Enabling Contextual Soft Moderation on Social Media through Contrastive Textual Deviation

Pujan Paudel, Mohammad Hammas Saeed, Rebecca Auger, Chris Wells, and Gianluca Stringhini, *Boston University*

https://www.usenix.org/conference/usenixsecurity24/presentation/paudel-enabling

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 33rd USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Artifact Appendices to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

USENIX Security '24 Artifact Appendix: Enabling Contextual Soft Moderation on Social Media through Contrastive Textual Deviation

Pujan Paudel, Mohammad Hammas Saeed, Rebecca Auger, Chris Wells, and Gianluca Stringhini

A Artifact Appendix

A.1 Abstract

This document presents the artifact of our Usenix Security'24 paper: "Enabling Contextual Soft Moderation on Social Media through Contrastive Textual Deviation." We first present a brief description of our artifact, alongside the hardware and software requirements needed to run the models and reproduce the experimental results associated with the artifact. We then describe the steps necessary to download the models and benchmarks used in the artifact, followed by a simple functionality test ensuring that the model proposed as part of the artifact can run inference properly against all the benchmarks. Next, we detail the evaluation workflow by discussing the major claims made in our paper, and subsequent experiments associated with the claims.

A.2 Description & Requirements

The primary evaluation environment of this artifact is preferred through interactive Python Jupyter notebooks. The datasets and models of the artifacts are interfaced through the Huggingface framework for easy and robust running of experiments for artifact evaluation purposes. Additionally, we also provide a Github repository with helper evaluation scripts to reproduce the benchmarking results.

A.2.1 Security, privacy, and ethical concerns

There are no risk for evaluators when executing this artifact in their machines or in the cloud. All datasets used in this work were either publicly released by other researchers or were collected using publicly available APIs and following those API's terms of service.

A.2.2 How to access

All the models and datasets associated with this artifact are publicly hosted in the archived Huggingface collection [1]. A set of companion Jupyter notebooks with experimental setup for the major claims, and detailed software dependencies listed in *requirements.txt* is available in our Github repository[2].

A.2.3 Hardware dependencies

The CTD models forming the core of the artifact have some important hardware requirements to run succesfully. Both the Google's FLAN-T5-XXL model (bootstrapped) and the fine-tuned CTD model need an estimated GPU vRAM of 22 GB to load the models and run evaluation on the benchmarks. Therefore, it is recommended to have access to an evaluation machine with a minimum of 22GB GPU memory, and a recommended GPU compute capability of 7.0. Additionally, 2 CPU cores alongside the aforementioned GPU hardware requirements are recommended for seamless evaluation.

A.2.4 Software dependencies

The major environmental dependencies needed during the artifact evaluation period is listed below.

- CUDA Toolkit (v11.3)
- CUDNN (v8.2)
- GCC (v9.3.0)

The list of software packages and the corresponding version and commit reference for these packages is available in *requirements.txt* of the companion repository. It is recommended that for succesful evaluation of the artifact, proper environmental dependencies as listed above are met, and appropriate packages are installed with the command *pip install -r requirements.txt*

A.2.5 Benchmarks

The 4 different datasets and the fine-tuned CTD model required by the experiments with this artifact are organized in the models and datasets category of the corresponding Huggingface collection. Each of the experimental notebooks associated with the claim load the corresponding models and datasets at the beginning, and once the models and datasets are downloaded, they are cached internally as per *transformer* library's default behavior.

[1] https://huggingface.co/collections/ppaudel/contrastive-textual-deviation-65e20c48680724cc9a809062

[2] https://github.com/codepujan/ctd-artifact/tree/v1.0.0

A.3 Set-up

The setup procedure for this artifact includes ensuring appropriate hardware requirementsand software requirements are met.

A.3.1 Installation

The process of downloading and installing the dependencies can be completed by running the script *setup.py* on the artifact companion repository.

A.3.2 Basic Test

A basic functionality test ensuring all the models and datasets are loaded properly can be verified by running the script *basic_test.py* on the artifact companion repository. The script loads all the benchmarks and runs an inference test with the bootstrapped model and the fine-tuned CTD model on all the benchmark dataset. Evaluators can expect the model's prediction output (i.e. *Support*, or *Refute*) for each of the dataset, alongside the expected output label from the groundtruth dataset.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *Bootstrapped Contrastive Textual Deviation (CTD) achieves better performance in stance detection over supervised baselines. This is proven by the experiment (E1) described in [Section 4.2 Bootstrapping CTD using LLMs] whose results are illustrated/ reported in [Table 6].*

(C2): *CTD Fine-tuned on the Perspectrum dataset (Fine-tuned CTD) achieves better performance in stance detection over bootstrapped CTD and other supervised baselines. This is proven by the experiment (E2) described in [Section 5.2 Comparison with existing baselines] whose results are illustrated / reported in [Table 7].*

(C3): *CTD Fine-tuned on the Perspectrum dataset (Fine-tuned CTD) can be integrated as a post-retrieval filter on results of existing soft moderation systems to further improve their F1 score.*

A.4.2 Experiments

(E1): *[Bootstrapping CTD] [2 human-minutes + 0.5 compute-hour + 1 GB disk]: Bootstrapped CTD performs better than supervised methods in Climate Skepticism dataset .*
How to: *The step-by-step flow to reproduce the results of this experiment is packaged in the Python script experiment1_bootstrapping.py present in the artifact companion repository.*
Preparation: *Make sure the "Basic Setup" test run succesfully.*
Execution: *Once the basic setup tests are confirmed succesfully, run the Python script experiment1_bootstrapping.py and observe the results for each benchmarks.*
Results: *The results corresponding to Table 6 are printed in the terminal.*

(E2): *[Evaluating Fine-tuned CTD] [2 human-minutes + 1 compute-hour + 1 GB disk]: CTD Fine-tuned on the Perspectrum dataset (Fine-tuned CTD) achieves better performance in stance detection over bootstrapped CTD and other supervised baselines.*
How to: *The step-by-step flow to reproduce the results of this experiment is packaged in the Python script experiment2_evaluating.py present in the artifact companion repository.*
Preparation: *Make sure the "Basic Setup" tests run succesfully.*
Execution: *Once the basic setup tests are confirmed succesfully, run the Python script experiment2_evaluating.py and observe the results for each benchmarks.*
Results: *The results corresponding to Table 7 are printed in the terminal.*

(E3): *[Integrating CTD on Election Denial Dataset] [2 human-minutes + 0.5 compute-hour + 1 GB disk]: Fine-tuned CTD can be integrated as a post-retrieval filter on results of existing soft moderation systems to further improve their F1 score.*
How to: *The step-by-step flow to reproduce the results of this experiment is packaged in the Python script experiment3_integrating.py alongside the artifact companion repository.*
Preparation: *Make sure the "Basic Setup" tests run succesfully.*
Execution: *Once the basic setup tests are confirmed succesfully, run the Python script experiment3_integrating.py and observe the results for each benchmarks.*
Results: *The results corresponding to Table 9 will printed in the terminal.*

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2024/.