A baseline for detecting Textual Attacks in Sentiment Analysis Classification using Density Estimation

Vincent Nguyen
ENSAE Paris
vincent_nguyen@ensae.fr

Solal Jarreau *
ENSAE Paris
solal.jarreau@ensae.fr

Abstract

Building NLP models that are resistant to computer destabilization has become a key element of research in recent years. While models are becoming more and more reliable and robust, concerns about the exploitation of their flaws involve the construction of tools to guarantee their robustness and to protect against computer attacks. As a result, adversarial defense have been aggressively developed over the past decade, showing convincing results in improving the robustness of models and their resistance to attacks. However, another crucial tool in protecting from attacks is to improve word-adversarial attacks detection. In this paper, we evaluate the performance of two attack detection methods on two prepared datasets and two transformer-based models [1]. Our main goal is to investigate and confirm the results obtained in Yoo et al. (Yoo et al., 2022), using density estimation.

1 Introduction

Natural Language Processing (NLP) has seen significant advancements in recent years, with numerous applications in various domains, from chatbots to sentiment analysis. However, with these advancements come concerns about the safety and fairness of NLP models. In particular, there is a growing awareness of the importance of fairness (Colombo et al., 2021; Pichler et al., 2022; Colombo et al., 2022b), out-of-distribution (OOD) generalization (Colombo, 2021), and adversarial defense (Yoo et al., 2022).

1.1 Adversarial attacks

Adversarial attacks in NLP have been identified as a problem for many years (Hulten et al., 2004; Picot et al., 2023a,b), especially as NLP models are being increasingly used in critical fields. They refer to the deliberate manipulation of NLP models to produce erroneous or misleading outputs.

These attacks are usually carried out by injecting specially crafted input data that can trick NLP models into making incorrect predictions, therefore known as evasion attacks (Barreno et al., 2006). Over the previous years, many adversarial attacks have shown success in destabilizing transformer-based models, especially with the rise of many easy-to-use frameworks, such as TextAttack (Morris et al., 2020) or OpenAttack (Zeng et al., 2021).

To tackle this issue, many papers in recent years has tried to improve the robustness of NLP algorithms, using various techniques of adversarial defense. Among them, robust encoding involves using the average embedding of all characters of a given word to protect against character-level adversarial example attacks (Belinkov and Bisk, 2017) ; randomization involves generating embedding vectors for each word from a convex hull of every single word and its synonyms, which improves the model's performance against adversarial attacks by synonym substitution (Zhou et al., 2021) ; and adversarial training involves combining specific adversarial examples with the original inputs as an augmented dataset to significantly enhance the model's robustness (Wang et al., 2021).

However, the detection of adversarial attacks has recently emerged as one of the fundamental issues in the development of NLP models. Indeed, the drawbacks of the above mentioned defense techniques are that they are often costly and prove to be inadequate in a number of situations for which the simple detection of adversarial attacks would be sufficient. In this respect, several recent papers have worked to develop a baseline for enhancing the detection of adversarial attacks (Xie et al., 2022) (Mozes et al., 2021). The pa-

*The two authors have equal contribution.

[1] https://github.com/VincentNg5/NLP_project

per by Yoo et al. (Yoo et al., 2022) gives a general overview, on four datasets and with four NLP models, of the main detection methods and their ability to efficiently detect adversarial attacks, using three different scores. In particular, the authors introduce a new detection method, based on robust density estimation, which we will try to reproduce in the rest of this paper.

2 Experiments Protocol

Our main goal is to evaluate an adversarial detection technique that is based on robust density estimation through Kernel PCA and Minimum Covariance Determinant. To do so, we use the widely-used IMDb Dataset (Maas et al., 2011).

2.1 Dataset

The IMDb dataset is a commonly used dataset for sentiment analysis, consisting of a collection of 50,000 movie reviews, split evenly into a training set and a test set. The dataset is labeled with binary sentiment labels, where 0 denotes a negative sentiment and 1 denotes a positive sentiment. The reviews were originally collected on the International Movie Database (IMDb), and the Dataset is loaded from HuggingFace. The dataset has been widely used for evaluating the performance of sentiment analysis models, and has been used in numerous studies in the field.

2.2 Construction of the adversarial examples

As the attacks are expensive to reproduce, we have recovered the attacks already created by Yoo et al. in their paper already quoted (Yoo et al., 2022). To do this, the dataset was split into two parts S_1 and S_2. Only the S_1 part was used to generate the attacks in TextAttack, and the successful contradictory examples were kept. The reviews in subset S_2 are kept as is and form the clean part of the dataset. Subsequently, two Transformer-based models are used, that both lie on the BERT model. BERT (standing for Bidirectional Encoder Representations from Transformers) is a pre-training method developed by Google researchers in 2018 (Devlin et al., 2018). The key innovation of this model is its ability to pre-train a deep bidirectional representation of text by jointly conditioning on both left and right contexts in all layers of the transformer. This means that BERT is able to capture the context of a given word not just from the words that come before it, but also from the words that come after it. BERT was pre-trained on a large corpus of text data, specifically the BooksCorpus and English Wikipedia datasets, and achieved state-of-the-art results on a range of NLP benchmarks.

The second model that we use, RoBERTa (Robustly Optimized BERT Pretraining Approach) is based on the BERT architecture. It was developed by Facebook AI researchers in 2019 (Liu et al., 2019). The main improvement of RoBERTa over BERT is its training strategy. The RoBERTa model is trained on a much larger and more diverse dataset than the original BERT model, and it also uses a longer sequence length during training. This allows RoBERTa to capture more complex relationships between words and phrases, and to better model the nuances of natural language.

2.3 Robust Density Estimation

Following the Yoo et al. paper, we build a strategy of adversarial detection that uses Robust Density Estimation with a Kernel-Principal Component Analysis and Minimum Covariance Determinant. The Principal Component Analysis aims at avoiding the curse of dimensionality that would arise from the high dimension of transformers-based models ($D = 768$), as well as the presence of outliers. However, PCA can only model linear relationships between variables, therefore we use Kernel PCA that extends the linear PCA approach by applying a non-linear transformation to the input data. This transformation maps the input data into a higher-dimensional feature space, where it becomes easier to model non-linear relationships between variables. Considering N centered samples $Z_{train} \in R^{D\times N} = [z_1, ..., z_N]$, a mapping function $\varphi : R^D \to R^{D_0}$, and its mapping applied to each sample $\Phi(Z_{train}) \in R^{D_0\times N}$, kernel PCA projects the data points to the eigenvectors with the P largest eigenvalues of the covariance $\Phi(Z_{train})\Phi(Z_{train})^T$.

The second problem that may arise from our data is that the use of a Kernel-PCA haven't tackled the problem of outliers that are present in the data. As a consequence, we use a Minimum Covariance Determinant (MCD), which is a robust statistical method used to estimate the parameters of a multivariate normal distribution in the presence of outliers (Rousseeuw, 1984). Given a set of n d-dimensional observations $X = x_1, x_2, ..., x_n$, the goal of MCD is to find a subset of the data Y with size $m \leq n$ that minimizes the determinant of its covariance matrix, subject to the constraint $m \geq d + 1$. By minimizing the influence of outliers on the covariance matrix, the MCD is able to provide more accurate estimates of the true underlying parameters of the distribution.

2.4 Implementation details

We load pretrained models provided by TextAttack [2]. Both kPCA and MCD are computed using the scikit-learn library (Pedregosa et al., 2012). We choose the input of the Kernel PCA to be the output of the last attention layer. We use radial basis function kernel and reduce the dimensions of the feature to $P = 100$. For MCD, we choose the default value of support fraction provided by scikit-learn.

2.5 Evaluation of the results

To evaluate the performances of our model, we use three different scores that gives us different informations. For all metrics, higher means better. First of all, the F1 score is a commonly used metric in classification tasks that measures the harmonic mean of precision and recall, providing a single score that highlights the trade-off between precision and recall. Secondly, the recall (or true positive rate) is a performance metric that measures the proportion of true positive predictions out of all actual positive instances in the data. In other words, it is the ratio of correctly predicted positive instances to all positive instances. Both of theses scores depends on the false positive rate (FPR) that we have fixed at 0.1, which means that 10% of normal samples are predicted to be attacks. However, the fact that these metrics depends on this parameter is inconvenient.

The other score is the the area under the receiver operating characteristic curve (AUC-ROC) and does not depends on the FPR. That is why the AUC score is the preferred metric in our paper. The ROC curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The AUC-ROC represents the probability that a randomly selected positive sample is ranked higher than a randomly selected negative sample, and is equal to the area under the ROC curve. A perfect classifier has an AUC-ROC of 1, while a random classifier has an AUC-ROC of 0.5. The AUC-ROC is a useful metric for evaluating classifier performance when the classes are imbalanced and the cost of false positives and false negatives are different.

3 Results

3.1 Log-likelihood with or without MinCovDet

The graphs below highlight the value of using a MinCovDet. The first graph presents the results of the log-likelihood of the covariance matrix obtained from the Kernel-PCA. We can clearly see that the log-likelihood is rather high for the clean examples, while the adversarial examples have a lower (and more spread out) log-likelihood.

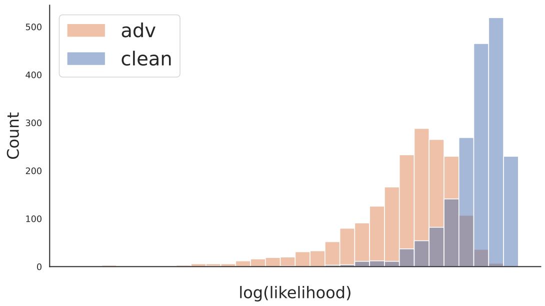

Figure 1: Log-likelihood of the covariance matrixes without MinCovDet

However, we note that in the second example, the use of the Minimal Covariance Determinant allows us to obtain much tighter results. Indeed, the MCD estimator assumes that a subset of the data is drawn from a multivariate normal distribution, and the subset is chosen to minimize the covariance matrix determinant. Therefore, the outliers are removed and the results seem clearly better, with a clear separation between the adversarial distribution and the clean distribution.

3.2 AUC and F1-Score

The table below resumes the performances of the Robust Density Estimation, for both BERT and RoBERTa. The most informative results have been put in bold. First, it appears that most of the models using mincovdet achieve higher AUC, F1, and recall scores compared to their counterparts using simple covariance matrices (cov). This suggests that our approach using mincovdet can be a better approach for tasks of adversarial detection.

[2] https://huggingface.co/textattack

BERT Models				
Metric	**MCD with TextFooler**	**Covariance with TextFooler**	**MCD with PWWS**	**Covariance with PWWS**
AUC	0.94	0.92	0.94	0.92
F1	0.85	0.80	0.84	0.79
Recall	0.82	0.73	0.80	0.72

BERT Models (continued)				
Metric	**MCD with BAE**	**Covariance with BAE**	**MCD with TF-adj**	**Covariance with TF-adj**
AUC	0.92	0.90	0.9	0.87
F1	0.80	0.72	0.77	0.66
Recall	0.73	0.63	0.69	0.55

RoBERTa Models				
Metric	**MCD with TextFooler**	**Covariance with TextFooler**	**MCD with PWWS**	**Covariance with PWWS**
AUC	0.95	0.92	0.95	0.94
F1	0.88	0.85	0.88	0.86
Recall	0.86	0.82	0.88	0.84

RoBERTa Models (continued)				
Metric	**MCD with BAE**	**Covariance with BAE**	**MCD with TF-adj**	**Covariance with TF-adj**
AUC	0.95	0.92	0.90	0.87
F1	0.87	0.83	0.76	0.73
Recall	0.85	0.78	0.68	0.64

Table 1: Performance Metrics for BERT and RoBERTa models

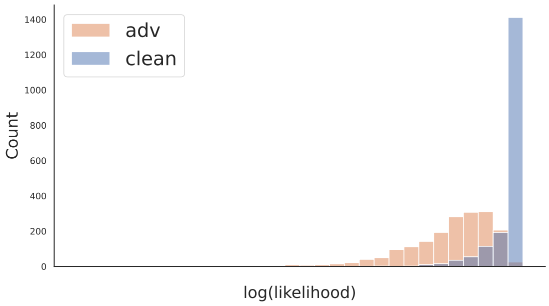

Figure 2: Log-likelihood of the covariance matrixes with MinCovDet

For instance, the RobeRTa with Textfooler attacks achieves a AUC score of 0.95, which is higher than the AUC score of its counterpart using simple covariance matrices with a score of 0.92.

Additionally, the Roberta models generally perform better than the BERT models across all metrics, which is probably due to differences in model architecture or training data (as mentioned above). Looking at the table, we can see that for all of the models tested, the highest scores (AUC, F1, and recall) are achieved by the RoBERTa models. As an example, we can compare the AUC scores and F1 scores of the models with PWWS attacks and MinCovDet. The RoBERTa model achieved an AUC score of 0.95, while the BERT model achieved an AUC score of 0.94. Similarly, RoBERTa achieved an F1 score of 0.88, while BERT achieved an F1 score of 0.84.

In summary, our method applied to transformers-based embeddings achieves good performance, with a AUC score greater than 0.9 for each attack.

4 Conclusion

In this work, we introduce a method based on density estimation for adversarial detection for transformers models in NLP. We find that our out-of-distribution detection works well and can handle unseen attacks. However, there remains room for future research on robustness, for example against stronger adversaries who use adaptive attacks. Another direction for future improvement could be to take into consideration information available in all the hidden layers of the transformers architecture (Colombo et al., 2022a; Darrin et al., 2023b) and to test our methods on OOD detection (Darrin et al., 2023a).

References

Peter Rousseeuw. 1984. Least median of squares regression. *Journal of the American statistical association*, 79:871–880.

Geoff Hulten, Anthony Penta, Gopalakrishnan Seshadrinathan, and Manav Mishra. 2004. Trends in spam products and methods.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony Joseph, and J. Tygar. 2006. Can machine learning be secure? volume 2006, pages 16–25.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2012. Scikit-learn: Machine learning in python.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

John 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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021. Adversarial training with fast gradient projection method against synonym substitution based text attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):13997–14005.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5482–5492, Online. Association for Computational Linguistics.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186, Online. Association for Computational Linguistics.

Pierre Colombo. 2021. *Learning to represent and generate text using information measures*. Ph.D. thesis, (PhD thesis) Institut polytechnique de Paris.

Pierre Colombo, Chloe Clavel, and Pablo Piantanida. 2021. A novel estimator of mutual information for learning to disentangle textual representations. *() ACL 2021*.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. 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*, pages 363–371, Online. Association for Computational Linguistics.

Pierre Colombo, Eduardo D. C. Gomes, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022a. Beyond mahalanobis-based scores for textual ood detection.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of word adversarial examples in text classification: Benchmark and baseline via robust density estimation.

Georg Pichler, Pierre Jean A Colombo, Malik Boudiaf, Günther Koliander, and Pablo Piantanida. 2022. A differential entropy estimator for training neural networks. In *() ICML 2022*.

Zhouhang Xie, Jonathan Brophy, Adam Noack, Wencong You, Kalyani Asthana, Carter Perkins, Sabrina Reis, Sameer Singh, and Daniel Lowd. 2022. Identifying adversarial attacks on text classifiers.

Pierre Colombo, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. 2022b. Learning disentangled textual representations via statistical measures of similarity. *ACL 2022*.

Marine Picot, Nathan Noiry, Pablo Piantanida, and Pierre Colombo. 2023a. Adversarial attack detection under realistic constraints.

Maxime Darrin, Pablo Piantanida, and Pierre Colombo. 2023a. Rainproof: An umbrella to shield text generators from out-of-distribution data. *arXiv preprint arXiv:2212.09171*.

Marine Picot, Guillaume Staerman, Federica Granese, Nathan Noiry, Francisco Messina, Pablo Piantanida, and Pierre Colombo. 2023b. A simple unsupervised data depth-based method to detect adversarial images.

Maxime Darrin, Guillaume Staerman, Eduardo Dadalto Câmara Gomes, Jackie CK Cheung, Pablo Piantanida, and Pierre Colombo. 2023b. Unsupervised layer-wise score aggregation for textual ood detection. *arXiv preprint arXiv:2302.09852*.