DETECTING AI GENERATED TEXT USING NEURAL NETWORKS

A Graduate Thesis

by

JESSE A GUERRERO

Submitted to Office of Graduate Studies
Texas A&M University-San Antonio
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2023

Major Subject: Computer Science

DETECTING AI GENERATED TEXT USING
NEURAL NETWORKS

A
Thesis
by
JESUS A.
GUERRERO

Approved as to style and content by:

Izzat Alsmadi

Izzat Alsmadi Ph. D.
Associate Professor
(Committee Chair)

Gongbo Liang, Ph.D.
Assistant Professor
(Committee Member)

Zechun Cao

Zechun Cao, Ph.D.
Assistant Professor
(Committee Member)

Izzat Alsmadi

Izzat Alsmadi Ph.D.
(Department Chair)

MAY 2023

ABSTRACT

Detecting AI Generated Text Using Neural Networks

May 2023

Jesse Guerrero, B.S, Texas A&M – San Antonio

Graduate Thesis Chair: Dr. Izzat Alsmadi

For humans, distinguishing machine generated text from human written text is mentally taxing and slow. NLP models have been created to do this more effectively and faster. But, what if some adversarial changes have been added to the machine generated text? This thesis discusses this issue and text detectors in general. The primary goal of this thesis is to describe the current state of text detectors in research and to discuss a adversarial issues in modern NLP transformers. Chapter 2 displays the current state of text detector literature as a Systematic Literature Review. Chapter 3 describes an experiment where RoBERTa was used to test transformers against simple mutations which cause mislabeling.

Dedication

I would like to thank my thesis advisors, Dr. Alsmadi & Dr. Liang. Their encouragement, knowledge and experience helped me navigate through the many challenges involved during the duration of this project. They taught me so much and opened many opportunities for me both academically and professionally, pushing me to do my absolute best. I would also like to thank my parents for their support that enabled me to attend college. They believed in me even when I did not believe in myself. Without them I would not had the confidence to even attempt college. Finally, I would like to thank God for granting me the strength and ability to persevere for this achievement.

Contents

Introduction

Detecting Deep Fake Text

Originally, the plan was to make models detecting video deep fakes by sequencing CNN frames of video and making some novel contribution. However, the academic body on vision was quite saturated and is/was difficult to contribute with true novelty. In preparation for the literature review of the thesis, natural language processing was chosen instead.

In replacement of detecting deep fake video, the idea was to detect deep fake text. With the switch, the number of research papers was comparatively quite small. For the SLR, there are only 50 papers in review because of this size. This SLR too, is the first of its kind in regards to NLP. Also, the papers inspired by this topic are unique and make a good contribution.

The motivation, as it had been, was to train a neural network to detect fake text. The proposed papers were in regards to creating detectors and attacking them. Though after the publication of Chat-GPT many more papers were added to the body quite quickly. Then, to make a more meaningful contribution the topics were changed to mutation, with the SLR coming in just in time as novel research before another literature review on detectors could be made.

Challenges with Deep Detectors

A major challenge with deep fake detectors is generalization to outside domains. For much of the literature each article featured a model which stuck to its trained domain. Twitter trained models stayed within twitter data, synthetic news stayed with news, generated academic papers stayed with academia. And this was necessary.

The classification accuracy rates of these models do not do well outside its domain unless subjected to many GPU hours and huge amounts of unique data. In the research given here, the experiments followed suite in sticking to image captioning for chapter 3.

Overtraining is another issue, similar to generalization. Even within a domain such as fake news or blogs, if one dataset is overused or there is some pattern or set vocabulary of the model, there will be over specification of the detector. Multiple datasets per domain is essential. This is especially true for prompt-answer generators such as Chat-GPT where the question asked can determine the pattern of the answer. Without some diversity of input data the model can be overtrained for specific Chat-GPT questions, or any other prompt text generators.

Another challenge is adversarial attacks. This is under researched and a challenge to creating text detectors. With simple changes to fake text it is possible to trick a detector. Mutation based adversarial attacks is covered in chapter 3 of the text. BERT transformers in particular appear to be quite susceptible to this. And the experiments given here propose a workflow for solving this issue.

Lastly, the field of deep learning and computer science in general develops very quickly. By the first edit of this thesis paper Chat-GPT gained publicity and many of the future steps mentioned in the below literature review are now being enacted. Though the SLR is still relevant, the future steps are being filled in. It easy to write a paper which at one point is highly important and relevant to then be outdated a year later. Still, there are so much open research avenues, it will probably take some time to be saturated.

Goals

The goal of this thesis is to make future work easier to accomplish for this domain. This is done by reviewing the state of the literature in regards to detecting fake text and to contribute to the literature by proposing a solution to mutation based adversarial attacks. Because in mid 2022 the body of research was so small, the review was the first of its kind in this domain. The review of the literature was made using the PRISMA methodology to sift down from a little over 1000 NLP research papers in search of those machine-centric text detector journals.

As well, chapter 3 on mutations, is an extension of Wolff [2020] where we used the COCO Image captioning dataset to create a potential solution to the transformer mutation problem. With these motivations set, the first order of business was to choose architectures for the given experiments. With the advice of Dr. Liang and Dr. Alsmadi, the proposed architecture was transformers. For the entirety of research RoBERTa was used to label sequenced data/text in a classical transformer fashion.

The idea is to mutate text to appear as human with small changes to words or letters. The most damaging component of mutation appeared to be using a vocabulary not recognized by the transformer. Automatically, the transformer took foreign vocabulary as human text and not synthetic. The research was based on this issue, which when published was quite novel.

Contributions

In this thesis, a snapshot of the literature was made and a workflow for beating mutation adversarial attacks was proposed with some example mutation operators. These contributions allow future authors to find new paths of research both for detecting fake text and attacking them with mutations.

The main contributions of this thesis are:

- **A collection of possible research avenues**: These will be mentioned multiple times throughout the text; low resource training, generalization, question response pattern recognition for generators like Chat-GPT, adversarial training for detectors and foreign language detectors.

- **An overarching view of the research done in 2022**: The finding from the literature review for this research domain is its status of being under researched.

- **Trends and statistics of the literature**: The statistic being increasing growth, predicted to increase much more in mid 2022. This help up to be true with the publicity to transformers in public media (GPT3.5)

- **A list of weaknesses the current overall research has**: These will be the same as research avenues. There are varying degrees as to which things are under researched, so that is written out in the SLR.
- **A workflow to using mutation operators on fake/human text from a previous paper**: People will adapt Chat-GPT to appear human. Predicting how they would do it, such as through pattern prompt response analysis will be necessary to catch these actions.
- **Nuances to adapting text detectors**: Throughout the text I mention statistics and nuances gathered as a result of experimentation to fine tuning a text detector.

Thesis Outline

The rest of this document consists of the following:

- **Background**: Provides context to the thesis paper in history as well as explains some of the technical parts necessary to understand the work.
- **Chapter 2**: Includes the literature review, surveying the current body of research. This is where we focus on trends, needs and statistics of what has already been written.
- **Chapter 3**: Here we focus on adversarial attacks via mutation using a unique dataset on unique mutation operators to create mutated text at run-time.
- **Conclusions**: Like the introduction we summarize everything from the literature as well as give some personal notes from creating the thesis.

Background

History & Context

Since the creation of the perceptron in the late 1950's, neural networks have been used as a theoretical model for machine learning but had been limited by the computational proficiency of our machines. Over the past three decades, increasing computational power has allowed neural network research to flourish at an unprecedented rate, not due to mathematical limitations, but by the processing ability of our machines. It all started with the perceptron, a larger layered mathematical formula, which processes numbers forward:

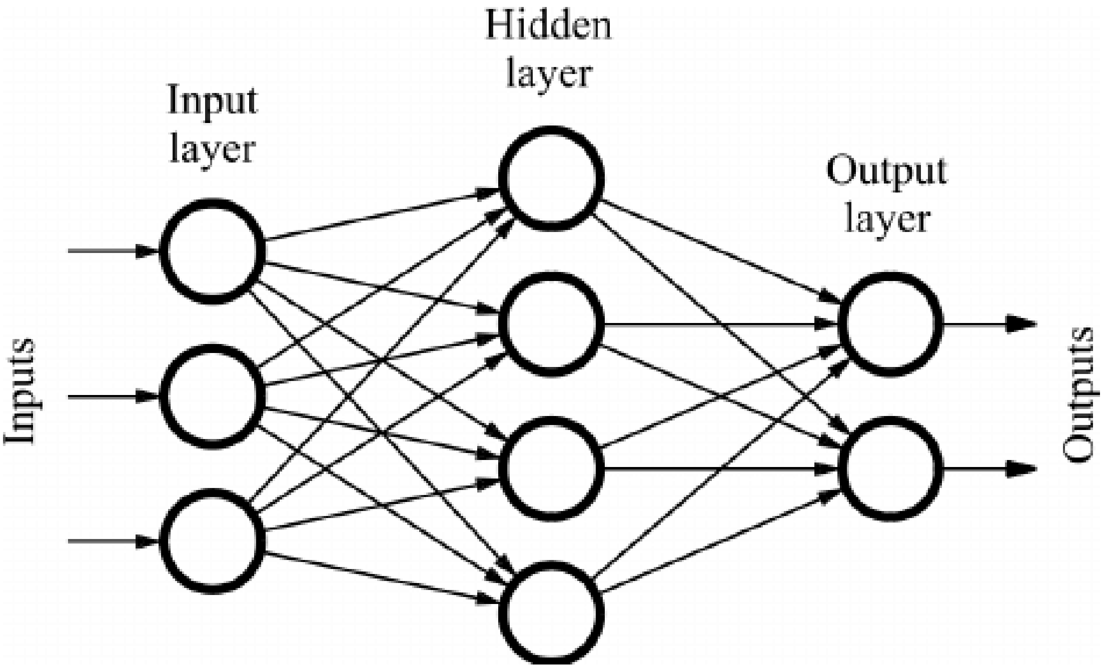


Figure 1.1: A basic perceptron

This above is a neural network and is typically much larger and grand. Numbers moving from left to right are known as "feeding forward" and from right to left, "back propagation". These concepts are old and dated, with math dating back to the 60s, 70s & 80s. Back propagation is the most complex of the two, requiring calculus to reverse the process in a "predictive" way.

In the 90s and early 2000s more research had been completed on these networks and with the advent of smaller and more powerful processing chips researchers were able to implement these mathematical models more commonly and with that more existing types of models were finally made as software.

Examples of neural networks include CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks) and Transformers in that chronological order. These, of course, have their own use cases due to how, mathematically, their output layer is calculated. But they are major parts of the academic body.

Today, particular fields of machine learning are more popular, such as vision, audio or video machine learning. NLP(natural language processing), before Chat-GPT, was much less popular and as such still has less research than other machine learning fields. Particularly, detecting machine generated text is under researched. Though, with the publication of Chat-GPT many more papers have been created.

The State of the Literature

In mid 2022 there was scarce research on detecting generated text. Just in the literature review in Chapter 2 a lot of time was spent gathering journals about the topic. Only 50 could be chosen by relevance, which is quite small for a general body of research.

Fake text detection is a very viable research field which has been growing faster and faster, though is still small and viable. If a paper is being written, it should be published quickly. The papers here are already becoming outdated with the publicity given to generated text.

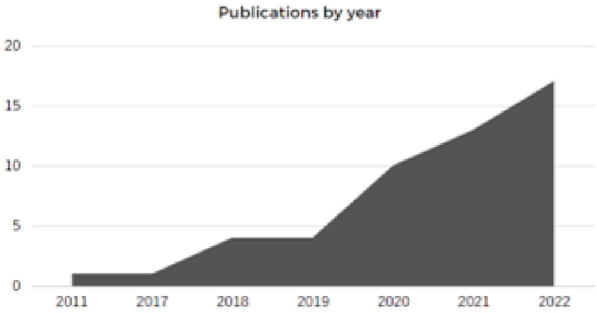


Figure 1.2: Number of detector articles per year

With all the attention, this should continue to increase. The methodologies, also, have changed, from a focus on traditional models to deep learning models, to a DL variant called transformers. These are the newer, more popular ways to both generate text and classify them.

About BERT Transformers

This architecture type is quite new, with the paper [vaswani2017attention] "attention is all you need" introducing its implementation in 2017. For years sequential input modeling was done by Recurrent Neural Networks(RNN), inclusive to language IO and sequences of picture frames for processing video. Shortly after the introduction of transformers, NLP adopted this architecture for its de facto best model for word sequences.

Transformers were then used for language translation, summarization, labeling, captioning and many others. One of the earliest implementations was GPT-1 or Generative Pre-trained Transformer. In years to come this model would evolve to become GPT-2, GPT-3, and finally Chat-GPT(GPT-3.5). Though the actual Chat-GPT model is not publicly available, it has an API to access it. At this time, as well, the GPT-4 API has released, which promises to do even better than Chat-GPT.

BERT is another model which adopted this architecture as a transformer. It is available for public research and very popular amongst academics for researching deep learning. BERT has pre-trained weights which were created from a large body of datasets. The whole reason BERT is so popular is both its architecture effectiveness and pre-training weights. Any researcher can take this existing model and “fine-tune” it for a task. This is what was done here. The modern implementation of transformers is being used by an updated version of the original BERT transformer called, RoBERTa.

RoBERTa Usage in this text

The difference between RoBERTa and BERT, is more data into training the model, some architecture changes, a longer word token input capability and as a result more generalization than BERT. And, since the Chat-GPT model itself is not available for research, we instead make due with RoBERTa.

With this transformer, the experiments were a “fine-tuning” of RoBERTa to classify machine generated text as human or synthetic. Then, next, try to trick the classifier to detect a text as human when it is fake. We found RoBERTa was very simple to trick and the entire purpose of this thesis was to discover and propose how we would account for how detectors fail.

Synthetic Text Detection: Systemic Literature Review

Within the text analysis and processing fields, generated text attacks have been made easier to create than ever before. To combat these attacks open sourcing models/APIs and datasets have become a major trend to create automated detection algorithms in defense of authenticity. For this purpose, synthetic text detection has become an increasingly viable topic of research. This review is written for the purpose of creating a snapshot of the state of current literature and easing the barrier to entry for future authors. Towards that goal, we identified research trends and challenges in this field.

2.2 Introduction

Studies regarding text generation before 2017 were generally scarce and far between. As the body of research grew for synthetic text generation, so did the research for detectors, though lagging behind. This paper discusses current trends of research and future viable research options with the goal of shortening the research process for Artificial Intelligence generated text detection. The main topic of discussion is the literature itself using the PRISMA methodology. Detection literature was reviewed systematically and put together in detail for novel research.

2.2.1 Related Surveys

There are seven surveys near the topic of synthetic text detection. Seven Fatima et al. [2022], Alsmadi et al. [2022], Jawahar et al. [2020], Dong et al. [2021], Li et al. [2021], Iqbal and Qureshi [2022], Celikyilmaz et al. [2020] are about reviewing the literature on the generation process whereby current techniques, domains, data sets and models are shown and reviewed as available. Those seven surveys are very useful for discovering where the body of research is today in regards to text generation.

In the current time very few detection based reviews and surveys exist, with only Jawahar et al. [2020] truly narrowing itself down to actual detection. This survey on binary classification is a valuable contribution to the state of detecting fake text. This article shows techniques and models commonly used in 2020 and since then there have been more relevant publications than ever before.

2.2.2 Main contributions

At the time of the previous surveys/reviews the body of research was perhaps too small for a worthwhile review of primary sources. Since then many of the techniques, models and data sets have become more accessible and as a defense against attack, open-source. Now in the year 2022 we can update and add to these related surveys. For this literature review we have these main contributions:

- A review of 50 related articles about synthetic text detection

- Shows recent innovations for detection.
- Shows gaps in current research for future work.

This is perhaps one of the first literature reviews on the narrow topic of generated text detection. To the best of our knowledge there have been no systemic literature reviews on detecting synthetic text. This study focuses on exploring the current research literature, showing the current ecosystem behind synthetic detection and preparing for future research.

2.3 Research Design

For our systematic literature review, we used current research tools to aid in following the systemic process, PRISMA. The research design here is to find and compile the most relevant body of research for distinguishing fake text and making inquiries. We setup 3 research questions, followed the review process and distilled research to include in the literature review. There were stages of collecting the articles involved, starting with collecting many articles by title then following an exclusion/inclusion process down to 50 papers.

For the actual searching itself the main search engine used was Google Scholar as it is an aggregate of other databases and engines. The publisher, article type and year was recorded and notated in a 3rd party app, Mendeley. 1,211 related articles were chosen from their titles on Google Scholar using specified keywords, approved by a supervising author.

The articles were scrapped from the Google Scholar website using Mendeley's browser extension scrapper and were automatically added and kept in a database of Mendeley's new reference manager to speed up the process. The collections feature of the application was used to separate the stages of the PRISMA methodology.

2.3.1 Research Questions

With the unifying goal of preparing for future research, the inclusion/exclusion process included more and more study of the given texts. These research questions and research objectives were created to guide the process of choosing articles and scrutinizing the literature for that purpose:

Table 1: Research Questions

Research Question	
RQ1	Which datasets are currently used in the literature to detect deep fake models?
RQ2	What accuracy evaluation methods are there for detection effectiveness?
RQ3	What impact have recent innovations had on fake text detection?

2.3.2 Research Objectives

For this study there are 5 objectives:

- To investigate the current existing techniques/approaches of detecting artificial text.
- To explore models and datasets created to detect artificial text.
- To explore accuracy evaluation of fake text detection.
- Show recent innovations since previous surveys.
- Show future work for further research.

2.3.3 Searching Strategy To Retrieve Studies

Given most to all studies were queried on Google Scholar with keywords including text generation, detection and synthetic text, the engine includes searches into various other databases such as IEEE, ArXiv, Semantic Scholar, Springer, ACM journals, Elsevier and more, even including schools inside one search page. Keywords were used in regards to their category. Specifically in this SLR each set of keywords contributed a number of articles but some were more valuable than others. "Text generation detection" was perhaps the most fruitful, though, the body of research is quite small. A wide variety of query keywords had to be used to gather the largest possible pool of articles regarding text generation. For a few cases the title did not appear to be about synthetic detection, though upon further reading was in fact relevant to detection and vice versa.

Table 2: Keywords used by category

Domain	Text generation method	Sample size	Text generation innovations	Classifier
		Large	Natural Language Processing	
Social Networks	GANs	Small	Text analysis	Word embedding
Fake news	Fake text	Sample	Text classification	CNN
Domain	Augmentation	Training		RNN
Low resource		Models		Transformers
				Detection
				LSTM
				Ensamble

*A note; while adapting this paper for the final thesis draft, now that ChatGPT received publicity, that is now a hot keyword, though not included here.

From these keywords they were assorted with AND, OR and quotation required clauses. Certain key words like "text generation" AND "language processing" were especially effective together at finding articles while "fake text" led to irrelevant topics. Though some keywords were only useful for finding niche articles.

2.3.4 Article Inclusion Exclusion Criteria

A total of 1,211 articles were found using the above queries. With duplicates removed, the remaining 1,041 articles were sifted. Many articles containing relevant keywords and titles were not about text generation, Natural Language Processing (NLP) and detection. Some articles were not machine centric and were about societal or human reading of generated text though they were about generated text detection. Others mentioned fake text as trolling, which is not the focus of this review.

In the partial review many of these papers were excluded by abstract because they were not machine centric or were based on societal differences. Out of the 1,041 articles after removing duplicates, a partial review left 381 articles eligible for full review. The following criteria was used to include or exclude these papers from this point on:

The following is the inclusion criteria:

- The article must include machine generated text classification or be highly relevant
- The article can include other languages in its dataset
- The article itself must be written in English
- Surveys on text generation are allowed

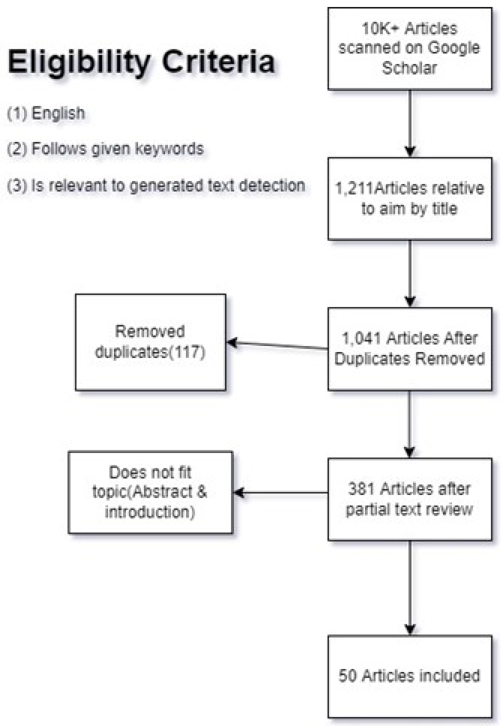
The following is the exclusion criteria:

- The models used must be machine-centric, meaning they require machine learning to determine if a sample is generated text.

2.4 Systematic Mapping Study Results

Here we show the results of the systematic study, stages of the inclusion/exclusion process, publisher names and dates of publication. This overviews the current status of synthetic text detection literature in late 2022 and records potential research gaps to be filled by future authors.

Figure 2.3 Inclusion exclusion process


Eligibility Criteria
(1) English
(2) Follows given keywords
(3) Is relevant to generated text detection
10K+ Articles scanned on Google Scholar
1,211Articles relative to aim by title
Removed duplicates(117)
1,041 Articles After Duplicates Removed
Does not fit topic(Abstract & introduction)
381 Articles after partial text review
50 Articles included

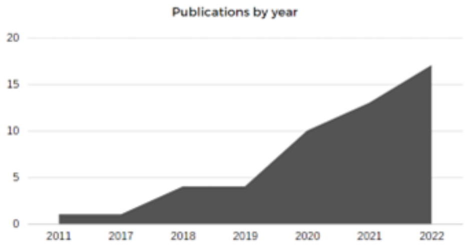


Figure 2.4: Publications by year

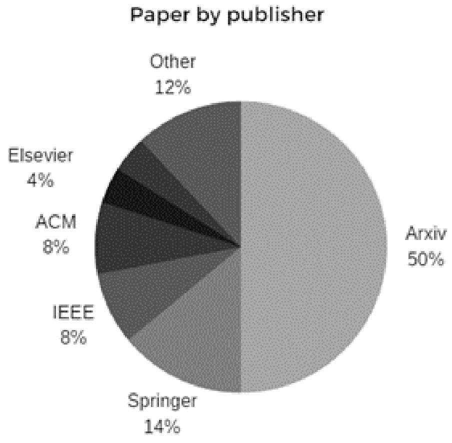


Figure 2.5: Articles by publisher

RQ1: *Which datasets/models are currently used in the literature to detect generated text?*

According to literature, the more training data for both human and machine generated text the better the outcome will become. Though there are many sources for both real and fake it is good to see the popular ones for specific domains. Below are datasets usable for training a model for actual detection and research in the time to come:

Open-source Datasets

- Hugging face: `https://huggingface.co/datasets`
  This website is the first place to look for datasets and models. Hugging

face has a plethora to choose from across many regards of machine learning. Most to all of those below can be found on the platform.

- GPT-3, GPT-3.5, GPT-4: https://openai.com/api/
  This is a high quality source of generated text. There are several models to choose from for GPT-3 though the models are not free to use.

- GPT-2: https://github.com/openai/gpt-2-output-dataset/
  For a while this dataset was standard in its use for text generation. This dataset includes samples of both synthetic and real text.

- Grover: https://github.com/rowanz/grover
  This is more of a collection of scripts for making a dataset oriented for news articles. The repo includes a detection model, text generator, accuracy evaluator and a web crawler for gathering authentic text source data.

- Authorship Attribution: https://bit.ly/3DNlLxw
  This is a dataset for detecting specific popular text generators. The csv samples for these generators are placed in the above link. The focus is more based on news/political articles.

- TuringBench: https://github.com/TuringBench/TuringBench
  The main website for TuringBench is more of a leaderboard about detecting which generator is being used. The dataset is given in a zip file and whoever gets the highest accuracy rating for detecting the generator wins.

- Academia papers: https://github.com/vijini/GeneratedTextDetection/tree/main/Dataset
  A niche dataset for synthetic academia papers, though, it is small and is not condensed in one file. This would be good to expand upon as a separate research paper.

- TweepFake: https://github.com/tizfa/tweepfake_deepfake_text_detection/
  A popular twitter dataset with human and machine tweets

Open source generative models

- Grover: https://grover.allenai.org/

- GPT-2: https://github.com/openai/gpt-2

- GPT-3 group(Paid): https://openai.com/api/

- Hugging face: https://bit.ly/3LwGszE
  I mention this again, with over 5,000 models to choose from, you can see the limitations of text generation, GPT-2 being the most popular on the platform.

- Web app w/ text generation (GPT-2), (Grover): https://app.inferkit.com/demo

Existing detective models

In late 2022 detectors usually are rarer than datasets, have to be trained with a generated text dataset, and are not often pre-built. Though new models and prebuilt detectors are being created all the time and now popping up more rapidly. There are likely many on the HuggingFace platform for example. Some models also serve a dual usage, having both detector and generator.

Below, an example of a pre-trained model, BERT, is good for general text classification modeling, requiring further building to fully classify generated text as real or fake. GLTR is another example model, a human detection helper which improves human-centric detection. GLTR colors words which are most suspiciously generated, boosting accuracy quite a bit with minimal learning from the person. The rest below are pre-built and available for detection testing:

- BERT based modeling
- GLTR: `gltr.io`
- Grover: `https://grover.allenai.org/`(2019)
- Open-AI GPT-2: `https://huggingface.co/openai-detector/` (2019)
- RoFT: `https://roft.io/` (gamification of human detection)

RQ2: *What accuracy evaluation methods are there for detection effectiveness?*

According to Li et al. [2022] a good general rule of model evaluation is testing the mislabel or error rate. This can mean testing against a given test/validation set or testing against an outside dataset. Using a recorded error rate you can also distinguish how effective a detector is per generative model.

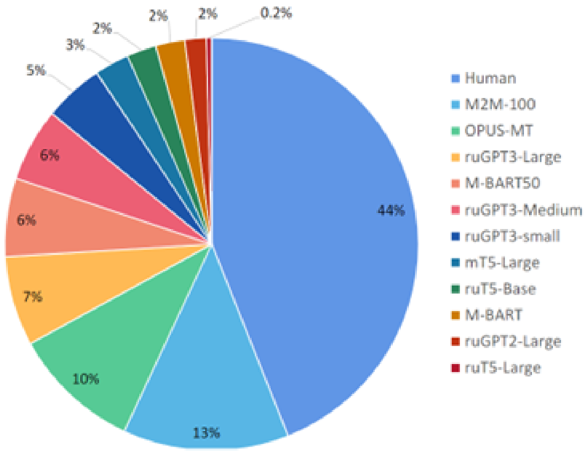
The standard way most detectors are evaluated in the literature is to stay in the same domain and use similar datasets. The error rates are based on a narrow binary classification and due to this there are much better error rates. You can see this throughout the literature where there is one domain, like TweepFakeMaurizio [2021], news oriented models Zellers et al. [2019], language based models Chen et al. [2022], and other niche categories sticking to their own evaluated domains.

To truly test a model against more real data here are two things done in the literature; adversarial testing Wolff [2020] and sourcing different generators Li et al. [2022]. For adversarial testing this means accounting for a post-processing phase of text generation whereby Greek and uncommon symbols are added along with misspellings and surely other techniques in adversarial fashion to cause the detector to mislabel the text as human written. Testing against this noise is a very good way to evaluate a real-world model. A typical solution for adversarial attack is with a preprocessing phase for the detector.

For evaluating accuracy against different sources, a good idea from Li et al. [2022] is to record the error rates of different common and popular generator. Below is an example of how a detector can mislabel a text,

The authors of the article found human text was most mislabeled as synthetic. This can be used to adjust a model and evaluate its accuracy and can further be divided by which sources the human written text originated. Though a major flaw later stated by the author is how common it is to fine tune a generator, making labelling all generators unfeasible.

Figure 2.6 Example evaluation based on source, **?**



Lastly, how a model will be used should always be part of the evaluation. Low resource training may be better for something like TweepFake, captioning or microblog detection. Though general language detection would require more resources and vary in effectiveness based on the text in question plus how similar the training set had been. So the evaluation can be divided up in some cases or narrowed in others. This way the evaluation is accurate to the model's purpose.
RQ3: *What impact have recent innovations had on fake text detection?*

Across the literature the general setting of recent innovations is to combat fake text, detecting and removing its effect on public discourse. Paid models like GPT-3 are generally superior at creating advanced generated text for fooling human detection but is made relatively open source for research against itself and other high quality generators.

There is a trend to open source datasets and models to protect the community from synthetic text by making it easier to create both generation and detection models. These motivations perhaps have had the greatest impact on future research. Many articles, as seen in Figure 2; publications by year, have been written more recently. Synthetic text will likely be more highly researched in the next few years (Since writing, this has and will become more true).

Particularly niche and low resource domains are being filled in with working models and novel solutions. Short and long form AI generated document detection models are now more numerous with their accuracy reported in their respective papers. These new models give us more options than the standard detector, opening up more difficult and niche types.

2.5 Identified research gaps

Here we discuss the limitations of current research. The gaps are separated by 5 aspects of synthetic text detection gathered from our research questions. From these gaps future research can be studied.

1. **Limited overall research.** As of mid 2022 there is a scarcity of research

papers regarding artificial text detection. The majority of time spent creating this SLR was in curating as many articles as possible and even with that time spent there were still a relatively few papers.

2. **Limited research on adversarial attacks.** Pre and post processing methodologies are missing for attacking detectors. Here is one example on adversarial attackingWolff [2020].

3. **Limited evaluation methodologies for detectors.** Several papers existed for evaluation methods though nothing thorough. For RQ2 the information was pulled from small parts of a group of papers but there was not much research outside of that.

4. **Low resource detector optimization.** Low resource training also had limited research. TweepFakeMaurizio [2021] and fake academic paper detectionLiyanage et al. [2022] were perhaps the most optimization related articles. There is a gap here.

5. **Research in other languages.** Most other languages have very limited research though some articles do existChen et al. [2022], S et al. [2022], Shamardina et al. [2022]. Data sets exist for Chinese, Russian and other languages as well but very few synthetic language detectors outside English.

2.6 Recommendations and future research directions

There is plenty of room for research in AI text detection across different aspects. Given the limited overall research the field can be taken from many angles. This includes studies on increasing accuracy for specific domains such as news, blogs, social media outlets, books, academia. Most to all domains are open for detection modeling.

Language specific research is a definite direction a person can take. Spanish, Hindu and many others do not have synthetic text detection research. Remaking previous research paper detectors in different languages is a good bet as well as dataset creation for future authors. There is little to no research on low resource generated text detection languages and domains as well.

Better, more robust, evaluation methods for detectors is a potential topic. Tasks like generalized accuracy tests or post/pre processing methodologies are open game. In this vein of evaluation adversarial detector attacks are a great and open avenue for research. In Wolff [2020] adding simple homoglyphs break most detectors and can easily be added to generation models. Misspelling also helps in adversarial text generation. Together the detector recall goes from 97% down to 0.26%, massively fooling detection

In addition there are many niche domains where text generation and even some detectors exist but there is no adversarial research behind the domain. A paper like Le et al. [2020] takes an adversarial approach to generating synthetic comments to fool detectors. To this end only a handful of research was found.

Another open but more difficult research niche is authorship attribution. Not just binary detection but multi-label detection for the most popular models. An example of this research is previously mentioned TuringBench, whereby an online leaderboard was created to detect which model the generator is sourced.

2.7 Summary

Natural language processing is trending in good fashion with plenty of open source projects and ideas for novelty. Automated AI text generation has grown tremendously in the past four years and now there is a fresh need for detection. As we enter a phase of defending trolls, bots and generated commentary recent advancements have allowed for more of itself. In this paper, our focus is on synthetic text generation with possible research trends and challenges.

A Mutation-based Text Generation for Adversarial Machine Learning Applications

Many natural language related applications involve text generation, created by humans or machines. While in many of those applications machines support humans, yet in few others, (e.g. adversarial machine learning, social bots and trolls) machines try to impersonate humans. In this scope, we proposed and evaluated several mutation-based text generation approaches. Unlike machine-based generated text, mutation-based generated text needs human text samples as inputs. We showed examples of mutation operators but this work can be extended in many aspects such as proposing new text-based mutation operators based on the nature of the application.

3.1 Introduction

Currently, text generation is widely used in Machine Learning (ML)-based or Artificial Intelligence (AI)-based natural language applications such as language to language translation, document summary, headline or abstract generation. Those applications can be classified into different categories. In one classification, they can be divided into short versus long text generation applications. Short text generation applications include examples such as predicting next word or statement, image caption generation, short language translation, and documents summarization. Long text generation applications include long text story completion, review generation, language translation, poetry generation, and question answering.

Large language models such as open AI chat GPT-1,2 and 3 can be used to masquerade humans in many of those short and long text generation applications. Unlike all previous applications where ML or AI-based text generation exists to support humans and their applications, in online social networks (OSN) such as Twitter and Facebook, text generation is used to fool and influence humans and public opinions through masquerading machine-generated texts as genuinely generated by humans. Social bots and trolls in OSNs are accounts that impersonate humans. They are controlled by ML or AI algorithms. ML algorithms are used to auto generate text in those bot/troll accounts.

As mentioned earlier, ML-based text generation can be used for non-malicious and malicious applications. Our focus in this paper is in malicious applications. There are several malicious ML-based approaches to text generation that can be observed in literature such as:

- Classical (e.g. professor/teacher forcing recurrent neural network, RNN): It aims to align generative behavior as closely as possible with teacher-forced behavior, Lamb et al. [2016].

- Conventional (e.g. based on hidden Markov models (Jing [2002]), method of moments (MoM), Jones [1969], Restricted Boltzmann machine (RBM), Holyoak [1987]).

- Cooperative training, (e.g. Yin et al. [2020], Lu et al. [2019]).
- Reinforcement based and Reinforcement free approaches (e.g. RankGAN, Lin et al. [2017], MaliGAN, Che et al. [2017], FMGAN, Chen et al. [2018], and GSGAN, Kusner and Hernández-Lobato [2016]).

In one classification to adversarial machine learning (AML) attacks, those attacks can be divided based on at which machine learning stage the attack is occurring (e.g. (1) learning/training stage, versus (2) testing stage).

3.1.1 Poisoning attacks: AML attacks on the learning stage: Manipulating the training data

Attackers can deliberately influence the training dataset to manipulate the results of a predictive model. A poisoning attack adds poisoned instances to the training set and introduces new errors into the model. If we consider one ML application, spam detection, filter of spam messages will be trained with adversary instances to incorrectly classify the spam messages as good messages leading to compromising of system's integrity. Alternatively, the Spam messages' classifier will be trained inappropriately to block the genuine messages thereby compromising system's availability, Newsome et al. [2006], Perdisci et al. [2006], Nelson et al. [2008], Rubinstein et al. [2009], Barreno et al. [2010], Biggio et al. [2012], Newell et al. [2014], Jagielski et al. [2018].

3.1.2 Evasion attacks: AML attacks on the testing stage: Manipulating the testing data

In this attack, attackers try to evade the detection system by manipulating the testing data, resulting in a wrong model classification. The core of adversarial evasion attack is that when an attacker can fool a classifier to think that a malicious input is actually benign, they can render a machine learning-based malware detector or intrusion detector ineffective, Russu et al. [2016], Zhang et al. [2020], Sayghe et al. [2020], Shu et al. [2022]. Chernikova and Oprea [2022]

Our mutation-based approach in this paper can fall under the first category of the last classification (i.e. poisoning attacks). A mutation instance is a data instance or object that is created from an original genuine instance with a slight change. Such changes on machine learning stage can reduce accuracy of classification algorithms through generating fake instances that are very close to genuine instances. Mutation changes are typically introduced to simulate actual real-world faults or mistakes. There are several scenarios to implement our mutation-based AML:

- Two class labels (Human/Mutation text): Human generated text versus mutation generated text. Such experiments will test classifiers sensitivity to changes injected by mutations in comparison with original genuine text.
- Two class labels (Human/Adversarial, mutation instances are added to adversarial instances

- Two class labels (Human/Adversarial, mutation instances are added to Human instances
- Three class labels (Human/Mutation/Adversarial instances).

In this paper, we will propose and evaluate mutation operators within the first category and leave investigating other categories in future papers.

The rest of the paper is organized as follows. Section 3.2 provides a summary of related research. Our paper goals and approaches are introduced in 3.3. Section 3.4 covers the experiments we performed to evaluate our proposed mutation operators. We have then a separate section, section 3.5 to compare with close contributions. Finally, Section 3.6 provides some concluding remarks as well as future extensions or directions.

3.2 Related Works

Machine learning NLP-based classifiers can be influenced by words misspelling and all forms of adversarial text perturbations.

Literature survey indicated an increasing trend in using pre-trained models in machine learning. Word/sentence embedding models and transformers are examples of those pre-trained models. Adversarial models may utilize same or similar pre-trained models as well. In another trend related to text generation models, literature showed effort to develop universal text perturbations to be used in both black and white-box attack settings, Alsmadi et al. [2022]. The literature on adversarial text analysis is quite rich (e.g. see Bravo-Santos et al. [2020]). Our focus is in a selection of those papers on adversarial text generation relevant to poisoning attacks in general or mutation-based approaches in particular.

Word swapping (of semantically equivalent words) attack, Alzantot et al. [2018] is similar to one example of our mutation operators where one word is swapped from original to mutation instances. In Alzantot et al. [2018], Laughlin et al. [2019], Sayghe et al. [2020] and Shi et al. [2022], words are swapped based on genetic algorithms. Genetic algorithms typically include 5 tasks in which mutation is one of them (Initial population, Fitness function, Selection, Crossover, and Mutation). As alternatives to genetic algorithms, word swapping attacks are also implemented using (1) neural machine translation in , Ribeiro et al. [2018], (2) swarm optimization-based search algorithm, Shi et al. [2022]. In Alzantot et al. [2018], mutation instances are used to fool a sentiment analysis classifier.

Words substitution can be also role-based lexical substitutions, Iyyer et al. [2018], or entity-based text perturbation, Liu and Chen [2022].

In Bhalerao et al. [2022] authors classified intentional and unintentional adversarial text perturbation into ten types, shown in Table 3.1.

As we mentioned earlier, this work is closely related to AML poisoning attacks, Newsome et al. [2006], Perdisci et al. [2006], Nelson et al. [2008], Rubinstein et al. [2009], Barreno et al. [2010], Biggio et al. [2012], Newell et al. [2014], Jagielski et al. [2018]. It is also related to AML text perturbations, Vijayaraghavan and Roy [2019], Eger et al. [2019], Gao et al. [2018], and Li et al. [2018].

Perturbation Type	Defense	Example	Definition
Combined Unicode	ACD	P.l.e.a.s.e.l.i.k.e.a.n.d. s.h.a.r.e	Insert a Unicode character between each original character.
Fake punctuation	CW2V	Pleas.e lik,e abd shar!e the v!deo	Randomly add zero or more punctuation marks between characters.
Neighboring key	CW2V	Plwase lime and sharr the vvideo	Replace character with keyboard-adjacent characters.
Random spaces	CW2V	Pl ease lik e and sha re th e video	Randomly insert zero or more spaces between characters.
Replace Unicode	UC	Pleãse lîke and sharê the video	Replace characters with Unicode look-alikes
Space separation	ACD	Please l i k e and s h a r e	Place spaces between characters.
Tandem character obfuscation	UC	PLE/\SE LIKE /\ND SH/\RE	Replace individual characters with characters that together look original
Transposition	CW2V	Please like adn sahre	Swap adjacent characters
Vowel repetition and deletion	CW2V	Pls likee nd sharee	Repeat or delete vowels.
Zero-width space separation	ACD	Please like and share the video	Place zero-width spaces (Unicode character 200c) between characters

Table 3.1: Adversarial text perturbations, Bhalerao et al. [2022]

3.3 Goals and Approaches

According to a previous paper Wolff [2020], a typical RoBERTa-based classifier mislabels synthetic text to human by very basic differences such as changing 'a's to alpha or 'e's to epsilon. This vulnerability can be used to trick detectors of synthetic text either intentionally or accidentally.

To compare synthetic text detectors sensitivity to mistakes or changes to human text we can break up these mutations into operators with the goal of supporting the creation of more generalized synthetic text detectors. Here, these operators will be introduced and be used to fine-tune RoBERTa's pre-trained model to detect mutations, such as the first scenario mentioned earlier, section 3.1.2.

3.3.1 Approach: Use a finite set of operators for research customization

We introduce some examples of mutation operators to implement are in Table 3.2. There are more advanced operators which can be used for attacking a detector on a more granular level in the future. For our research here however, we will be using more basic mutation operators such as these:

These 7 operators can replicate simple mistakes and changes which can happen to human written text, including the 2 operators used in previous cited works. These were chosen for their ease of implementation and usage in previous research. We will leave more advanced and numerous operators for future papers.

As for the implementation, most of these operators use word maps which iterate through each string replacing the words with the intended character, word insertion or deletion. Punctuation and excess special characters are removed for simplicity. Though there is different methods for the different mutation operators.

The random word operator is in fact a list of random words. Arbitrary words are chosen and replaced with a random word from the same list. Limits to the number of mutations should be added to limit the operator from completely fuzzing the string as well. The code from our GitHub, `https://github.com/JesseGuerrero/Mutation-Based-Text-Detection` has some written operators

Mutation Operator	Example	Definition
Randomization	Plz shr and hate film	Use all below mutation operators
Misspelling words	Plz sharr and like the vid	Misspell a few words
Deleting articles	Please share and like video	Delete a few articles, including starting ones
Random word with random word	Please roar and tree video	Replace a random word with another random word
Synonym replacement	Please disseminate and prefer the video	Replace a word with its synonym
Antonym replacement	Please hide and hate the video	Replace a word with its antonym
Replace ”a”, ”e”	Pls lik nd shar the video	Replace some a’s and e’s with epsilon & alpha

Table 3.2: Experimental mutation operators

which can be viewed as examples.

In our implementation word maps are limited to 3000 of the most common words, synonyms and antonyms. These words can be pulled from any API service such as RapidAPI to get lists of words, synonyms, adverbs, verbs, etc.

3.3.2 Approach: Test mutations by Evasion attacks on neural network detectors

We will be testing these 7 operators against RoBERTa pre-trained models. Of these 7 operators we want to test how they will affect a previously researched synthetic text detector, how a fine-tuned mutation text detector will be affected and as well as see the differences between the different mutation operators. Lastly we want to see how shorter text affects these results. More details on the next section, 3.4.

3.4 Experiments and Analysis

With these mutations, we can introduce mutation detection with a classic binary RoBERTa classifier. This part of the experiment is the extension portion of the previous author’s work mentioned before with an actual solution to the vulnerabilities in that paper.

3.4.1 Experiment methodology

We used an existing RoBERTa classifier which is meant to classify synthetic and human generated text to test how it would classify mutated text. It is still the pre-trained binary model, however it is being retrained to detect mutation rather than synthetic text.

The data set used was the full COCO images data set where hand written captions are placed for each image. A total of 5 captions are human created per image. The captions were parsed into a re-usable format and were used to train the human portions of the model.

Across the training, testing and validation sets there were over 700,000 human texts used to train the model. Two models were made from this data set. The first was based on individual captions. The second was based on these 5 captions combined per unique image name for calling via a map.

Six operators were used as mutations for this classifier. They are; (1) replacing synonyms, (2) antonyms, (3) random words, (4) removing articles, (5) replacing a with alpha and e with epsilon, and lastly, (6) the most common misspellings.

The training data was duplicated for the mutation data sets. Over 700,000 texts were used for training and the same texts were re-used for mutations. This meant for individual captions there were over 1.4 million text instances with both mutation and human labels. For combined captions there were 1/5 of the total instances.

The data sets were selected as they were already labelled from COCO dataset. The training set went to training, the validation set went to validation and testing to testing. For training the mutation label, the mutation operators were used at run-time.

The operator was randomly chosen at run-time with a simple random function among 6 operators. Each operator was used so the classifier can learn to detect all 6 of these mutation types in one classifier.

So far as testing is concerned, the same testing data set from COCO was re-used with an operator manipulating a whole set. A total of 7 testing data sets were created for each of the 6 operators and a seventh randomized data set, like the mutations at run-time. This formed our metrics of how accurately the model can correctly label each instance of the mutation data sets as mutations and how accurately the model can label human text. In a total 8 operators, 1 human and a seventh randomizing the first six mutations.

3.4.2 Preliminary Results

If we were to apply these operators to the previously researched synthetic text we should get poor results for detecting mutated text. Given text derived from human text, though just modified, is still synthetic, we can see that mutation poses a vulnerability to detecting machine and human generated texts. Here are the results:

Operator Type	**Accuracy**
None	∼88.80%(1000 samples)
Randomized	∼01.00%(1000 samples)
Replace Alpha, Epsilon	∼01.01%(1000 samples)
Misspelling words	∼00.00%(1000 samples)
Delete articles	∼01.60%(1000 samples)
Synonym replacement	∼00.00%(1000 samples)
Replace random word	∼07.79%(1000 samples)
Antonym replacement	∼09.89%(1000 samples)

Table 3.3: Preliminary Results

As we can see from 1000 samples the accuracy is quite poor when modifying the text. The original detector without mutations had a recall of over 97% detection of synthetic and human text in-distribution. For our research outside of the paper we have an out of distribution pure human text data set as 88% accurate as the 1st row in the table.

This means the detector is quite good at classifying human text out of distribution and is even good at detecting in-distribution synthetic text. The model does those things above human distinction which in the past was around 54% accurate. Our issue from our modeling is mutation from which the model does not perform well.

3.4.3 Experiments results & analysis

So far as the first run through with individual captions, the results were pretty good. A total of 2,490 texts were tested for the detector from the testing data set. In total overall the detector accuracy was about 91% and each epoch took 13 hours for a total of 4 epochs or 52 hours total.

Operator Type	Accuracy
None	∼71.48%(2490)
Randomized	∼99.83%(2490)
Replace alpha, epsilon	∼99.95%(2490)
Misspelling words	∼99.95%(2490)
Delete articles	∼59.87%(2490)
Synonym replacement	∼99.91%(2490)
Replace random word	∼100%(2490)
Antonym replacement	∼99.03%(2490)

Table 3.4: Individual captions, short language modeling

*For the live thesis defense a different model was trained as the old was lost. The live results differ by 10% for overall accuracy.

The most inaccurate operator overall was always the "delete articles operator". This can be due to some semantic issues or just the difficulty of detecting what is *not* there rather than what *is there* to a RoBERTa classifier. For this first run, the other operators ranged from 59% accurate to 100% accurate, with human detection being 71%.

For the second run the captions for text chunks were combined per image. This meant all 5 captions were now one text and were fed to the training model. This means 1/5 th of the instances but more per text. The results were a slightly lower; total overall detector accuracy of 88% with 2490 texts being tested.

The epochs took about 2 hours each this way as well. A total of 10 epochs or 20 hours were used to finish the model training. Same as before, the delete articles operator was the weakest mutator and was in fact even weaker the second time.

Besides the weakest operators, the other operators were all above 95%, much better than the original neural network. If we were to remove the lowest performing operator we would in fact have 95% accuracy for the 1st run and 97%

Operator Type	Accuracy
None	∼93.65%(2490)
Randomized	∼98.96%(2490)
Replace alpha, epsilon	∼99.92%(2490)
Misspelling words	∼99.80%(2490)
Delete articles	∼25.42%(2490)
Synonym replacement	∼99.76%(2490)
Replace random word	∼98.43%(2490)
Antonym replacement	∼92.37%(2490)

Table 3.5: Combined captions, longer language modeling

accuracy for the 2nd. This means in reality the combined text may be the better approach to train.

3.4.4 Experiment: Models issues and weaknesses

The main issue of the evaluated models is possible a bias issue. It seems that models work mostly for semi in-distribution data sets. So accuracy is altered quite a bit by outside data sets. Both the individually captioned model and the grouped model were tested in an out of distribution data set. The individually captioned data set was 2.2% accurate for human detection and 100% accurate for Alpha/Epsilon mutation, while the grouped captioned data set was 55.7% accurate for human detection and 100% for Alpha/Epsilon mutation. The issue appears to be the out of distribution set may have had vocabulary that didn't exist in distribution and the detector defaults to the mutation label. Still in this sense, the longer the text-set the better is the performance.

In the future we can use this work flow with more diverse training data sets to generalize models to out of distribution and use these models to prevent simple mutation from fooling binary detectors. Lastly, the mutation operator "delete articles" seems like a great way to fool the classifier into mislabeling mutated text.

3.5 A Comparison Study

3.5.1 Machine Text Generation

Machine text generation is a field of study in Natural Language Processing (NLP) that combines computational linguistics and artificial intelligence that has progressed significantly in recent years. Neural network approaches are widely used for this task and keep dominant in the field. The state-of-the-art methods may include Transformers Vaswani et al. [2017], BERT Devlin et al. [2018], GPT-3 Brown et al. [2020], RoBERTa Liu et al. [2019], etc. The models are trained on a large amount of text data. For example, the GPT-2 model was trained on text scrapped from eight million web pages Radford et al. [2019], and is able to generate human-like text. Due to the high text generation performance, such methods are very popular on tasks, such as image caption generation, text summarization, machine translation, moving script-writing, and poetry composition. However, the output of such methods is often open-ended.

Different papers show promising results regarding transfomers, ensemble learning and RNNs. Works like Li et al. [2022], Tourille et al. [2022] and Wolff [2020] include the ability to detect GPT models very accurately. In Li et al. [2022] for example, the ensemble model is able to detect which model is being used to generate text with 66% accuracy, not just what is synthetic and what is real. Author attribution is difficult, so this is a relatively high number.

In Tourille et al. [2022] transformers are used to detect generated tweets from different models with an above 90% accuracy. Similar results were found in Najee-Ullah et al. [2022] with an above 95% accuracy. It is common to find papers based on transformers with such great results.

There are perhaps a few dozen papers all showing amazing accuracy for transformers. However there are much less on adversarial attacks and only 1 or 2 on mutating generated texts for detection. This is why this paper was written, to contribute to this topic which is scarce in research.

Through this work, we propose a mutation-based text generation method that can be distinguished from the existing text generation method fundamentally. Unlike the neural network based methods, the mutation-based method generates output based on the given text under a given condition. The text is generated in a tightly controlled environment, and the output is closed-ended. The well-controlled environment makes the output of the mutation-based method suitable for serious security test tasks, such as machine learning model vulnerability tests and SQL injection defense and detection.

Given a text corpus (e.g., a sentence or a paragraph), $\mathcal{T}$, which contains an ordered set of words, $\mathcal{W} = \{w_1, w_2, ..., w_n\}$, and an ordered set of punctuation, $\mathcal{P} = \{p_1, p_2, ..., p_m\}$, a mutation operator, $\mu(\cdot)$ is used to generate the mutation-based text. For instance, given a character-level mutation operator, $\mu_c(\cdot)$:

$$\mathcal{W}' = \mu_c(w_i, \rho, \sigma), \tag{3.1}$$

where $\mathcal{W}'$ is the output of $\mu_c(\cdot)$, which replaces the letter ρ in w_i ($w_i \in \mathcal{W}$) with a mutation σ. Then, the final output of $\mathcal{T}$ is $\mathcal{T}' = \langle \mathcal{W}', \mathcal{P} \rangle$. For instance, assume $\mathcal{T} =$ `"Text generation is interesting!"`, $\mathcal{W} = \{\texttt{Text}, \texttt{generation}, \texttt{is}, \texttt{interesting}\}$, and $\mathcal{P} = \{\texttt{!}\}$, $w_i = \texttt{generation}$, and a character-level mutation operator $\mu_u(\cdot)$, where $\rho = \texttt{a}$ and $\sigma = \alpha$ (the Greek letter alpha). Then, the output text corpus, $\mathcal{T}'$ is generated as:

$$\begin{aligned} \mathcal{T}' &= \langle \mathcal{W}', \mathcal{P} \rangle \\ &= \langle \mu_c(w_i, \rho, \sigma), \mathcal{P} \rangle \\ &= \langle \mu_c(\texttt{generation}, \texttt{a}, \alpha), \mathcal{P} \rangle \\ &= \texttt{Text gener}\alpha\texttt{tion is interesting!} \end{aligned} \tag{3.2}$$

3.5.2 Mutation Testing in the Language Domain

The proposed mutation-based text generation is inspired by the advances in mutation analysis in software testing and the idea of ”broiling frog syndrome” The well-controlled and close-ended environment makes generating precise text output possible. By using the proposed tool, researchers could test a language model by changing the input slightly and step-by-step.

Charm Bravo-Santos et al. [2020] is a chat-bot testing tool closely related to our mutation-based text generator that extended Botium **?**—a popular framework for chat-bot testing—by integrating eight mutation operators. Though both Charm and ours use mutation operators, Charm is proposed as an extension to Botium and relies on Botium to work. In addition, Charm only works for chat-bot testing.

Unlike Charm, ours is a general-purpose language generation method, which is designed to work alone. Our method can be used to analyze any type of language models that accept a sequence of text as input. In addition, the users are not limited by the pre-defined mutation operators. Our proposed mutation-based text generator is a general framework. The users could easily design their own mutation operators for their specific tasks within our framework.

3.6 Conclusions

Automatic text generation techniques are adopted into various domains, from question-answering to AI-driven education. Due to the progress of neural network (NN) techniques, NN-based approaches dominate the field. Though advanced techniques may be applied to control text generation direction, the text is still generated in a widely open-ended fashion. For instance, a NN-based approach can generate a greeting message to greet a specific person. However, it is hard to control the exact wording used in the message. Such open-ended text generation might work fine for content generation tasks. However, due to lack of precise control, using open-ended text generation methods to systematically evaluate flaws in language analysis models may be non-trivial.

Unlike the existing language models, our proposed mutation-based text-generation framework provides a tightly controlled environment for text generation that extends text-generation techniques to the field of cyber security (i.e., flaws evaluation for language analysis models). The output of our framework can be used to systematically evaluate any machine learning models or software systems that use a sequence of text as input, such as SQL injection detection Hlaing and Khaing [2020] and software debugging Zhao et al. [2022]. Researchers may also design their own mutation operators under our framework.

We demonstrated the proposed text-generation framework using the RoBERTa-based detector that is pre-trained for separating human-written text from synthetic ones. Our experiments showed that the RoBERTa-based detector has a significant flaw. As a detection method, it is extremely vulnerable to simple adversarial attacks, such as replacing the English letter ”a” with the Greek letter ”α” or removing the articles—a, an, the—from a sentence. We also demonstrated that simply including the adversarial samples (i.e., the mutation texts) in the fine-tuning stage of the classifier would significantly improve the model robustness on such types of attack. However, we believe that this issue should be better addressed on the feature level since any changes at the text level will lead to changes in the tokenization stage that will eventually lead to a different embedding vector being fed into the classification network. Thus, one future direction of this work is reducing the distance between the original and mutation samples in the feature space. Some potentially useful methods might include using contrastive learning and siamese network Koch et al. [2015], Liang et al. [2021] as well as dynamic feature alignment Zhang et al. [2022], Dong et al. [2020].

Besides improving the robustness of the RoBERTa-based detector, we plan to continue to work on the development of the proposed mutation-based text generation framework. Currently, the framework only works in a two-step testing scenario. Users need to use the framework to generate the testing cases and feed the testing cases into the downstream model in separate steps. We plan to release a library that can be directly imported into any downstream applications. Tools for easily creating and editing mutation operators will also be created. In addition, a graphical user interface may also be developed.

In conclusion, we propose a general-purpose, mutation-based text generation framework that produces close-ended, precise text. The output of our framework can be used in various downstream applications that take text sequences as input, providing a systematic way to evaluate the robustness of such models. We believe the proposed framework offers a new direction to systematically evaluate language models that will be very useful to those who are seeking insightful analysis of such models.

Conclusion & Summary

Detecting synthetic text is a very viable research avenue for new Machine Learning researchers, students and graduates. Outside of machine-centric articles, the body of research is very saturated. Cultural, societal and psychological works are thoroughly covered in domains like Fake News or the cultural implications of synthetic text.

3.6.1 Summary

In chapter 2 we cover how fake text detection is open to research. We also stated, within Computer Science some fields also are saturated, such as Computer Vision and Computer Networks. However, for machine centric synthetic text detection, as in machine on machine generation and detection, there are plenty of topics to claim as contribution. In chapter 2 you can see what research avenues are most available.

In chapter 3 we use one of these research avenues found in the Systemic Literature Review. Particularly, adversarial attacks on detectors using mutations. As said in the chapter the normal accuracy of these transformer detectors has a 97% recall rate, quite high. By simply adding mutations the accuracy moves below 10%.

This is the question we seek to answer in chapter 3, ”can we defend against mutations?” The answer is yes, we can. The experiment successfully showed we could learn 6 types of mutations and classify them as mutation. Future work from chapter 3 would then become two parts, finding more operators that are either more human or more AI-oriented, and also, combining the mutation label with synthetic to detect fake text that has been mutated under the right label.

3.6.2 Conclusion

The greatest barrier to entry for new academics will be learning to create models from neural networks and understanding the models themselves outside a black box. There are plenty of viable options for academia, including fake foreign language detection, low resource learning, detector adversarial attacks, domain specific attacks such as fake news and student essays and generalization. Because of this, the trend of today is rapid growth of all NLP fields at an increasing rate.

As for adversarial attacks, the COCO dataset was a great way to test how to overcome an attack on a detector based on mutation. The original under 10% accuracy of an attacked detector was boosted to 85%+ by having a heuristic as to what that mutation would be. This can be applied to any detector for any number of mutation operators.

This would be particularly helpful for students who write essays with GPT-3.5 and whom edit the essay to trick their professor. The future research in question from that chapter will be coming up with heuristics which accurately catch these deep fake texts.

Bibliography

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection, 2020. URL http://link.springer.com/10.1007/978-3-030-44041-1_114.

Izzat Alsmadi, Nura Aljaafari, Mahmoud Nazzal, Shadan Alhamed, Ahmad H. Sawalmeh, Conrado P. Vizcarra, Abdallah Khreishah, Muhammad Anan, Abdulelah Algosaibi, Mohammed Abdulaziz Al-Naeem, Adel Aldalbahi, and Abdulaziz Al-Humam. Adversarial machine learning in text processing: A literature survey. *IEEE Access*, 10:17043–17077, 2022. ISSN 21693536. doi: 10.1109/ACCESS.2022.3146405.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

C Wong arXiv preprint arXiv:1712.05419, undefined 2017, and Catherine Wong. Dancin seq2seq: Fooling text classifiers withadversarial text example generation. *arxiv.org*, 12 2017. URL http://arxiv.org/abs/1712.05419https://arxiv.org/abs/1712.05419.

R Avros, Z Volkovich International conference on machine learning, , and undefined 2018. Detection of computer-generated papers using one-class svm and cluster approaches. *Springer*, 2018. URL https://link.springer.com/chapter/10.1007/978-3-319-96133-0_4.

M Bao, J Li, J Zhang, H Peng 2019 International Joint ..., and undefined 2019. Learning semantic coherence for machine generated spam text detection. *ieeexplore.ieee.org*, 2019. URL https://ieeexplore.ieee.org/abstract/document/8852340/.

Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

Rasika Bhalerao, Mohammad Al-Rubaie, Anand Bhaskar, and Igor Markov. Data-driven mitigation of adversarial text perturbation. *arXiv preprint arXiv:2202.09483*, 2022.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Jérémie Bogaert, Marie-Catherine de Marneffe, Antonin Descampe, and Francois-Xavier Standaert. Automatic and manual detection of generated news: Case study, limitations and challenges. pages 18–26. ACM, 6 2022. ISBN 9781450392426. doi: 10.1145/3512732.3533589. URL https://dl.acm.org/doi/10.1145/3512732.3533589.

Sergio Bravo-Santos, Esther Guerra, and Juan de Lara. Testing chatbots with charm. In *International Conference on the Quality of Information and Communications Technology*, pages 426–438. Springer, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. Evaluation of text generation: A survey. 6 2020. doi: 10.48550/arxiv.2006.14799. URL https://arxiv.org/abs/2006.14799v2.

Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*, 2017.

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. Adversarial text generation via feature-mover's distance. *Advances in Neural Information Processing Systems*, 31, 2018.

X Chen, P Jin, S Jing, C Xie 2022 IEEE 10th Joint International, and undefined 2022. Automatic detection of chinese generated essayss based on pretrained bert. *ieeexplore.ieee.org*, 2022. URL https://ieeexplore.ieee.org/abstract/document/9836571/.

Alesia Chernikova and Alina Oprea. Fence: Feasible evasion attacks on neural networks in constrained environments. *ACM Transactions on Privacy and Security*, 25(4):1–34, 2022.

Ayesha Priyambada Das, Ajit Kumar Nayak, and Mamata Nayak. A survey on machine learning based text categorization. *academia.edu*, 2018. doi: 10.9790/0661-2002035156. URL https://www.academia.edu/download/56747916/I2002035156.pdf.

GH de Rosa, JP Papa Pattern Recognition, and undefined 2021. A survey on text generation using generative adversarial networks. *Elsevier*, 2021. URL https://www.sciencedirect.com/science/article/pii/S0031320321002855.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Nirav Diwan, Tanmoy Chakravorty, and Zubair Shafiq. Fingerprinting fine-tuned language models in the wild. 6 2021. URL http://arxiv.org/abs/2106.01703.

Chenhe Dong, Ying Shen, Min Yang, Yinghui Li, Haifan Gong, Miaoxin Chen, and Junxin Li. A survey of natural language generation. *undefined*, 1:38, 2021. doi: 10.1145/3554727. URL https://doi.org/10.1145/3554727.

Jiahua Dong, Yang Cong, Gan Sun, Yuyang Liu, and Xiaowei Xu. Cscl: Critical semantic-consistent learning for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 745–762. Springer, 2020.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. Roft: A tool for evaluating human detection of machine-generated text. pages 189–196, 10 2020. doi: 10.48550/arxiv.2010.03070. URL https://arxiv.org/abs/2010.03070v1.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. Text processing like humans do: Visually attacking and shielding nlp systems. *arXiv preprint arXiv:1903.11508*, 2019.

Noureen Fatima, Ali Shariq Imran, Zenun Kastrati, Sher Muhammad Daudpota, and Abdullah Soomro. A systematic literature review on text generation using deep neural network models. *IEEE Access*, 10:53490–53503, 5 2022. doi: 10.1109/ACCESS.2022.3174108.

Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. Unsupervised and distributional detection of machine-generated text. *arxiv.org*, 11 2021. URL http://arxiv.org/abs/2111.02878.

M Gambini. Developing and experimenting approaches for deepfake text detection on social media. 2020. URL https://etd.adm.unipi.it/t/etd-07032020-115029/.

Margherita Gambini, Tiziano Fagni, Fabrizio Falchi, and Maurizio Tesconi. On pushing deepfake tweet detection capabilities to the limits. pages 154–163. ACM, 6 2022. ISBN 9781450391917. doi: 10.1145/3501247.3531560. URL https://dl.acm.org/doi/10.1145/3501247.3531560.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. Gltr: Statistical detection and visualization of generated text. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, pages 111–116, 6 2019. URL http://arxiv.org/abs/1906.04043.

Zar Chi Su Su Hlaing and Myo Khaing. A detection and prevention technique on sql injection attacks. In *2020 IEEE Conference on Computer Applications (ICCA)*, pages 1–6. IEEE, 2020.

Keith J Holyoak. Parallel distributed processing: explorations in the microstructure of cognition. *Science*, 236:992–997, 1987.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. pages 1808–1822, 11 2019. doi: 10.48550/arxiv.1911.00650. URL https://arxiv.org/abs/1911.00650v2.

Touseef Iqbal and Shaima Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, 34:2515–2528, 6 2022. ISSN 1319-1578. doi: 10.1016/J.JKSUCI.2020.04.001.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

Ganesh Jawahar, Muhammad Abdul-Mageed, V.S. Laks Lakshmanan, M Abdul-Mageed arXiv preprint arXiv . . . , and undefined 2020. Automatic detection of machine generated text: A critical survey. *arxiv.org*, pages 2296–2309, 1 2020. doi: 10.18653/V1/2020.COLING-MAIN.208. URL `https://arxiv.org/abs/2011.01314https://aclanthology.org/2020.coling-main.208`.

Hongyan Jing. Using hidden markov modeling to decompose human-written summaries. *Computational linguistics*, 28(4):527–543, 2002.

DS Jones. Field computation by moment methods, 1969.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. Artificial text detection via examining the topology of attention maps. *arxiv.org*, 9 2021. URL `http://arxiv.org/abs/2109.04825`.

Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29, 2016.

Brandon Laughlin, Christopher Collins, Karthik Sankaranarayanan, and Khalil El-Khatib. A visual analytics framework for adversarial text generation. In *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–10. IEEE, 2019.

Thomas Lavergne, Tanguy Urvoy, François Yvon, T Lavergne, Á T Urvoy, and F Yvon. Filtering artificial texts with statistical machine learning techniques. *Springer*, 45:25–43, 3 2011. doi: 10.1007/s10579-009-9113-0. URL `https://link.springer.com/article/10.1007/s10579-009-9113-0`.

Thai Le, Suhang Wang, and Dongwon Lee. Malcom: Generating malicious comments to attack neural fake news detection models. *2020 IEEE International Conference on Data Mining (ICDM)*, 2020-Novem:282–291, 8 2020. URL `http://arxiv.org/abs/2009.01048`.

Bin Li, Yixuan Weng, Qiya Song, and Hanjun Deng. Artificial text detection with multiple training strategies. *dialog-21.ru*, 2022. doi: 10.28995/2075-7182-2022-20-375-381. URL `https://www.dialog-21.ru/media/5777/libplusetal104.pdf`.

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

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji Rong Wen. Pretrained language models for text generation: A survey. *IJCAI International Joint Conference on Artificial Intelligence*, pages 4492–4499, 1 2021. ISSN 10450823. doi: 10.24963/ijcai.2021/612. URL `https://doi.org/10.1145/nnnnnnn.nnnnnnnhttps://arxiv.org/abs/2201.05273v4`.

Gongbo Liang, Connor Greenwell, Yu Zhang, Xin Xing, Xiaoqin Wang, Ramakanth Kavuluru, and Nathan Jacobs. Contrastive cross-modal pre-training: A general strategy for small sample medical imaging. *IEEE Journal of Biomedical and Health Informatics*, 26(4):1640–1649, 2021.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. Adversarial ranking for language generation. *Advances in neural information processing systems*, 30, 2017.

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

Zhengyuan Liu and Nancy Chen. Entity-based de-noising modeling for controllable dialogue summarization. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 407–418, 2022.

Vijini Liyanage, Davide Buscaldi, A Nazarenko arXiv preprint arXiv:2202.02013, undefined 2022, and Adeline Nazarenko. A benchmark corpus for the detection of automatically generated text in academic publications. *arxiv.org*, 2 2022. URL `http://arxiv.org/abs/2202.02013https://arxiv.org/abs/2202.02013`.

Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. Neural text generation: Past, present and beyond. 3 2018. URL `http://arxiv.org/abs/1803.07133`.

Sidi Lu, Lantao Yu, Siyuan Feng, Yaoming Zhu, and Weinan Zhang. Cot: Cooperative training for generative modeling of discrete data. In *International Conference on Machine Learning*, pages 4164–4172. PMLR, 2019.

Narek Maloyan, Lomonosov Msu, Bulat Nutfullin, and Eugene Ilyushin. Dialog-22 ruatd generated text detection. 6 2022. URL http://arxiv.org/abs/2206.08029.

Tiziano; Falchi Fagni Fabrizio; Gambini Margherita; Martella Antonio; Tesconi Maurizio. Tweepfake: about detecting deepfake tweets. *PloS one*, 16:e0251415–NA, 2021. doi: 10.1371/journal.pone.0251415.

Ahmad Najee-Ullah, Luis Landeros, Yaroslav Balytskyi, and Sang-Yoon Chang. Towards detection of ai-generated texts and misinformation, 2022. URL https://link.springer.com/10.1007/978-3-031-10183-0_10.

AS Nayak. Deepspot: spotting fake reviews with sentiment analysis and text generation. 2019. URL https://csu-csus.esploro.exlibrisgroup.com/view/pdfCoverPage?instCode=01CALS_USL&filePid=13232648180001671&download=true.

Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles Sutton, J Doug Tygar, and Kai Xia. Exploiting machine learning to subvert your spam filter. *LEET*, 8(1):9, 2008.

Andrew Newell, Rahul Potharaju, Luojie Xiang, and Cristina Nita-Rotaru. On the practicality of integrity attacks on document-level sentiment analysis. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, pages 83–93, 2014.

James Newsome, Brad Karp, and Dawn Song. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*, pages 81–105. Springer, 2006.

Minh Tien Nguyen. Detection of automatically generated texts. 2018. URL https://tel.archives-ouvertes.fr/tel-01919207/.

Roberto Perdisci, David Dagon, Wenke Lee, Prahlad Fogla, and Monirul Sharif. Misleading worm signature generators using deliberate noise injection. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 15–pp. IEEE, 2006.

Saad Ahmed Qazi, Hina Kirn, Muhammad Anwar, Ashina Sadiq, Hafiz M Zeeshan, Imran Mehmood, and Rizwan Aslam Butt. Deepfake tweets detection using deep learning algorithms. *mdpi.com*, 2022. doi: 10.3390/engproc2022020002. URL https://www.mdpi.com/1747460.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J Doug Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pages 1–14, 2009.

Paolo Russu, Ambra Demontis, Battista Biggio, Giorgio Fumera, and Fabio Roli. Secure kernel machines against evasion attacks. In *Proceedings of the 2016 ACM workshop on artificial intelligence and security*, pages 59–69, 2016.

Skrylnikov S S, Posokhov P A, Makhnytkina O V, , , and . Artificial text detection in russian language: a bert-based approach. *dialog-21.ru*, 2022. doi: 10.28995/2075-7182-2022-21-470-476. URL https://www.dialog-21.ru/media/5786/posokhovpaplusetal117.pdf.

Sina Mahdipour Saravani, Indrakshi Indrajit Ray, and Indrakshi Indrajit Ray. Automated identification of social media bots using deepfake text detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13146 LNCS: 111–123, 2021. ISSN 16113349. doi: 10.1007/978-3-030-92571-0_7. URL https://link.springer.com/chapter/10.1007/978-3-030-92571-0_7.

Ali Sayghe, Junbo Zhao, and Charalambos Konstantinou. Evasion attacks with adversarial deep learning against power system state estimation. In *2020 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2020.

T Schuster, R Schuster, DJ Shah Computational ..., and undefined 2020. The limitations of stylometry for detecting machine-generated fake news. *direct.mit.edu*, 2020. doi: 10.1162/COLI. URL https://direct.mit.edu/coli/article-abstract/46/2/499/93369.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. Findings of the the ruatd shared task 2022 on artificial text detection in russian. *arxiv.org*, 6 2022. URL http://arxiv.org/abs/2206.01583.

Dingmeng Shi, Zhaocheng Ge, and Tengfei Zhao. Word-level textual adversarial attacking based on genetic algorithm. In *Third International Conference on Computer Communication and Network Security (CCNS 2022)*, volume 12453, pages 272–276. SPIE, 2022.

Rui Shu, Tianpei Xia, Laurie Williams, and Tim Menzies. Omni: automated ensemble with unexpected models against adversarial evasion attack. *Empirical Software Engineering*, 27(1):1–32, 2022.

Harald Stiff and Fredrik Johansson. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 5 2021. ISSN 23644168. doi: 10.1007/S41060-021-00299-5.

Reuben Tan, Bryan A. Plummer, and Kate Saenko. Detecting cross-modal inconsistency to defend against neural fake news. 9 2020. doi: 10.18653/v1/2020.

emnlp-main.163. URL http://arxiv.org/abs/2009.07698http://dx.doi.org/10.18653/v1/2020.emnlp-main.163.

Chen Tang, Frank Guerin, Yucheng Li, C Lin arXiv preprint arXiv:2203.03047, undefined 2022, and Chenghua Lin. Recent advances in neural text generation: A task-agnostic survey. *arxiv.org*, 3 2022. URL http://arxiv.org/abs/2203.03047https://arxiv.org/abs/2203.03047.

Senait G. Tesfagergish, Robertas Damaševičius, and Jurgita Kapočiūtė-Dzikienė. Deep fake recognition in tweets using text augmentation, word embeddings and deep learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12954 LNCS:523–538, 2021. ISSN 16113349. doi: 10.1007/978-3-030-86979-3_37.

Julien Tourille, Babacar Sow, Adrian Popescu, A Popescu of the 1st International Workshop on . . . , undefined 2022, and Adrian Popescu. Automatic detection of bot-generated tweets. *dl.acm.org*, pages 44–51, 6 2022. doi: 10.1145/3512732.3533584. URL https://dl.acm.org/doi/abs/10.1145/3512732.3533584.

Adaku Uchendu, Vladislav Mikhailov, Jooyoung Lee, Saranya Venkatraman, Tatiana Shavrina, and Ekaterina Artemova. Tutorial on artificial text detection. *artificial-text-detection.github.io*, 2021. URL https://artificial-text-detection.github.io/ATD_Tutorial.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Prashanth Vijayaraghavan and Deb Roy. Generating black-box adversarial examples for text classifiers using a deep reinforced model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 711–726. Springer, 2019.

W Wang, A Feng Mathematical Problems in Engineering, and undefined 2021. Self-information loss compensation learning for machine-generated text detection. *hindawi.com*, 2021. URL https://www.hindawi.com/journals/mpe/2021/6669468/.

Max Wolff. Attacking neural text detectors. 2 2020. URL http://arxiv.org/abs/2002.11768.

Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arxiv.org*, 4 2020. URL http://arxiv.org/abs/2004.01881.

Seyhmus Yilmaz, S Zavrak arXiv preprint arXiv:2207.08230, undefined 2022, and Sultan Zavrak. Troll tweet detection using contextualized word representations. *arxiv.org*, 7 2022. URL http://arxiv.org/abs/2207.08230https://arxiv.org/abs/2207.08230.

Haiyan Yin, Dingcheng Li, Xu Li, and Ping Li. Meta-cotgan: A meta cooperative training paradigm for improving adversarial text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9466–9473, 2020.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in Neural Information Processing Systems*, 32, 5 2019. ISSN 10495258. doi: 10.48550/arxiv.1905.12616. URL https://arxiv.org/abs/1905.12616v3.

Fuyong Zhang, Yi Wang, Shigang Liu, and Hua Wang. Decision-based evasion attacks on tree ensemble classifiers. *World Wide Web*, 23(5):2957–2977, 2020.

Yu Zhang, Gongbo Liang, and Nathan Jacobs. Dynamic feature alignment for semi-supervised domain adaptation. *British Machine Vision Conference (BMVC)*, 2022.

Yu Zhao, Ting Su, Yang Liu, Wei Zheng, Xiaoxue Wu, Ramakanth Kavuluru, William GJ Halfond, and Tingting Yu. Recdroid+: Automated end-to-end crash reproduction from bug reports for android apps. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(3):1–33, 2022.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. Neural deepfake detection with factual structure of text. *arxiv.org*, 10 2020. URL http://arxiv.org/abs/2010.07475.

A. Attachments

A.1 Digital attachments

A.1.1 GitHub repos

https://github.com/mwolff31/attacking_neural_text_detectors
https://github.com/JesseGuerrero/Mutation-Based-Text-Detection

A.1.2 Datasets used

https://github.com/tylin/coco-caption

VITA

Jesus A. Guerrero or ”Jesse”, was born in San Antonio, Texas, on December 9th, 1991. They attended elementary to high school in the SAISD district of San Antonio, graduating high school with honors in May 2010. In August 2017 they began schooling for a bachelors of Business Administration at Texas A&M - San Antonio, graduating in December 2020. They entered graduate school at Texas A&M - San Antonio in January 2021 and received a Masters of Science degree in Computer Science in May 2023.

PUBLICATIONS

Guerrero, J., & Alsmadi, I. (2022). Synthetic Text Detection: Systemic Literature Review. ArXiv Preprint ArXiv:2210. 06336.

Guerrero, J., Liang, G., & Alsmadi, I. (2022). A Mutation-based Text Generation for Adversarial Machine Learning Applications. ArXiv Preprint ArXiv:2212. 11808.

Liang, G., Guerrero, J., & Alsmadi, I. (2023). Mutation-Based Adversarial Attacks on Neural Text Detectors. ArXiv Preprint ArXiv:2302. 05794.

Kishiyama, B., Guerrero, J., & Alsmadi, I. (2023). Security Policies Automation in Software Defined Networking. Available at SSRN 4384690.

Liang, G., Guerrero, J., Zheng, F., & Alsmadi, I. (2023). Enhancing Neural Text Detector Robustness with u Attacking and RR-Training. Electronics, 12(8), 1948.