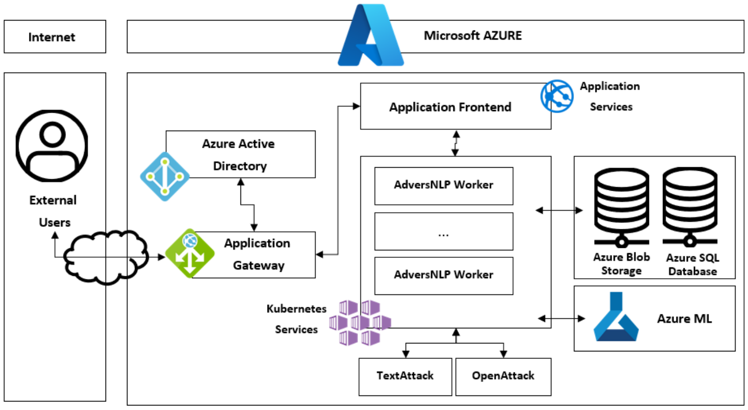


Figure 1. The AdversNLP architecture (Azure)

	TextAttack	OpenAttack	ATLAS	AdversNLP
Attack Classification Models	✓	✓		✓
Provides text pre-processing capabilities	✓	✓		✓
Provides a wide array of metrics to evaluate the attacks	✓	✓		✓
Provides adversarial tactics/techniques & recommendations			✓	✓
Customized for NLP applications	✓	✓		✓
Provides a friendly User Interface (UI)				✓
Provides user-friendly KPI's				✓
Cloud-based				✓

Table 2. AdversNLP and state-of-the-art toolkits and frameworks.

science and machine learning. The back-end is powered by the TextAttack and OpenAttack (highlighted in Section. 2).

- **Azure Deployment:** *AdversNLP* is deployed on Azure using Azure App Services and Azure Kubernetes Service (AKS). Azure App Services allows for easy deployment of web applications, while AKS provides a managed Kubernetes environment for container orchestration and scaling.

- **Access Management:** Azure Active Directory (Azure AD) is used to manage user identities and access to the *AdversNLP* application. Azure Application Gateway acts as a web traffic load-balancer and provides an additional layer of security and protection to the application.

The application interface (see Appendix), helps customize attacking recipes, including the upload of a model, and the selection of attacks to simulate (based on first 3 steps of the audit guide). The latter triggers a series of events, simulating attacks, generation of robustness KPIs and deep-dive into adversarial examples. Overall, the *AdversNLP* framework is not about reinventing the wheel but rather about taking the state-of-the-art advancements in terms of toolkits and extending them in a way that makes them accessible to a broader audience (see Table. 2).

4. Experiments

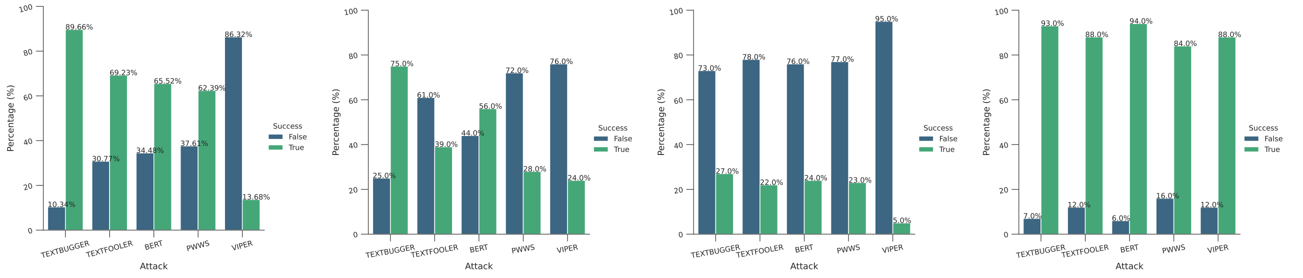
We evaluate the effectiveness of *AdversNLP* on 4 different NLP use cases. The chosen use-cases put in perspective sensitive and trending applications of NLP in financial services (not limited), ranging from document-level ESG classification and sentiment scoring to Fake news detection and multi-modal stock index classification using both historical prices and news feed from social media. To summarize, Table. 3 highlights the use cases covered, target models and their original performance evaluation.

4.1. Simulated attacks

Based on the insights collected from the first 3 steps of the audit guide, different attacking recipes can be customized. For the use cases described in Table. 3, we orchestrate a standard attacking campaign, focusing on *Blackbox* attacking methods (Li et al., 2020; Ren et al., 2019; Jin et al., 2020). The selected attacks, simulate a pseudo-real threat scenario, exploiting accessible assets such as architecture, loss function and input samples. Moreover, and to showcase the relevancy and effectiveness of Blind and *Whitebox* attacks we also consider, respectively, VIPER and TextBugger (Eger et al., 2020; Li et al., 2019). For each dataset, we evaluate the different attacking recipes on 100 randomly selected samples that the model is able to classify correctly.

Use Case	Dataset	Model	F1
FakeNews Detection	Labelled combination of fake/reel news[5]	BERT(Sigmoid)	90%
DowJones Closing	Combination of Reddit news and DJ stats[6]	Universal Sentence Encoder (Sigmoid)	82%
ESG Topic	E/S/G labelled articles (Python News API[7])	BERT (Softmax)	87%
ESG Sentiment	ESG articles with sentiment	BERT (Sigmoid)	73%

Table 3. Use cases, target models and performance evaluation



(a) FakeNews (b) ESG Topic (c) DowJones Closing (d) ESG Sentiment

Figure 2. Adversarial attacks success rate

Consider Figure. 2, showing the success rate of all five simulated attacks per use case. Regardless of the family of attack *Blackbox/Whitebox*, all four fine-tuned models can be fooled by all recipes. However, we do notice that the success rate differs for each model and that there is no silver bullet. The *Whitebox* attack (TextBugger) naturally achieves the highest success rate since it has the most knowledge. Similarly, we notice that the blind attack (VIPER) has the most difficulty tricking the model. The best of the three *Blackbox* recipes is use-case dependant.

Another metric to consider is the fluency of the adversaries. To highlight this, we use the universal sentence encoder and the cosine similarity score to compute the semantic resemblance of successful adversarial examples. Shown in Figure. 3, we illustrate the distribution of the similarity score across the different attacks and use-cases. Overall, most recipes perform fairly well at maintaining the semantics of the sentence, nevertheless we see that the variance is lower for TextBugger. Meanwhile, *VIPER*, is the poorest when it comes to creating semantically similar adversaries, mainly because of its techniques which replaces characters with visually similar ones.

5. Shielding

In this section, we present the logic and results of our experiments testing the effectiveness of adversarial training and binary adversarial filters. For space constraint, we only highlight results over the Fake News and Stock Index use cases.

Table 4. Adversarial detection evaluation (PWWS)

	Class	Precision	Recall	F1Score
Fake News	Org	0.82	0.69	0.75
	Adv	0.73	0.85	0.78
DowJones	Org	0.57	0.48	0.52
	Adv	0.55	0.63	0.59

5.1. Adversarial detection

Building a binary classifier for original and adversarial examples is a well-known shielding technique, it serves as a proxy filtering out adversarial examples before feeding it to the model under attack. To train such a binary classifier, a dataset of original and adversarial examples is required. To that aim, we select around 1K samples for each use case and attack them with the PWWS method. The resulting (successful) adversarial candidates are used to fine-tune an out-of-the-box BERT model. The results can be seen in Table. 4. As expected, the results are use-case specific and while for Fake News 85% of adversaries can be identified as such, for the Dow Jones model, a recall of only 63% is achieved. Note that in general the Fake News detection outperforms its Dow Jones counterpart. Furthermore, we believe the results can still be enhanced through parameters fine-tuning and creating a bigger training set.

[1] https://www.gatesnotes.com/The-Age-of-AI-Has-Begun

[2] https://atlas.mitre.org/

[3] https://github.com/huggingface/transformers

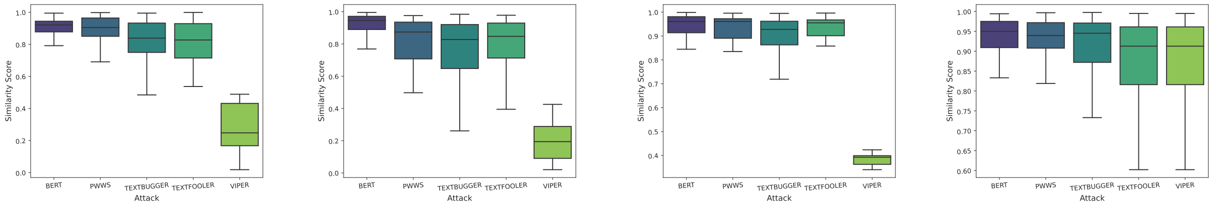
[4] https://attack.mitre.org/

[5] https://www.kaggle.com/c/fake-news

[6] https://www.kaggle.com/datasets/aaron7sun/stocknews

[7] https://newsapi.org/

[8] https://azure.microsoft.com/en-us/



(a) FakeNews (b) ESG Topic (c) DowJones Closing (d) ESG Sentiment

Figure 3. Semantic similarity (Original vs. Adversaries)

Table 5. Adversarial training evaluation (PWWS)

	Method	Attack Success
Fake News	Original	62%
	+Adv training	No Benefit
DowJones	Original	23%
	+Adv training	No Benefit

5.2. Adversarial training

Following the guidelines of (Li et al., 2020) we test the efficiency of adversarial training as a shielding method. By fine-tuning the target model, with the PWWS generated adversarial samples (see previous subsection), and afterwards evaluating on the same holdout test set, we can compare the difference in robustness. As highlighted in Table. 5, adversarial training might not be very effective and in some cases can be counter productive reducing the overall accuracy of the model and even resulting in over-fitting.

Overall, adversarial training effectiveness is use case dependent and further investigation is required when and how adversarial training is optimal to use (Wang & Bansal, 2018; Wang et al., 2021a; Wang & Wang, 2020).

6. Conclusions

In this work, we introduce an early version of *AdversNLP*, a practical guide for assessing NLP model robustness against text adversarial attacks. Our framework combines the technical tools of OpenAttack and TextAttack with the best practices from the ATLAS MITRE framework, providing a user-friendly interface to automate adversarial attacks and illustrate results using key performance indicators (KPIs). *AdversNLP* makes it easier for non-technical and practitioners to assess and improve the security of their NLP models. We hope that this work will inspire further contributions and spark the development of more robust and secure NLP models.

7. Future works

Multiple features are currently under development and will be integrated in the new update of *AdversNLP*. For instance, automated assessment of adversarial training and binary adversaries filters effectiveness (as explained in Section. 5). Furthermore, we are investigating the usage of Azure OpenAI Services to process and combine the practical audit guide input, the robustness and shielding assessment and generate a tailored report, providing a through analysis of the targeted use case and step by step recommendations to enhance robustness. Finally, due to rise of LLMs in public discourse, and the early stage maturity of LLM applications, we are looking into integrating automatic safety checks and risk mitigation mechanisms for LLMs into our framework. Although new vulnerabilities are discovered regularly, OWASP provides a good starting point by releasing a list of ten known vulnerabilities (OWASP, 2023).

References

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. Generating natural language adversarial examples, 2018.

Belinkov, Y. and Bisk, Y. Synthetic and natural noise both break neural machine translation, 2018.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Cheng, M., Yi, J., Chen, P.-Y., Zhang, H., and Hsieh, C.-J. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples, 2020.

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. Hotflip: White-box adversarial examples for text classification, 2018.

Eger, S., Şahin, G. G., Rücklé, A., Lee, J.-U., Schulz, C., Mesgar, M., Swarnkar, K., Simpson, E., and Gurevych, I. Text processing like humans do: Visually attacking and shielding nlp systems, 2020.

Gao, J., Lanchantin, J., Soffa, M. L., and Qi, Y. Black-box generation of adversarial text sequences to evade deep learning classifiers, 2018.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks, 2018.

Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. Is bert really robust? a strong baseline for natural language attack on text classification and entailment, 2020.

Li, J., Ji, S., Du, T., Li, B., and Wang, T. TextBugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*. Internet Society, 2019. doi: 10.14722/ndss.2019.23138. URL https://doi.org/10.14722%2Fndss.2019.23138.

Li, L., Ma, R., Guo, Q., Xue, X., and Qiu, X. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6193–6202, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.500. URL https://aclanthology.org/2020.emnlp-main.500.

Liang, B., Li, H., Su, M., Bian, P., Li, X., and Shi, W. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, jul 2018. doi: 10.24963/ijcai.2018/585. URL https://doi.org/10.24963%2Fijcai.2018%2F585.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, 2020.

OWASP. Owasp top 10 for llms. 2023. URL https://owasp.org/www-project-top-10-for-large-language\-model-applications/.

Pruthi, D., Dhingra, B., and Lipton, Z. C. Combating adversarial misspellings with robust word recognition, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL https://arxiv.org/abs/1910.10683.

Ren, S., Deng, Y., He, K., and Che, W. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1103. URL https://aclanthology.org/P19-1103.

Samanta, S. and Mehta, S. Towards crafting text adversarial samples, 2017.

Sato, M., Suzuki, J., Shindo, H., and Matsumoto, Y. Interpretable adversarial perturbation in input embedding space for text, 2018.

Wang, X., Jin, H., Yang, Y., and He, K. Natural language adversarial defense through synonym encoding, 2021a.

Wang, X., Xiong, Y., and He, K. Detecting textual adversarial examples through randomized substitution and vote, 2021b. URL https://arxiv.org/abs/2109.05698.

Wang, Y. and Bansal, M. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2091. URL https://aclanthology.org/N18-2091.

Wang, Z. and Wang, H. Defense of word-level adversarial attacks via random substitution encoding, 2020.

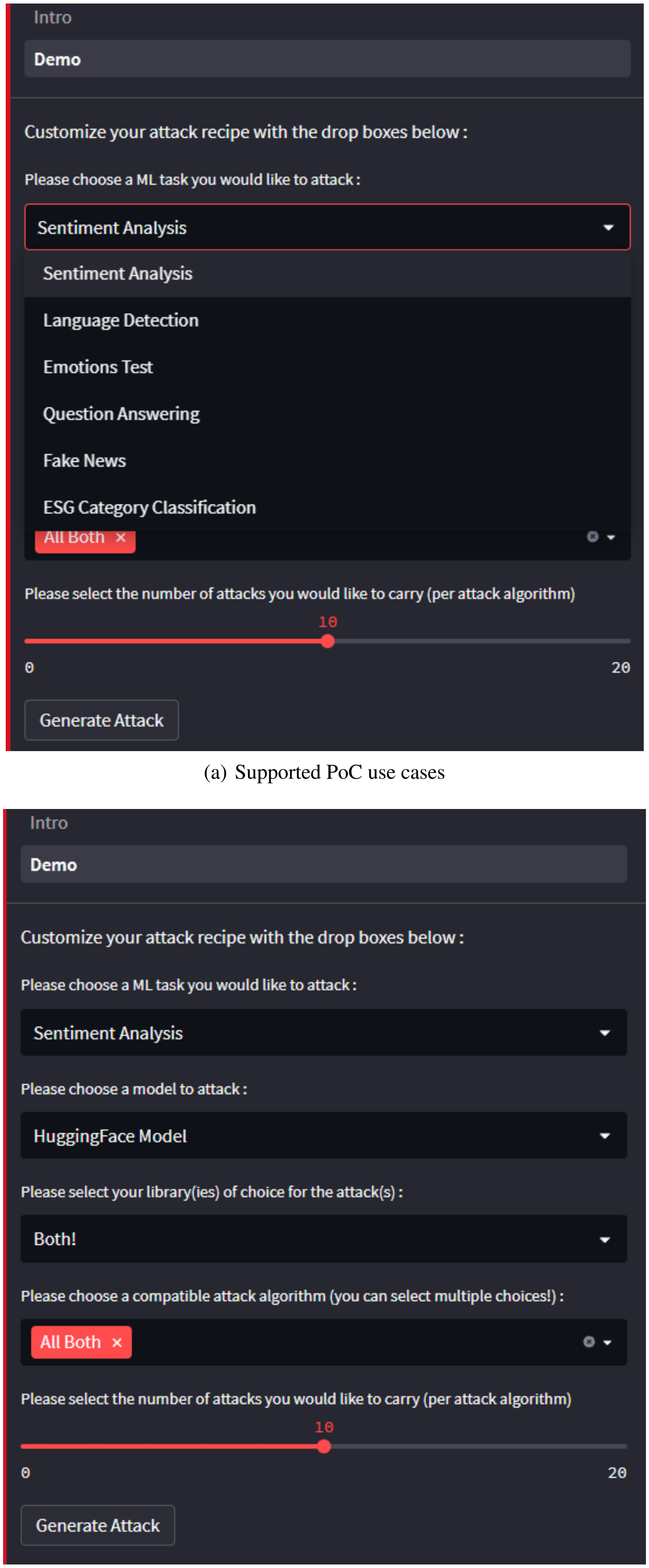
Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. Bloomberggpt: A large language model for finance, 2023.

Yin, X., Kolouri, S., and Rohde, G. K. Gat: Generative adversarial training for adversarial example detection and robust classification, 2022.

Yoo, K., Kim, J., Jang, J., and Kwak, N. Detection of word adversarial examples in text classification: Benchmark and baseline via robust density estimation, 2022. URL https://arxiv.org/abs/2203.01677.

Zeng, G., Qi, F., Zhou, Q., Zhang, T., Hou, B., Zang, Y., Liu, Z., and Sun, M. Openattack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 363–371, 2021. doi: 10.18653/v1/2021.acl-demo.43. URL https://aclanthology.org/2021.acl-demo.43.

Zhou, Y., Jiang, J.-Y., Chang, K.-W., and Wang, W. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1496. URL https://aclanthology.org/D19-1496.



Customize your attack recipe with the drop boxes below :

Please choose a ML task you would like to attack : Sentiment Analysis

Please select the number of attacks you would like to carry (per attack algorithm): 10

(a) Supported PoC use cases

Customize your attack recipe with the drop boxes below :

Please choose a ML task you would like to attack : Sentiment Analysis

Please choose a model to attack : HuggingFace Model

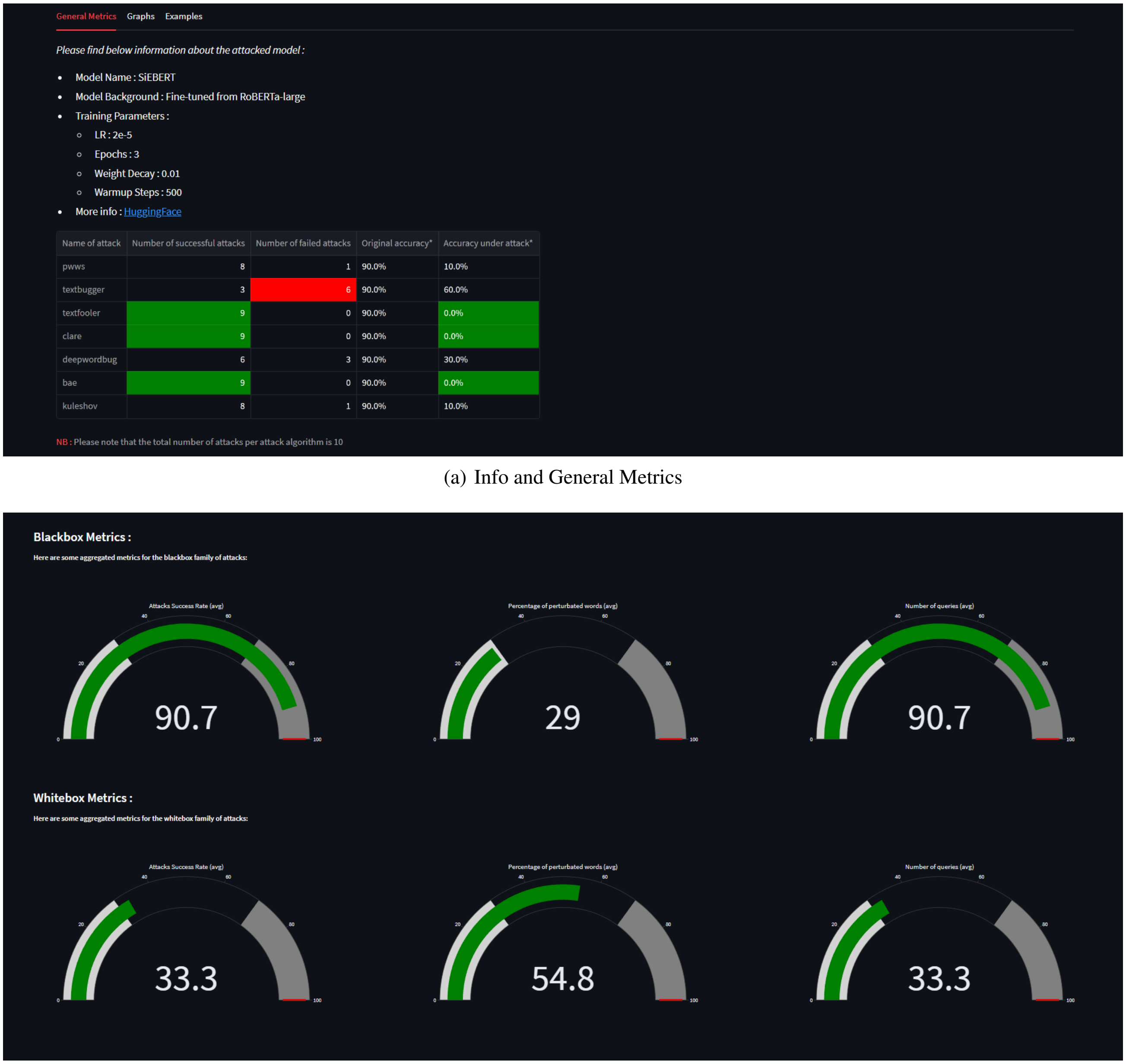
Please select your library(ies) of choice for the attack(s) : Both!

Please choose a compatible attack algorithm (you can select multiple choices!) : All Both ×

Please select the number of attacks you would like to carry (per attack algorithm): 10

(b) Attack scope and parameters

Figure 4. AdversNLP UI : Customizing simulated attacks



Please find below information about the attacked model :

- Model Name : SiEBERT
- Model Background : Fine-tuned from RoBERTa-large
- Training Parameters :
  - LR : 2e-5
  - Epochs : 3
  - Weight Decay : 0.01
  - Warmup Steps : 500
- More info : HuggingFace

Name of attack	Number of successful attacks	Number of failed attacks	Original accuracy*	Accuracy under attack*
pwws	8	1	90.0%	10.0%
textbugger	3	6	90.0%	60.0%
textfooler	9	0	90.0%	0.0%
clare	9	0	90.0%	0.0%
deepwordbug	6	3	90.0%	30.0%
bae	9	0	90.0%	0.0%
kuleshov	8	1	90.0%	10.0%

NB : Please note that the total number of attacks per attack algorithm is 10

(a) Info and General Metrics

Blackbox Metrics :

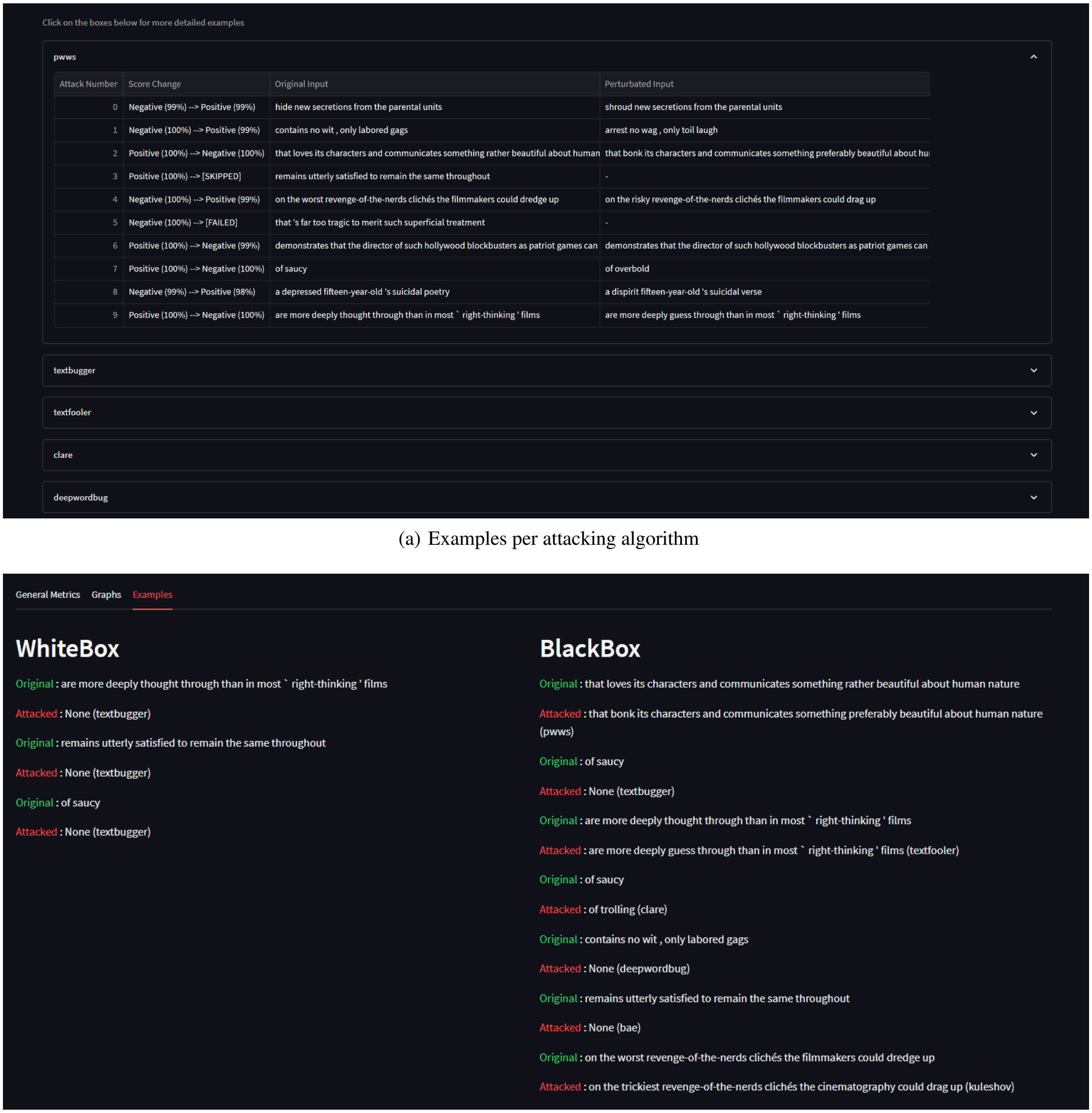
Here are some aggregated metrics for the blackbox family of attacks:

Whitebox Metrics :

Here are some aggregated metrics for the whitebox family of attacks:

(b) Black Box vs White Box KPIs

Figure 5. AdversNLP UI : Model internal workings/ attack stats and aggregated KPIs



Click on the boxes below for more detailed examples

pwws

Attack Number	Score Change	Original Input	Perturbated Input
0	Negative (99%) --> Positive (99%)	hide new secretions from the parental units	shroud new secretions from the parental units
1	Negative (100%) --> Positive (99%)	contains no wit , only labored gags	arrest no wag , only toil laugh
2	Positive (100%) --> Negative (100%)	that loves its characters and communicates something rather beautiful about human	that bonk its characters and communicates something preferably beautiful about hu
3	Positive (100%) --> [SKIPPED]	remains utterly satisfied to remain the same throughout	-
4	Negative (100%) --> Positive (99%)	on the worst revenge-of-the-nerds clichés the filmmakers could dredge up	on the risky revenge-of-the-nerds clichés the filmmakers could drag up
5	Negative (100%) --> [FAILED]	that 's far too tragic to merit such superficial treatment	-
6	Positive (100%) --> Negative (99%)	demonstrates that the director of such hollywood blockbusters as patriot games can	demonstrates that the director of such hollywood blockbusters as patriot games can
7	Positive (100%) --> Negative (100%)	of saucy	of overbold
8	Negative (99%) --> Positive (98%)	a depressed fifteen-year-old 's suicidal poetry	a dispirit fifteen-year-old 's suicidal verse
9	Positive (100%) --> Negative (100%)	are more deeply thought through than in most ` right-thinking ' films	are more deeply guess through than in most ` right-thinking ' films

textbugger

textfooler

clare

deepwordbug

(a) Examples per attacking algorithm

WhiteBox

Original : are more deeply thought through than in most ` right-thinking ' films

Attacked : None (textbugger)

Original : remains utterly satisfied to remain the same throughout

Attacked : None (textbugger)

Original : of saucy

Attacked : None (textbugger)

BlackBox

Original : that loves its characters and communicates something rather beautiful about human nature

Attacked : that bonk its characters and communicates something preferably beautiful about human nature (pwws)

Original : of saucy

Attacked : None (textbugger)

Original : are more deeply thought through than in most ` right-thinking ' films

Attacked : are more deeply guess through than in most ` right-thinking ' films (textfooler)

Original : of saucy

Attacked : of trolling (clare)

Original : contains no wit , only labored gags

Attacked : None (deepwordbug)

Original : remains utterly satisfied to remain the same throughout

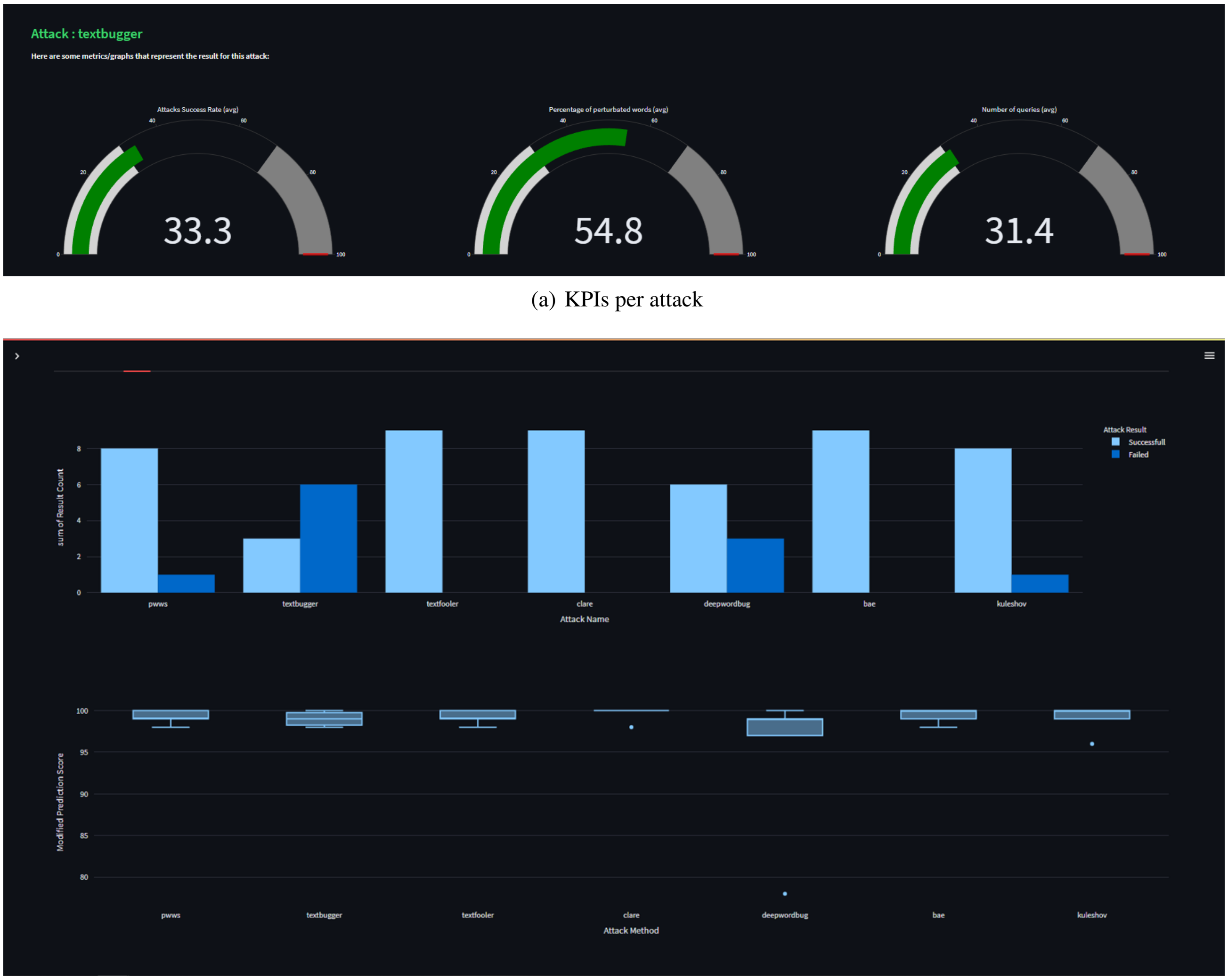
Attacked : None (bae)

Original : on the worst revenge-of-the-nerds clichés the filmmakers could dredge up

Attacked : on the trickiest revenge-of-the-nerds clichés the cinematography could drag up (kuleshov)

(b) Examples *BlackBox vs WhiteBox*

Figure 6. AdversNLP UI : Original vs Perturbed Samples



Attack : textbugger

Here are some metrics/graphs that represent the result for this attack:

(a) KPIs per attack

(b) Success rate and certainty distribution

Figure 7. AdversNLP UI : Attack results visualization