Generation of Adversarial Fake Reviews using GEMS

Suraj Chatrathi*
Georgia Institute of Technology
Atlanta, Georgia, USA
schatrathi3@gatech.edu

Madelyn Scandlen*
Georgia Institute of Technology
Atlanta, Georgia, USA
mscandlen3@gatech.edu

Nikolai Warner*
Georgia Institute of Technology
Atlanta, Georgia, USA
nwarner30@gatech.edu

ABSTRACT

We propose the review generation framework GEMS (Generate, Explain, and Maintain Style) to improve upon fake review detection, explainability, and generalizability from one domain to another. Online reviews have become increasingly ubiquitous, and users trust and seek them out to make informed decisions about spending their time and money. Bad actors can use online reviews for malicious purposes, such as targeted review bombing, or propping up a scam product with positive reviews. Detection results have improved over the last few years, but in order to increase adoption, users must understand and trust AI moderation decisions. Using real and AI generated review data from Yelp and other sources, we leverage the GEMS framework by generating reviews steered by semantic and stylistic labeling which can be used to better mimic real user reviews in the name of threat modeling. These generated reviews were tested through a word-level neural network to assess if the generated reviews would fool a state-of-the-art detection classifier, finding that the stylized reviews sacrificed ability to evade detection but were able to be more emotive.

KEYWORDS

datasets, neural networks, generative adversarial network, text deceptive reviews

ACM Reference Format:
Suraj Chatrathi, Madelyn Scandlen, and Nikolai Warner. 2023. Generation of Adversarial Fake Reviews using GEMS. In *Proceedings of CSE 8803 (Data Science for Social Networks).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online reviews are important to both customers and companies, especially when potential future customers use reviews to gauge whether or not they will use or buy a product or service from a company in the future [1]. Customer reviews have increased a lot in the last decade, and it has been observed that positive and negative reviews bring positive and negative financial effects to companies accordingly [2]. Because of the power these reviews hold, it is important to combat the spread of fake reviews. To define them, fake reviews are posted by users who have not actually had any experience with the product they are reviewing [3]. Fake reviews come in different forms but the ones that are most interesting from a research standpoint are reviews that are both untruthful and about the product in question [4]. It is very hard even for humans to differentiate between a fake and real review. Even using manually annotated data, the best models have only been able to reach around 60% accuracy [5]. These massive fake review attacks are called crowd-turfing [6].

As computing power rises and online platforms gain more and more popularity, bad actors will increasingly turn to automating their fake review attacks. This makes it important for detection models to be robust to computer generated reviews as they are cheaper and less-time consuming than human generated reviews. Automated crowd-turfing is a relatively new form of attack but one that will gain popularity as the technology develops [6]. as it is much more cost and time-effective than manually creating fake reviews.

In addition to this, there has been a lack of research into how style affects human users' ability to discern real reviews from computer generated ones. It is our intuition that reviews that incorporate some sort of style whether it be excitement, humor, or even anger, will be more convincing to humans than a review that sounds monotonous and too businesslike.

This brings into scope our objective, which is to build an adversarial framework that can generate fake reviews for the purpose of threat modeling. Not only will studying this help to improve the robustness of these fake review detection models, it will also be able to generate data for future work, as annotating fake reviews by hand is a slow-process. The main research question we are trying to solve is: How to leverage word level embeddings to generate realistic adversarial reviews that can fool both AI detection algorithms human readers? Within this there is a sub-question: Does incorporating style control affect the efficacy of the adversarial reviews/attacks?

The largest challenge to detecting fake reviews is the lack of ground-truth labels. Even websites like Yelp that have algorithms to filter fake reviews must perform this task unsupervised. There has not been much work done in generating deceptive reviews that can fool humans and filtering models. The current state-of-the-art generative adversarial network for fake reviews, FakeGAN, had a bottleneck in its generation capabilities and was focused on creating a strong discriminator rather than a generator [7]. FakeGAN's performance was only measured on one dataset of hotel reviews and did not incorporate the expressive linguistic and behavioral features from previous research [5], [8].

We propose a model Generate, Explain, and Maintain Style (GEMS) based on large language models that can fool both humans and detection models in different topic domains by incorporating

*All authors contributed equally to this research.

Data Science for Social Networks, December 2022, Atlanta, GA, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

stylistic control using Plug and Play Language Models (PPLM) [9]. These four style categories will be: humor, surprise, anger, and serious. We will then combine the generated deceptive reviews with the original dataset which contains both human and computer-generated reviews, adding ground truth labels of machine-written or human-written to evaluate if GEMS can fool a detection model trained on fake reviews.

The data used came from filtered Yelp data as well as data from a paper by Salminen et al. which contains both human and computer-generated reviews [10].

The reviews that GEMS was able to generate were able to succeed in incorporating and showing off a degree of stylistic control compared to the reviews generated by multiple baselines, however the performance regarding the fake review detection task was less then stellar.

2 LITERATURE REVIEW

Detecting deceptive reviews is a popular natural language processing task similar to detecting spam emails and fake news articles. Linguistic features were first looked at into detecting fake reviews, with n-grams proving to perform the best in statistical detection tests, but only with 67.8% accuracy [8]. Due to the difficulty of accurately detecting fake reviews by only linguistic features, methods using behavioral data of reviewers were proposed, creating graph representations of the review network. FraudEagle was one of the first algorithms to predict spam reviewers based on a graph representation [11]. The shortcoming of using linguistic features is the low performance on unseen observations. Review networks fail for the cold-start problem, where behavioral information of fake reviewers does not exist.

Neural networks using embeddings of linguistic and behavioral features were proposed to account for the cold-start problem. Wang et al. proposed jointly embedding linguistic and behavioral features to detect fake reviews in the most difficult contexts [12]. Behavior is encoded using an embedding learning model representing the product, reviewer, and review as a 3-tuple, and textual information was encoded using a convolutional neural network. The model using these joint-embeddings outperformed the state-of-the-art methods, demonstrating the importance of including linguistic and behavioral features in unsupervised detection tasks. Being an unsupervised task can allow this model to be employed by websites that rely on reviews such as Yelp, but a supervised task with ground truth labels may improve accuracy. The current research of deceptive review detection is built from these early findings about the importance of feature extraction, and the generative model we propose expands on this.

Neural network detectors are limited by the lack of ground truth labels for real and fake reviews, so creating a generative model to write deceptive reviews with machine-written labels can improve detection. Aghakhani et al. create FakeGAN, which is based on Generative Adversarial Network, to generate deceptive reviews [7]. FakeGAN uses a semi-supervised technique of using two discriminators to evaluate for deceptiveness and truthfulness and a generator. The learning of adversarial reviews improved compared to truthful reviews, but the combination of discriminators for both tasks had the highest accuracy. The implementation of FakeGAN causes slower training time for each adversarial step, which can be a bottleneck for large datasets. Our model GEMS will generate more deceptive reviews to increase the performance of classifiers of detecting adversarial reviews on larger datasets.

Generative adversarial networks have been used for other tasks, such as generating fake news articles. Le et al. proposed a framework Malcom to generate high-quality end-to-end adversarial news texts, which fooled state-of-the-art fake news detection models into classifying machine-generated texts as real news 90% of the time on average [13]. Malcom incorporated style and attack modules in the generator to generate high-quality texts that can deceive humans and also deceive detection classifiers. Fake news is different from fake reviews in linguistic style and in the metadata. The source of fake news as credible or incredible can inform detection tasks, whereas spam reviewers have different behaviors to real reviewers that will be learned in our model. Our model will use a similar stylistic ideology to generate reviews.

3 DATA

So far we have identified two datasets to use for generating fake reviews.

3.0.1 Yelp Reviews Dataset. This is a dataset containing over five million reviews from Yelp on a wide variety of services that was taken from Kaggle. The original data came in the form of two separate datasets, one for Yelp reviews, and one for the corresponding Yelp users. If any row in the data had missing values, the row was removed. The 'date' feature in the reviews dataset was converted from string values to dateTime objects and the '$yelping_since$' feature from the user dataset was converted into numerical values representing the age of the user's account. The two datasets were combined with a join function on the $user_id feature$. All of this pre-processing was done through the use of Pandas.

This dataset provides a huge amount of text data in the form of user reviews, as well as accompanying user data that provides more context for the reviews such as account age and the number of posts a user has made on Yelp. Having this type of data would allow our model to better learn better because they are Real reviews from real people that have passed through Yelp's filtering system.

As mentioned, there were over five million reviews in total, with the reviews having a mean of 611 characters and 113 words. In addition to this, the vocabulary size was very large, numbering 262046. On the user side, the mean account age was 9.3 years and the mean number of reviews by user was 23. Since all of these have passed through Yelp's own filtering system, their ground truth labels are assumed to "real human-generated reviews".

There were some interesting insights to be gleaned from the data, as seen in Figure 1. The user and review tables were joined to measure the correlation between the combined features. We observe that there is a high correlation between users finding reviews funny and useful and also cool and useful. This gives credence to our goal of wanting to incorporate style control into the review generation, especially because humor is one of our four categories we focus on. In addition to this, there are slight correlations between account age and the average number of words/characters per review, which suggests that longer-tenured users write longer reviews than newer ones.

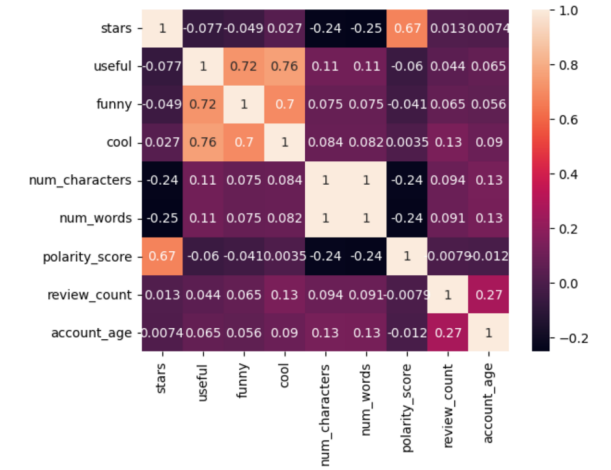

Figure 1: Correlation matrix of Yelp review data

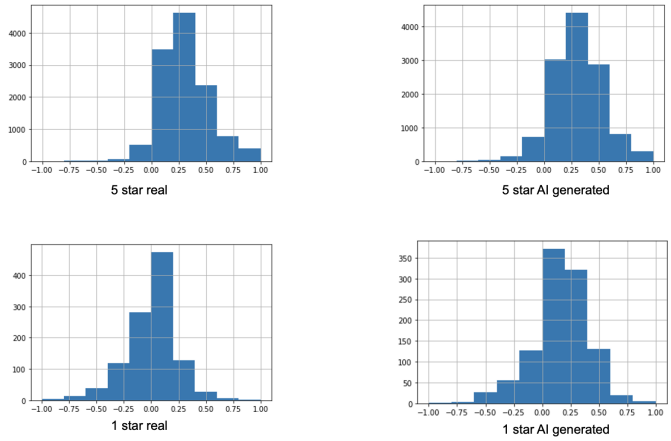

Figure 2: Sentiment Polarity Analysis

3.0.2 Retail Products Fake Reviews. This dataset contains fake and real reviews about a wide array of retail products including books, games, bathroom products, Kindle books, clothing, sports equipment, and movies. The fake reviews in this dataset were generated and it will be important for our framework as it can use these generated reviews to learn to generate reviews itself, and by comparing those with the language and composition of the real reviews, it will be able to differentiate and produce higher quality generations than the ones found in this dataset. The dataset is taken from a recent study done on generating fake reviews by Salminen et al. [10]. We will pre-process the data using Pandas and then split it up into training, test, and validation groups. We will pass a filter over the review text to get rid of any special characters or extraneous information like links. In addition to this, since this dataset is quite large we will simply remove data points that have missing values. In total there are 40,000 data points in this dataset. Half of them are real reviews, and half of them are fake. There are over 2500 unique tokens in the dataset and the reviews are around 2 sentences on average. Each review has a mean word length of 67 and a mean character length of 351, and on average having 4 words per sentence.

There are some useful insights to take away from this dataset as well after initial exploratory analysis. As seen in Figure 2, it was observed that the sentiment of the fake reviews tended to be higher no matter if they were one star reviews or five star reviews, this is an important distinction because it shows a possible aspect of the computer-generated reviews that can be easily identified by a detection task. The sentiment here is based on a polarity score from -1 (most negative) to 1 (most positive). This will be useful during review generation as our framework can use this difference to better generate reviews that will try and evade detection.

4 EXPERIMENT

4.1 Experiment Setting

The task is to generate realistic, AI-written reviews based on a dataset of real, human-written Yelp reviews. The proposed model GEMS will incorporate style to make the reviews more emotive and more likely to evade human detection and flagging. The reviews will be evaluated in a classification task where a convolutional neural network (CNN) will attempt to distinguish the human-written reviews from the computer-generated reviews. The baselines proposed are a character-level recurrent neural network inspired by Yao et al. [6] and a fine-tuned Generative Pre-trained Transformer-2 (GPT-2), developed by Radford et al. [14]. For classification, an 80-20 train-test split was performed on the Fake Reviews dataset to train the classifier and evaluate its performance on Salminen's computer-generated reviews [10]. Precision, recall, and F1-score were assessed to compare the GEMS model with style to the baselines.

4.2 Baselines

We implemented a character-level Recurrent Neural Network (RNN) for our baseline architecture based on Yao et al., who use a generative RNN framework to perform review customization [6]. An initial review may contain a positive or negative sentiment, and have details that concern a particular restaurant or establishment. The authors have a three step attack: first, choose the type of contextual information to be capture, like food. Second, identify words in the reference reviews that match the given contextual domain. Finally, replace words by possible lexical similarity score. The features are rather hand-crafted to perform review customization. When we implemented a generative character based RNN, we chose to customize reviews instead by the sentiment of the prompt ("I hate this restaurant", or "I love this restaurant"). Since the model was trained from scratch, many of the reviews suffered from small but noticeable grammatical or spelling errors. We therefore focused our efforts on word-level language models to explore stylistic control.

Our second baseline was a fine-tuned GPT-2 framework [14], for language tasks. We fine-tuned the GPT-2 with pre-trained weights model on the Yelp dataset, to learn to match the style of generated reviews. Without fine-tuning, the GPT-2 model effectively generated text based off input prompts, but the generations were often out of domain from the customer review focus area. GPT-2 is a very large model, and so in order to effectively train it, we used gradient-accumulation. In gradient accumulation, instead of making updates

to the model parameters after every batch of data, each batch is split into several mini-batches. This allows all of the data to fit on the memory of the GPU card, while approximating the updates of larger batch sizes needed for stable training. After gradients were collected for several mini-batches, the loss was back-propagated upon and the optimizer was stepped. To perform the fine-tuning, we employed a Nvidia RTX 3090 with 24GB of VRAM and a Xeon 2175 multi-core CPU.

4.3 Proposed Method

After fine-tuning GPT-2 on our Yelp reviews dataset, the model was very capable of controlling sentiment just from the tone of our initial prompt. We hypothesize that this is because the Yelp reviews dataset contains an abundance of very low and very high rated reviews. This is generally a phenomenon because customers aren't motivated to review a business unless they have a particularly favorable or averse experience. Using the prompt "This restaurant is awful", we generated the following reviews:

- I've been here twice and still have no idea what went wrong. But the food was alright. We were asked to leave and if we wanted to go back, we could go back. It was my fault for not checking in with the manager before we left.
- I went back to the last time I went and it was the worst place in town. I'm sure it was bad enough that it can't be repeated.
- I ordered the first casserole and it was ok. The second casserole was OK. The third was okay. The fourth was ok. The fifth was ok. The sixth was ok. The seventh was ok.

Using the prompt "This restaurant is amazing", we generated the following reviews:

- I was excited to try it out. I think the food is very good. I love the chicken wings and the desserts. It was amazing that the staff was friendly and the food was delicious. The service was good. I'm sure I'll try the shrimp and shrimp tacos again.
- I'm sure it would have been nice if they had a menu but the prices were crazy and we never went again.
- I've been here since I went back in the 90s and I'm still in love. We've had great food and drinks. We will definitely come back here again.

The baseline fine-tuned GPT-2 model on our reviews dataset represents a very capable model and controlling basic style. We trained it on 25,000 reviews out of 1,500,000 reviews due to compute hardware limitations and time-efficiency. Even with this relatively small subset, each epoch took 15 minutes due to the size of the model. However, our goal is generate a realistic distribution of reviews, not just ones that are polarized as poor or stellar. Thus, we also explore controlling the tone of reviews in addition to the sentiment.

To make generations more realistic, we explored generating a variety of styles for review. We classified a subset of reviews by sentiment as well as tone. The possible sentiments were positive, negative, and neutral. The possible tones were serious, funny, surprised, and angry. Based off a Dathathri et al., we implemented a style-control from a Bag of Words Plug and Play Language Model (PPLM) [9]. For each tone, we created a representative bag of words dictionary containing common phrases that define the given tone. For example, some words defining anger are "sucks", "hate", "terrible", and "worst." Example characteristic words for serious are "gravely", "grim", and "alarming." Words defining surprise included "amaze", "confuse", and "taken aback." Finally, words characterizing humor included "priceless", "jest", and "wise guy." These bag of words were informed based upon manual annotations of the dataset.

After the bag of word dictionaries are defined, the PPLM module performs several forward- and backwards-propagations on the fine-tuned model. It calculates the probability of generating the desired words under the original distribution, then updates the weights to increase the likelihood of generating desired words of a particular tone or sentiment in the latent representation.

There were several hyper-parameters to tune to create more realistic generations: in particular, the PPLM model ran into two common issues. The generations either didn't include many words from the target domain, or the generations were repetitive. Here are examples of failed repetitive generations:

- Our family favorite restaurant has a huge selection of food items, but they always seem to be busy. The menu is always changing and it always surprises surprise surprise surprise surprise.
- This restaurant and its surroundings have been a favorite spot for over 30 years. Our staff has a great vibe and we always have a smile on our faces. We have also had the chance to meet some of the best laugh laugh laugh laugh in the world.

During PPLM generation, a hyper-parameter called the KL scale how much the model penalizes the KL divergence between the original distribution and the modified, style-controlled distribution. A higher KL scale would decrease the probability of repetitive generations, but also decrease the probability of incorporating novel, desired words into the generations.

Similarly, the PPLM module relies on updating the latent representation during decoding. To use GPT-2 for generation, each word is generated sequentially until the model encounters an end-of-sequence (EOS) token. Therefore, previous time steps affect the probability distribution at the current time step. Hyper-parameters defining the window length controlled how many previous words are considered when updating the current distribution to encourage generation of target words. Similarly, the horizon length defined how many time steps into the future were considered. We also tuned the step size update, which controlled how large of an update was made on each iteration, as well as the overall number of iterations.

5 RESULTS

5.1 Generation

Generation was performed by feeding a set of prompts to the models and allowing for a character-by-character or word-by-word output to be decoded and concatenated until an end-of-sequence token was generated or 40 words were reached. The prompts were obtained by finding the most popular n-grams that start the Yelp reviews, varying from n=1 to n=9. This generation method was created to target existing reviews and include novelty in generation. The reviews were more on topic as the length of the prompts increased;

Model	Review
Human	I've eaten here numerous times and am still amazed how popular these places are.
Character-RNN	I've received small fancy and a staff other her side and trensy hands di for an hour. A little sport, it's well.
Fine-tuned GPT-2	I've just finished reading the next chapter of Fringe! It's a gorgeous novel. Thank you so much for reading.
Fine-tuned GPT-2 with *serious* style modifier	I've always found the more I've learned the less I like the way it looks and the more I like it I don't get it anymore. It's all so bad and the fact the bad stuff happens to me the more I'm bad.

Table 1: Generations on prompt *I've*

Style Modifier	Review
Serious	I've been coming here since the very beginning, so I can't really say I've had a good run in. The first time I tried this place was a couple of years ago, and the place really took care of me. I've had my fair share
Surprise	I've been coming here since the very beginning, so I can't really say I've had a good run in. The first time I visited, my boyfriend had to take my picture, and the second time he did he was so disappointed. I've had my share of
Humor	I've been coming here since the very beginning, so I can't really say I love it, but it was fun to watch it. I don't like to think I'm going back to the old school humor. The comedy is so funny. The laugh is
Anger	I've been coming here since the very beginning! We have the most amazing store I've ever seen and the worst. The worst place to shop for your junk. We hate it so much and hate to be rude but I will never hate you so bad. Thank you

Table 2: Fine-tuned GPT-2 generations on prompt *I've been coming here* with different style modifiers

however, depending on the prompt this could be plagiarized and quickly detected by humans. The researchers conducted qualitative analysis to find the best length of prompts for the reviews that could remain on-topic about a product or service, while also appearing novel. Generated reviews can be found in Table 1.

The GEMS model generated reviews in four different styles for each prompt. Styles like anger and serious were qualitatively better than the other styles. Comparison between styles for each prompt can be seen in Table 2. There were limitations on generation due to the large amount of data and high computing resources needed for text generation tasks, so 160 reviews, 40 for each style, were generated in total.

Model	Precision	Recall	F1-Score
Character-RNN	0.877	0.924	0.900
Fine-tuned GPT-2	0.887	0.924	0.905
GEMS	0.962	0.924	0.943

Table 3: Detection metrics for the models

5.2 Detection

To evaluate the proposed method, we build a classification neural network to detect whether a given sequence of text is a true human written review or a generated review. We train the classifier on the Fake Reviews dataset, which is a balanced set of original human-written reviews and computer-generated reviews by Salminen et al. [10]. We use a sample of 16111 generated reviews and 16234 real reviews for training, where the text is cleaned and sentence embeddings are created using word token indices in the vocabulary, padded to length of 60 words. The architecture was a convolutional neural network (CNN), which achieved the best detection performance from Wang, consisting of an embedding layer, a global pooling layer, and a sigmoid activation to output class predictions for each sequence of tokens [12]. The classifier achieved 0.9238 binary accuracy on the balanced training data after training for 10 epochs. On the test set sample of the Fake Reviews dataset, consisting of 4105 generated reviews and 3982 real reviews, the classifier achieves an accuracy of 0.868.

This classifier is then used to evaluate the set of generated reviews from the Yelp dataset and the real Yelp review counterparts. The real Yelp reviews in the test sets consisted of all reviews beginning with a prompt, with duplicate reviews removed. Three sets of data are tested, consisting of the 3413 real reviews and the generated reviews by each of the three models. The classifier was first evaluated on real Yelp reviews and 450 character-level-RNN generated reviews, then on 3413 real reviews and 450 fine-tuned GPT-2 generated reviews, and lastly on 3413 real Yelp reviews and the 160 style reviews. Results are shown in Table 3. The classifier achieved high performance at detecting across all models, showing that the generations are not able to avoid being flagged.

The character-level-RNN generations were the most difficult to correctly predict as computer-generated or human-written, with the classifier achieving an F1-score of 0.900. The classifier achieved an F1-score of 0.905 on the fine-tuned GPT-2 reviews. The GEMS model with style control was the most correctly identified with an F1-score of 0.943. A potential reason the character-level reviews were the hardest to detect was the large presence of nonsense non-English language words that would be realized as unknown tokens that fooled the word-level classifier.

6 CONCLUSION

Through developing the GEMS framework we hope that it lays down a foundation for threat modeling against computer-generated fake review attacks, as we believe these will only increase in frequency in the future.

6.1 Ethical Discussion

As with any generative model, there are clear ethical concerns that have to be accounted for with this kind of development. The ability to steer review generation with style could lead to malicious activity when in the hands of bad actors, and for that reason we decided to not release the code publicly. While code being open-source is often beneficial to the community, the risks do not outweigh the advantages in this case. This is a big problem in general, in recent years we have seen an uptick in deep-fakes and other misuses of generative modeling, especially in the vision field. Similarly, we do not wish to contribute to any potential malicious activities by releasing our code.

6.2 Limitations and Shortcomings

While the GEMS framework found some success there were also some limitations. Chief of these were that the generation and evaluation when using detection models took too much computation power, more than we had access to. Also, having to fine-tune on large amounts of user data for the GPT-2 baseline model took a lot of manual effort and could have been done more thoroughly if not for lack of time. In addition to this, computer-generated review data is very sparse in general, which did not allow us to have data coming from a rich set of sources, which in the end could lead to some bias in terms of our experimental results. Another limitation is that the bag-of-words framework that we used to steer the GPT-2 model was not robust for every task. Lastly, since the Yelp reviews used were already filtered, there was no noisy data in the form of badly written human reviews, which only made it easier for the detection model to classify the generated reviews as fake.

Some of the shortcomings included the fact that a large input context was needed for the reviews generated by GEMS to stay on topic. In addition to this, the detection model we used in the experiments was easily able to detect the generated reviews, meaning there is much room for improvement there. Also, the GPT-2 reviews that had style incorporate were easier to detect than the ones that were not, which leads us to believe that style can make reviews more identifiable, and more conspicuous. This means that in the future there must be effort put into making the generated reviews seem more natural even with an incorporation of style.

6.3 Extensions and Future Work

A possible extension to the project would be improving the scalability to train on more real reviews and to generate more reviews using GEMS. The fake review detection would also be improved by increasing the number of generated reviews and using more features selected in state-of-the-art detection models. In addition to this, create a large, hand-annotated dataset with a style feature based on the four style categories outlined earlier. This would help for future developments to be able to generate more believable reviews. Future work can involve the incorporation of Generative Adversarial Networks into the GEMS framework and also the incorporation of behavioral features. This would combine things like user attributes such as post frequency and time of review posted with linguistic features already within GEMS.

6.4 Contributions

All team members contributed a similar amount of effort to this project.

REFERENCES

[1] Eileen Fitzpatrick, Joan Bachenko, and Tommaso Fornaciari. 2017. Automatic detection of verbal deception. *Computational Linguistics*, 43, 1, (Apr. 2017), 269–271. DOI: 10.1162/COLI_r_00282.

[2] Nga N. Ho-Dac, Stephen J. Carson, and William L. Moore. 2013. The effects of positive and negative online customer reviews: do brand strength and category maturity matter? *Journal of Marketing*, 77, 6, 37–53.

[3] Atefeh Heydari, Mohammad ali Tavakoli, Naomie Salim, and Zahra Heydari. 2015. Detection of review spam: a survey. *Expert System Applications*, 42, 3634–3642.

[4] Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. Association for Computing Machinery, New York, NY, USA, 219–230.

[5] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 309–319.

[6] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y. Zhao. 2017. Automated crowdturfing attacks and defenses in online review systems. (2017).

[7] Hojjat Aghakhani, Aravind Machiry, Shirin Nilizadeh, Christopher Krügel, and Giovanni Vigna. 2018. Detecting deceptive reviews using generative adversarial networks. *2018 IEEE Security and Privacy Workshops (SPW)*, 89–95.

[8] Arjun Mukherjee, Bing Liu, and Natalie S. Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*. Association for Computing Machinery, New York, NY, USA, 191–200.

[9] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *CoRR*, abs/1912.02164.

[10] Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, and Bernard J. Jansen. 2022. Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services*.

[11] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion fraud detection in online reviews by network effects. In *Proceedings of the 7th International AAAI Conference on Web and Social Media*.

[12] Xuepeng Wang, Kang Liu, and Jun Zhao. 2017. Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 366–376.

[13] Thai Le, Suhang Wang, and Dongwon Lee. 2020. Malcom: generating malicious comments to attack neural fake news detection models. *IEEE International Conference on Data Mining*.

[14] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In.

Received 8 December 2022