Text Generation with AI Driven Detection and prevention

Submitted by

B.SWETHA

M.VARALAKSHMI

C.NATHIYA

Text Generation with AI Driven Detection and Prevention

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 promptanswer 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:

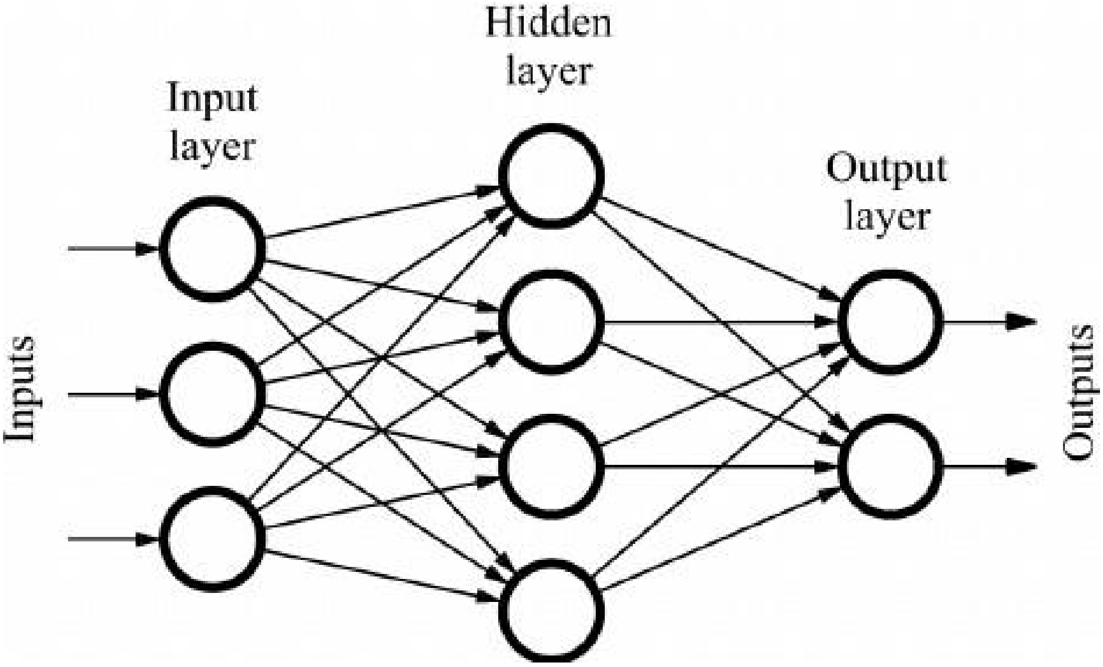


Figure 1.1: A basic perceptron

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.

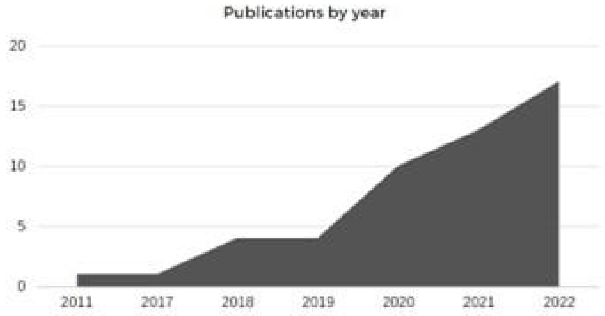


Figure 1.2: Number of detector articles per year

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.

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.

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.

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.

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.

Searching Strategy To Retrieve Studies

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

.

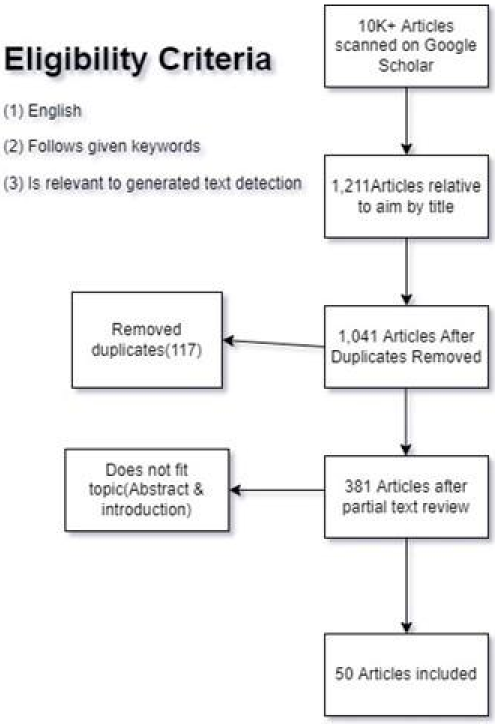
Article Inclusion Exclusion Criteria

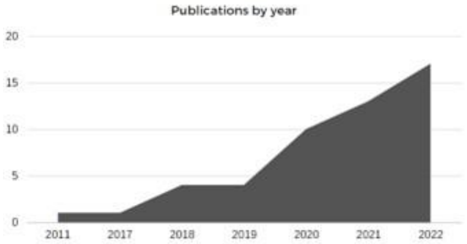
The following is the inclusion criteria:

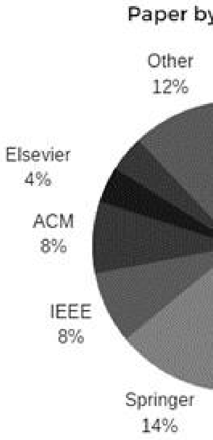
- The article must include machine generated text classification or be highly relevant
- The article can include other languages in its dataset
- The article itself must be written in English
- Surveys on text generation are allowed The following is the

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

Systematic Mapping Study Results


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
1,041 Articles After Duplicates Removed
Removed duplicates(117)
381 Articles after partial text review
Does not fit topic(Abstract & introduction)
50 Articles included

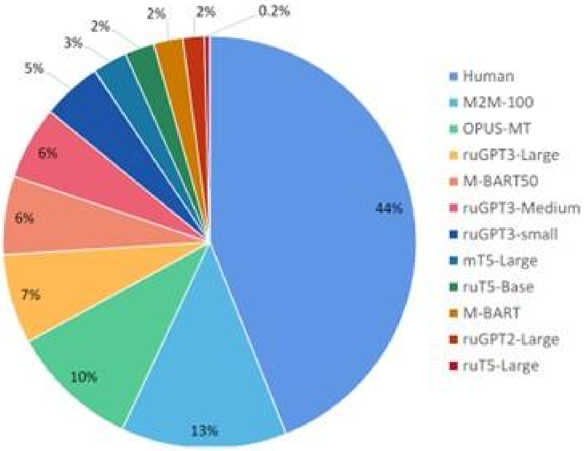




Publications by year

Articles by publisher

Example evaluation based on source



These new models give us more options than the standard detector, opening up more difficult and niche types.

Identified research gaps

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.

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 .

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].

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.

Our mutation-based approach in this paper can fall under the first category of the last classification (i.e. poisoning attacks). 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).

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.

Related Works

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

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 keyboardadjacent 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	Plea˜se lˆıke and sharˆe 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

Adversarial text perturbations

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.

Approach: Use a finite set of operators for research customization

. 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.

https://github.com/ JesseGuerrero/Mutation-Based-Text-Detection has some written operators which can be viewed as examples.

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

Experimental mutation operators

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.

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..

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.

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.

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:

	Accuracy
	∼88.80%(1000 samples)
Randomized	
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)

Preliminary Results

.

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	
None	
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)

Individual captions, short language modeling

%