Fake Review Generation via Large Language Models

Group 3:
ID: 18064052, 17126493, 17112496, 18001745

Abstract

The growing landscape of synthetically generated data is altering historical beliefs about the reliability and authenticity of digital interactions. However, modern acceleration in generative models have democratised access, essentially allowing anyone with consumer-grade hardware to generate and produce data that appears authentic and human-like. While scholarly research on the subject is nascent, this study aims to examine and critically analyse the feasibility — and evasive ability — of generating synthetic reviews via large language models. We trained GPT-2 on scraped Amazon reviews and tested its human-likeness against commonly cited SOTA review classification models in which we show that the SOTA models do not generalise well across product domains, and are easily fooled by such synthetic data. Furthermore, a comprehensive evaluation of these synthetic reviews is also provided with a focus on the textual features that classify a review to appear authentic.

1 Introduction

In 2020, the COVID-19 outbreak forced governments worldwide to impose strict lockdowns, which was followed by an unprecedented acceleration in e-commerce sales as consumers carried out their purchases online. This led to the industry — which is expected to grow at 12.7 percent in 2022 — disrupting buying patterns and traditional retail marketing solutions [1].

Scholarly research on the subject suggests that a product's review is a critical factor in influencing its popularity and appeal [2]. However, as with many areas of the internet, while well-intentioned valid users are creating online reviews, there are also malevolent users and bots maliciously affecting product authenticity and revenue. A recent study by go-to-market security provider *Cheq* indicates that online reviews influenced $3.8 trillion of global e-commerce spending in 2021, with fake reviews having a direct influence on over $152 billion of online spending [3].

While the overall risk is hard to quantify, given the sparse scholarly research on the subject, this study aims to investigate the feasibility of generating human-like fake reviews using large language models; along with analysing their evasive ability to deceive state-of-the-art (SOTA) model-based review classifiers against genuine (human) reviews. It is organised as follows: after the introduction, an overview of related work surrounding standard review classification techniques is presented. Thereafter, a description and motivation of our chosen datasets are put forward, followed by a review of the methodology surrounding language model review generation and the model-based classification techniques used. Following this, we outline the experimental design following how the generation and classification models were built, fine-tuned, and evaluated. Next, we perform a comprehensive evaluation surrounding our iterative language model improvement process and critique outcomes from review generations while testing SOTA and classical classification techniques against our findings. Finally, the paper concludes by discussing the implications of our results and suggestions for future research.

2 Related Work

The rapid growth in online marketing has changed the way people approach new products. The promotion was previously done by advertisements or word of mouth, but previous research showed that when it comes to online shopping, user reviews were a critical factor in affecting final decisions [4]. However, customers cannot determine whether the reviews are genuine or misleading [5]. Thus, detecting misleading fake reviews would be one

of the significant studies that affect the reputation of all e-commerce platforms. Currently, fake review detections mainly focus on two domains: textual and behavioural features. The textual analysis examines the context of the reviews while the behavioural analysis inspects the spammers' behaviours [5]. An early study [6] provided a novel idea for tackling this complicated issue. Instead of understanding the contextual meaning of the review, [6] mentioned that spammers tend to duplicate their reviews and apply them to different places just by altering the keywords. By applying logistic regression, [6] achieved 78% area under the curve (AUC). Inspired by this concept, several studies focused on finding the similarity score between different reviews to determine whether the reviews are fake or not [7][8]. Similar to examining the spammers' behaviours, [9] combined the relationships between the reviews, users, and shops to generate a review graph. That way, the only material required is the credibility and impartiality of the users and shops, and there will be no need to understand the meaning of the review content. Also, studies dug into users' spatial patterns and found that fake reviews are usually generated during weekdays [10].

In terms of textual features, there are also several different approaches to represent the properties, such as bag-of-words (n-gram) [11], emotional word lists (e.g., Negative or positive words) [12], and part-of-speech tagging [13]. Earlier studies started with logistic regression and K Nearest Neighbours (KNN) [6], yet the performance was limited and had crucial weaknesses [14]. Moreover, not dealing with high-dimensional/sparse data effectively was inadequate as online reviews are usually relatively short compared to articles and news.

New trends in fake review detection are based on Support Vector Machine (SVM), and neural networks, which could be seen in most of the recent studies (add citations for this) [15, 16, 17]. Using a real-life dataset from Yelp, [15] applied SVM with different features. They obtained 84% and 86% accuracy using textual features and behavioural features, respectively. With the hotel reviews dataset, [16] took N-gram as their features, where they achieved 90% accuracy with SVM. Finally, deep feedforward neural network (DFFNN) and convolution neural network (CNN) were carried out by [17]. Based on four datasets (hotel, restaurant, doctor, and Amazon reviews), they extracted n-gram and emotional words. The SVM methods they applied scored 76.25% – 85.31% accuracy, while the neural networks scored 81.6% – 89.8% accuracy.

In pursuit of generating a human-like deceptive review dataset, we look to fine-tune pre-trained large language models. Fine-tuning these models instead of training from scratch is almost the standard approach [18] as it is efficient, cheaper, and leads to higher quality generation [19]. Examples like the GPT family have seen tremendous popularity in multiple domains of NLP, from generating patent claims [20] to question-answer chatbot systems [21], due to their impressive ability to generate high quality, coherent text that is almost indistinguishable to human-generated data. Causal language modelling is the name given to this next word prediction given causal conditioning on all previously seen sentence tokens. Following such NLP literature, we choose GPT-2 as our base generation model as it is more powerful than its predecessor, GPT, and it is openly available to the public, which the latest GPT-3 is not.

3 Methodology

3.1 Data

We use two source datasets; the "Gold" dataset and a sampled Amazon dataset. We choose the former as many studies used the 'Gold' dataset, derived from TripAdvisor reviews, in which the authors paid human participants via Amazon MTurk, a platform for buying services from others, to write deceptive reviews for classification [22] [23]. The Amazon dataset contains reviews collected across multiple product categories, with labels of deceptive and non-deceptive provided by Amazon's filtering system [17]. We choose this as Amazon is one of the largest e-commerce platforms, making it a lucrative and representational data source for classification and generation.

3.2 GPT-2 Causal Language Modelling

We wish to build a GPT-2 based review generation model under two simple motivations: firstly, as outlined previously, there is not enough ethically useable data out there to train great review classifiers. Literature showed that many studies used the same 'Gold' dataset. This meant a minimal, niche training set - one that was expensive, inefficient at a large scale, and could lead to unrepresentational learning by classifiers, crippling

generalisation to other platforms. The work of a deep learning-based model alleviates many of these restrictions - there are many genuine reviews on the internet, and with this model, we could generate as many deceptive reviews in a much quicker and inexpensive manner, enabling others to train better models potentially. Secondly, it is of interest to see how powerful these large language models are at generating coherent text that matches those of actual human reviews. Many fake review generators are algorithmic or template-based, where we wish to illustrate the effectiveness and evasiveness of model-based generation.

3.2.1 Training and Generation Pipeline

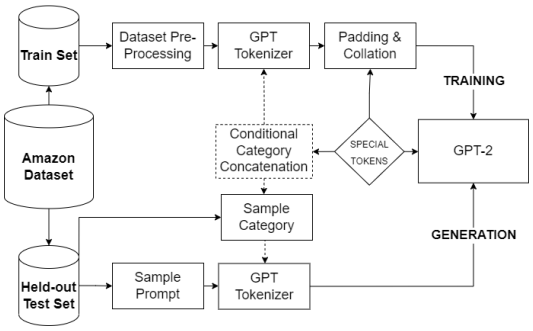

Figure 1: GPT-2 Training and Generation Pipeline with optional conditional modelling.

The training and generation pipeline is outlined in Figure 1. Due to the independent nature of reviews, i.e., $r_1 \not\perp r_2$, we can process reviews independently, simplifying our training process. We only use genuine human reviews from the Amazon dataset for training, as we wish to mimic their authenticity. The dataset is first pre-processed from its raw form to remove undesired attributes such as emojis and non-English characters. In order to feed GPT-2 the data, it must first be tokenized under specification to the input rules of the model, such as concatenating a beginning-of-sentence (BOS) and end-of-sentence (EOS) token to sequences. HuggingFace library provides a GPT-2 tokenizer to transform our inputs this way. The review text tokens are then padded and batched through the data collator with an optional concatenation of the respective review's product category if category conditioning is required. Special tokens are introduced to enable such conditioning, and we define a custom tokenizer for this process. Finally, we fine-tune GPT-2 on the review dataset. Full implementation details can be found in Section 4.1.

Within the dataset generation pipeline, we prompt under reviews from the held-out test set, such that we allow GPT to better generalise by generating unseen data. In order to generate reviews, GPT-2 requires what is known as a prompt. The prompt acts as a sentence starter, in which GPT-2 will then finish the sentence following a maximum likelihood-based decoding strategy. To generate the final dataset, review generation followed two sampling methods. One way, denoted as **non-guided**, samples one random category from the product categories and one random generic English sentence starter word such as {'The,' 'A,'...} and allows GPT-2 to complete the review freely. The second method denoted **guided**, samples a random review from the test-set; its category and a random number of tokens from the start of the review are used to construct the prompt. We hope this may lead to more interesting reviews as the Amazon dataset contains bizarre phrases and unique sentence starters, such as ***"Asura's Wrath is a button masher action game..."***.

3.3 Evaluation of Outputs

Under this generation we wish to aim for three quality metrics when evaluating our generated reviews:

- A realistic level of coherency within generated text.
- Little repetition in generated text.
- Reviews should follow themes of a product category.

Each of these is associated with a different process within the training pipeline. Coherency is enabled via good textual inputs into the model. Repetition control is tuned by identifying the best decoding strategies for the model. However, GPT-2 is a large model and can easily drift from the goal of review generation while generating longer text sequences. We look to tackle this by evaluating the use of conditional product category modeling [24].

3.4 Review Classification

To generate bot reviews that could deceive the classifier, text-based features were used. Non-textual features in the dataset were discarded as the GPT model can only generate text. As previous studies have suggested, we employ Naïve Bayes, SVM, and regression models for baselines. Furthermore, we experiment with various text vectorisation techniques, such as N-gram, TF-IDF, and ELMo, along with a different text classifier, BERT.

We set up two experiments, where M is a classifier model:

- M (*Amazon_real* + *Amazon_fake*)
- M (*Amazon_real* + *GPT-2 generated fake*)

We consider that GPT-2 generated reviews are fake as humans do not write them. So, first, we train and test our model on the official Amazon dataset we have. Next, we combine the genuine reviews from the Amazon dataset with the generated reviews to deceive a fake review classifier. By doing so, we can evaluate GPT-2's susceptibility by comparing the test accuracy of the model resulting from the two separate experiments. For reproducibility, hyperparameters for each model are described in section. 4.2.

4 Experimental Setup and Details

4.1 GPT-2 Training and Generation Setup

4.1.1 Environment and Model Setup

OpenAI has made public four versions of the GPT-2 model: a 124M, 355M, 774M, and a 1.5B parameter version, each of which was trained and evaluated against WebText, a heterogeneous dataset of text content from 45M web pages [25]. Each GPT-2 version achieves a lower perplexity (noramlised inverse probability) on the evaluation set than its predecessor. Minimising perplexity also means maximising the probability of predicted tokens, meaning it generates unseen tokens accurately [26]. Due to computation limitations of our training environment, we trade-off perplexity for model speed and use a distilled version of GPT-2 through HuggingFace, an open-source library for NLP modelling [27]. DistilGPT2 only has 82M parameters, with six layers of 768 dimensions and 12 multi-attention heads, giving a 2x improvement in speed over the large GPT-2 model. We find this trade-off is necessary for our work, and on the WikiText-103 benchmark, DistilGPT2 still scores a good perplexity of 21.1 compared to the 16.3 of the large GPT-2 model [28]. For training we utilised Google Colab [29], a GPU-accelerated cloud-based Jupyter notebook environment.

4.1.2 Training Details

Basic training setup

For each experiment involving training GPT-2, the HuggingFace Transformers library was used for loading and training the models and tokenizing and treating inputs to align with what GPT-2 expects. Further to this, Pandas and PyTorch were used to load, group, and convert our raw-text data into the manageable Torch Dataset format for more straightforward dataset manipulation.

We explicitly introduce three special tokens following [24]'s work on conditional prompting of GPT, which saw the addition of the unknown (UNK), padding (PAD), and separator (SEP) tokens. Before training, we shuffle and split the data into a train and validation set using an 80/20 split. A custom Torch Dataset class was built to tokenize and pre-pad our inputs. GPT-2 has a maximum input sequence length of 1024 tokens. To ensure independence between samples during training, the Dataset class pads all inputs to this max length so that the DataCollator, later on, does not attempt to group overflowing reviews, which we noted led to incoherent generation issues later on. If category conditioning is enabled, the Dataset class also concatenates the product review's category to the input via the SEP special token. The tokenized inputs form a sequence of vocabulary 'input_ids' and a corresponding attention mask. The sequence has its output token labels set to the corresponding token itself. GPT-2 will shift these tokens during training under casual modelling, enabling learning over next token prediction. All other model configurations were inherited from the DistilGPT2 default configurations via HuggingFace's AutoConfig module. A per-device batch size of 4 and a weight decay of 0.01 per HuggingFace's recommendations. The model is trained, and perplexity on the validation set is measured to complete training.

Effect of learning rate, epoch count and layer freezing on generation

To identify the best training parameters, we train GPT-2 on a range of learning rates (LR); at 2e-4, 2e-5, and finally at 2e-6 for six epochs to observe how much training is required before we observe review-like generation, and also to identify the optimal epoch count. Next, fine-tuning follows two routes: one way is to train all model layers, while the second way is to freeze all model layers but a few top layers. We compare all-layer to [24] and unfreeze the top-6 layers of GPT-2. The intuition of the second method follows that we wish to preserve GPT's powerful ability to generate coherent and natural-sounding language, in which the base layers act to preserve the majority of the original model's weights. Both methods were evaluated against our metrics, and the better method was selected.

Review-text only training vs category-conditioned training
For this experiment, we wish to identify how best to generate reviews from GPT-2. We experiment with generation under training only over the review text itself and generation where we concatenate the review category to the review text, denoted as conditional generation. This forces the model to associate specific categories with heuristics from the category's reviews. For example, the "Books" category contains references to keywords like read and learn, in which we want to appear in reviews such that reviews sound unique to their category under our evaluation metrics. We compare both models by feeding selected prompts and observing the quality of generated text under these distinct regimes.

4.1.3 Decoding Experiments

It is known that following different decoding strategies, the language model can produce very different generations. GPT-2 uses **auto-regressive** language generation. This means the probability distribution of a word sequence can be decomposed into a product of conditional next word distributions

$$P(w_{1:T}|W_0) = \prod_{t=1}^{T} P(w_t|w_{1:t-1}, W_0) : w_{1:0} = \emptyset$$

where W_0 is the initial context prompt and T is the determined length of sequence terminating with an EOS token. Given the initial context, the decoding strategy then looks over this distribution to select the most likely future sequence. In this experiment we consider popular decoding strategies such as **Beam Search** [30], **Top-K** [31] and **Top-P** [32] decoding. Each method is trialed with default and tuned parameters, and the quality of generations is evaluated against our metrics, where the best method that generates the most human-like reviews is selected for final dataset generation.

4.1.4 Sampling and Full Dataset Generation

Finally, the fine-tuned GPT-2 model is used to generate the final review dataset under the identified optimal settings. As explained in Section 3.2.1, we consider two sampling techniques for the generation to enable varied generations. First, 10K reviews were generated, with half of the reviews following the **non-guided** method and the other half following the **guided** method. We generate three distinct reviews for each prompt to observe the variety of generations under a single prompt. We impose a minimal review length limit of at least ten tokens for each generation to classify the review as useable, as some generations come out very short. To compare the original human and generated datasets, we plot out distributions over essential lexical characteristics of the data, such as number of characters, number of words, and the average length of words within each review over their respective datasets. This comparison is carried out to verify if the generated dataset follows the heuristics of the human dataset and gain insight into the properties of our final dataset.

4.2 Review Classification

Constructing Baseline Models. We started out by creating a functional baseline classifier. The Gold dataset was first implemented as it should be well-constructed and previous studies showed that the classification result on this dataset was generally satisfying. Seven models were carried out at the beginning, namely Linear Support Vector, Multinomial Naive Bayes, Ridge Regression, Stochastic Gradient Descent, Passive Aggressive, Perceptron, and K-Nearest Neighbors. Vetorization was done by sending the unigrams of the reviews to a TF-IDF (term frequency-inverse document frequency) transformer. After all the models were trained with the Gold dataset and proved working, we further tested them with the Amazon dataset to check the compatibility of our model with real world dataset. Last three models mentioned in the previous paragraph were discarded because of the poor performance.

Improving Word Vectorization The next step of fine-tuning the classifier was to improve the vectorization of our reviews. Instead of using unigrams, we also experimented bigrams, trigrams, and different combinations of various N-grams. A totally different word vectorizaton method called ELMo was introduced to examine the potential of newer vectorizaton algorithm. ELMo converted sentences into 1024 vectors based on the entire sentence, i.e., the same word can have different representations in separated sentences.

BERT Applying BERT is relatively simple as it comes with its own word vectorizer. A sentence of text would be converted into input ids, token type ids, and an attention mask. We set the maximum epochs to five and save the best model after evaluating each epochs.

Examine GPT-2 Generated data To verify if the GPT-2 generated data is similar enough to authentic human written reviews, we create a new training dataset for our classifier by replace the fake reviews in the Amazon dataset with GPT-2 generate data. If GPT-2 generated reviews were relatively more similar to amazon real reviews compared to amazon fake reviews, the new classifier should have a lower accuracy as it would find it more difficult to distinguish between real reviews and GPT-2 generated reviews.

5 Results and Discussion

5.1 GPT-2 Causal Language Modelling

Table 1: Each experimental model end-of-training perplexity on evaluation set.

Model	End-of-Training Perplexity
Default HuggingFace Collation + Padding	45.07
LineByLine Collation and Padding	1.48
LineByLine Collation + Padding + Unfreezing Top-6 Blocks	1.43
LineByLine Collation + Padding + Topic-Modelling + Unfreezing Top-6 Blocks	1.47
LineByLine + Padding + Topic-Modelling	1.48

5.1.1 Basic Training Findings

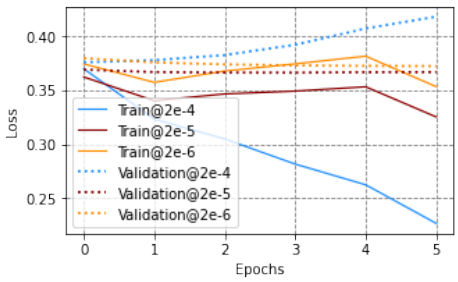

Figure 2: GPT Training and Validation loss per epoch.

Across all training results, we find that GPT-2 converges to validation minima within only a few epochs. Figure 2 shows that the optimal learning rate for fine-tuning is 2e-5, with 2e-4 appearing to diverge in validation loss. All learning rates overfit beyond four epochs, with the optimal epoch count between 3 and 4 cycles. 2e-5 was utilised for training future models. Table 1 notes the end-of-training perplexity for all trained models. We see imposing the independence assumption on our reviews (Line-By-Line Collation) minimises perplexity, however, minimising perplexity does not always translate to human-perfect language [33]. To understand how generations change during training, at each epoch, we generate with prompt **"I"** under default greedy decoder settings where trailing red means the generation entered a repetitive state:

- **1**: I was very pleased with this product. I have been using it for a couple of years now and it is a great product. I have been using it...
- **2**: I love this bag. It's a little bulky and I'm not sure if it's a good size for my bag. I'm not sure if...
- **3**: I have a lot of fun with this product. I have a...
- **4**: I have a lot of fun with this product. I have a...
- **5**: I have been using this for a couple of years and it works great. I have a lot of room for my dogs and they love it. I have a lot of ro...
- **6**: I have been using this for a couple of years and it works great. I have a small dog and I love it. I have a dog that loves it and it is a great addition to my family.

Interestingly, the most coherent generation occurs after just one epoch of training. The first five epochs exhibited repetition, which increased severity around epochs 3-4. The final epoch does not repeat at all but lacks some coherency when referencing the dog and oneself while the most creative reviews seem to be created at epochs 1, 5, and 6. When fine-tuning the frozen and unfrozen GPT-2 models, Table 1 shows the lowest perplexity was exhibited under the frozen models, and these models tended to not overfit as easily during training. However, under the same generation test, we note that the coherency of the frozen models was worse than the unfrozen ones which could be due to top layers parameters updating towards a preference over review-based tokens while the bottom layers preserved general text generation. Appendix 7.1 shows examples between the two. With these priors, further experiments look to address issues that will aid in better text generation, and moving forward, models were trained unfrozen.

5.1.2 Decoding Strategy

From our results we note that the more interesting generations, as expected, were derived under sampling over the next word distribution via a mixture of Top-K and Top-P with $K = 30$, $P = 0.7$ and a repetition penalty of 1.5. This is expected as beam search simply maximises the next sequence likelihood with no variety and tends to lead to boring, but coherent reviews. The mixture of Top-K restricts the sampling space to disregard very low probabilities beyond K while also using Top-P to adaptively sample up till the threshold P and hence fixes the issue of generation over very sharp next word distributions. However, this led to quite a lot of incoherency towards end of sequence generations and some repetition. Scoring based on our

evaluation metrics, beam-search, although sometimes boring, produced very coherent text and with a repetition penalty set to 5, we note the reviews look very human-like. Appendix 7.2 shows examples across all strategies. Beam-search with 5 beams was used going forward.

5.1.3 Category-Conditioned Generation vs Open Generation

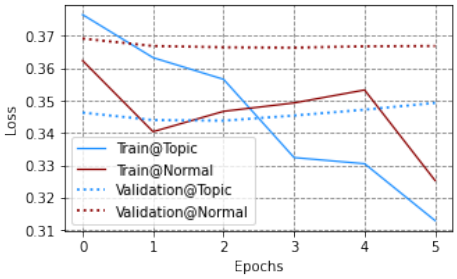

Figure 3: GPT Training and Validation loss for normal and conditional models.

```
Prompt: I love this....

I love this.  I have been using it for over a year now and am very pleased with the
quality of the product.

I love this.  I have been using it for a couple of years now and am very happy with it.

I love this.  It is a little small and easy to put together.  I am very happy with it.
```

```
Category: Watches
Prompt: <|BOS|>Watches<|SEP|>I love this....

I love this.  I wear it every day and am very pleased with the quality of the watch.

I love this.  I bought it to wear as a dress-up for my husband and he loves it.  It is
very well made and looks great on him.

I love this. I have a large wrist and can wear it all the time. It is very easy to put
together, perfect for everyday use.
```

Figure 4: Open generation vs category-conditioned generations {Prompt: **I love this.**, Category: **Watches**}

Figure 3 shows the conditional model reached a lower validation and training loss, but from Table 1, we see they evaluate to the same perplexity. Figure 4 shows examples from generations under the optimal decoding strategy for open and category-conditioned generation. The prompt used was **"I love this."** and the category is chosen **Watches**. All texts appear coherent. We see that open-generation leads to shorter reviews that are not repetitive internally but are across multiple generations. The reviews are not particularly exciting either. The category models incorporate the category very well, with category-related keywords induced in the generation. The category model tends to also produce longer, more interesting reviews with no repetition across multiple generations. Once again, we show that perplexity does not always mean better. The category model was used moving forward.

5.1.4 Review Generation and Sampling

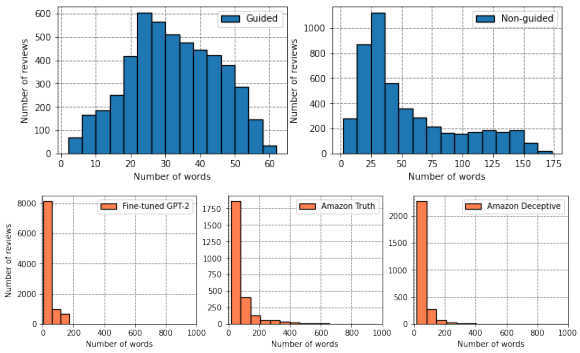

Figure 5: Number of words per review between GPT-2 Guided vs Non-Guided (top) and GPT-2 vs Amazon dataset (bottom).

Generating the fake review dataset under the final GPT-2 review model, we obtain interesting results for the lexical distributions. For example, Figure 5 show that the guided reviews seem to have lower variance in the number of words in comparison to the non-guided reviews. Furthermore, the distribution is spread out well, suggesting the guided reviews follow a more template-based format. On the other hand, the non-guided reviews primarily exhibit fewer words on average but can generate longer sequences. We compares the number of words distribution between the generated set and the Amazon dataset. The distribution across both datasets is similar, with most review lengths between 25-75 words. The original dataset has a few longer review sequences that the generated does not exhibit. We note that the deceptive dataset appears to be a very sharp distribution, suggesting that Amazon's flagging system picked these up as potential template-based reviews. The average word length per review was found to be near-identical across both datasets (see Appendix 7.3). These lexical results suggest that the GPT-2 generated dataset closely aligns with the original and may form a good substitute for deceiving classifiers that work off of textual attributes of the input.

We also note that although overall coherency is high, the generated reviews are not perfect. There is mild repetition among different generations and some coherency issues nearing the ends of reviews, which we attribute to the messy sentiment exhibited by the realistic nature of our train set. Appendix 7.4 shows examples from the final dataset. *The generated review dataset can be found in gpt_generated_dataset.csv.*

5.2 Review Classification

After a few experiments, we decided to abandon the gold dataset, which is too different from the real-world scenario. Both labels' reviews had patterns that the classifiers could easily pick up. Table 2 showed how easy the gold dataset could be classified compared to the Amazon dataset. A worth mentioning finding is that a classifier trained by a specific dataset could only classify reviews from the same dataset. If we take the 91% accuracy model trained by the Gold dataset and perform classification on the Amazon test set, the performance will drop to 60%, which is worse than the model trained by the Amazon dataset itself.

Table 2: In-domain classification.

Dataset\ Model	**SVM**	**Naive Bayes**	**Ridge Regression**	**SGD**	**BERT**
Gold	91%	73%	90%	89%	64%
Amazon	67%	66%	67%	66%	61%

In terms of classification methods, basic machine learning techniques outperformed BERT as Table 2 demonstrated. A possible explanation for this was that the strength of BERT was understanding the meaning of the content but not picking up the language pattern. For instance, the meaning of a real review could be identical to a fake review, yet, the word choice might be different. Breaking down the reviews using N-gram could not fully explain the meaning of the content, but it might discover word patterns that would be more useful when classifying real-fake reviews.

Another key factor that affects the performance is word vectorization. ELMo was similar to BERT as they both focused on understanding the meaning of the text, which could be unnecessary in real-fake review classification. Not surprisingly, the classifier trained with Elmo vectorised Amazon dataset barely improved, with a maximum accuracy of 66%. Finally, we had different combinations of N-grams. Unigram contained a lot of information for real-fake classification as word frequency was one main feature that distinguishes the two classes. Moreover, Bigram provided additional information about term frequency which was also applicable. However, Trigram introduced noise as three-word-long terms in a sentence usually do not provide any useful information. After examining different combinations of N-grams, the best mixture we found was to combine unigrams with bigrams, which usually had higher accuracy than other combinations.

The first classification experiment was to test the performance of our working classifiers with the real-world dataset, the Amazon dataset, which got a maximum accuracy of 67%. The second classification experiment examined the quality of the GPT-2 generated dataset and whether it could deceive our classifier. A new dataset was created by replacing fake reviews in the Amazon dataset with GPT-2 generated reviews, and a new classifier was trained on this new dataset. Whether the replacement GPT-2 generated reviews were guided or non-guided, the classifier had a hard time classifying the reviews. The accuracy dropped to 50% for all models, meaning that the GPT-2 generated reviews perfectly blended in the real Amazon reviews, and the classifier could only perform random guesses.

6 Conclusion

Overall, we find that the much-used 'Gold' dataset does lead to good classification over the dataset itself, scoring 91% accuracy, however the model did not generalise well to reviews taken from Amazon (67% accuracy) and was easily susceptible to the GPT-2 dataset (61% accuracy). This may be due to the Gold dataset's poor representation of other review modalities across different product types. Our GPT-2 review generator performs very well in creating deceptive reviews that follow any chosen product category. The final generated dataset was very human-like and met most evaluation metrics. However, it was not a perfect generation as GPT-2 could stray from the theme of the review towards the end of long generations, leading to decreased coherency even under categorical conditioning.
Future work could implement decoding strategies beyond maximum likelihood for interesting generations. Humans communicate surprisingly, not in a 'most likely' fashion. Typical decoding [34] can fix this as words are sampled based on their relative information content. We could also train GPT-2 on a much large review corpus for a deeper breadth to review generations, and further analyse the emotional content of reviews between the datasets to incorporate as a feature for classification. We note such work may lead to malicious detection evasion of deceptive reviews, and as such the ethics surrounding the concept must be examined carefully, however, we believe our work is an essential step into introducing large, labelled datasets for the genre of review classification to alleviate concerns.

References

[1] Gaubys, j. (n.d.). global ecommerce sales growth (2019–2025). https://www.oberlo.co.uk/statistics/global-ecommerce-sales-growth [Online; accessed 31-March-2022].

[2] Yi Jin Lim, Abdullah Osman, Shahrul Nizam Salahuddin, Abdul Rahim Romle, and Safizal Abdullah. Factors influencing online shopping behavior: The mediating role of purchase intention. *Procedia Economics and Finance*, 35:401–410, 2016. 7th International Economics Business Management Conference (IEBMC 2015).

[3] Fake online reviews are a $152 billion problem - here's how to silence them. https://www.weforum.org/agenda/2021/08/fake-online-reviews-are-a-152-billion-problem-heres-how-to-silence-them [Online; accessed 31-March-2022].

[4] Saleh Nagi Alsubari, Sachin N. Deshmukh, Ahmed Abdullah Alqarni, Nizar Alsharif, Theyazn H. H. Aldhyani, Fawaz Waselallah Alsaade, and Osamah I. Khalaf. Data analytics for the identification of fake reviews using supervised learning. *Computers, Materials &amp Continua*, 70(2):3189–3204, 2022.

[5] Atefeh Heydari, Mohammad ali Tavakoli, Naomie Salim, and Zahra Heydari. Detection of review spam: A survey. *Expert Systems with Applications*, 42(7):3634–3642, May 2015.

[6] Nitin Jindal and Bing Liu. Analyzing and detecting review spam. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, October 2007.

[7] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review spam. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, page 2488–2493. AAAI Press, 2011.

[8] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*. ACM Press, 2010.

[9] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018.

[10] Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, and Jidong Shao. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. *Proceedings of the International AAAI Conference on Web and Social Media*, 9(1):634–637, Aug. 2021.

[11] Nidhi A. Patel and Rakesh Patel. A survey on fake review detection using machine learning techniques. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, December 2018.

[12] Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2014.

[13] Luyang Li, Bing Qin, Wenjing Ren, and Ting Liu. Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, 254:33–41, September 2017.

[14] Aliaksandr Barushka and Petr Hajek. Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Applied Intelligence*, 48(10):3538–3556, March 2018.

[15] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. What yelp fake review filter might be doing? *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1):409–418, Aug. 2021.

[16] Hadeer Ahmed, Issa Traore, and Sherif Saad. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9, December 2017.

[17] Petr Hajek, Aliaksandr Barushka, and Michal Munk. Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications*, 32(23):17259–17274, February 2020.

[18] Bonan Min, Hayley Ross, Elior Sulem, Amir Veyseh, Thien Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey, 11 2021.

[19] XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, sep 2020.

[20] Jieh-Sheng Lee and Jieh Hsiang. Patent claim generation by fine-tuning openai gpt-2, 06 2019.

[21] Xiangyang Mou, Chenghao Yang, Mo Yu, Bingsheng Yao, Xiaoxiao Guo, Saloni Potdar, and Hui Su. Narrative Question Answering with Cutting-Edge Open-Domain QA Techniques: A Comprehensive Study. *Transactions of the Association for Computational Linguistics*, 9:1032–1046, 09 2021.

[22] Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon gyo Jung, and Bernard J. Jansen. Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services*, 64:102771, 2022.

[23] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. 2011.

[24] Conditional text generation by fine tuning gpt-2, 2022. https://www.ivanlai.project-ds.net/post/conditional-text-generation-by-fine-tuning-gpt-2 [Online; accessed 31-March-2022].

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

[26] Wikipedia contributors. Perplexity — Wikipedia, the free encyclopedia, 2022. [Online; accessed 31-March-2022].

[27] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2019.

[28] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[29] Google colab - cloud computing platform. https://colab.research.google.com [Online; accessed 31-March-2022].

[30] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models, 2016.

[31] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation, 2018.

[32] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2019.

[33] Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. Lower perplexity is not always human-like, 2021.

[34] Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Typical decoding for natural language generation, 2022.

7 Appendix

7.1 Appendix A - Examples of Unfrozen vs Frozen Generations

7.1.1 Unfrozen

This is a great product. I have been using it for about 3 months now and love the color! I am very happy with this purchase.

7.1.2 Frozen

This is a great product. It's very easy to use and the colors are beautiful, but I'm not sure if it will last for many years or even decades. The only thing that annoys me about this item was how long you can hold one in place.

7.2 Appendix B - Examples of Top-K, Top-P and Beam Search Generations

The used category and prompt for all was "**Beauty**" and "**This**". Only the first generation was used from each.

7.2.1 Top-K @50

This was great! I got the size I wanted, and it's perfect, just have to wear it in the dark. The collar feels snug and the bottom piece is very well made, but the quality of the jewelry is not what you'd expect for real jewelry. I am so glad to say I made my own wedding band...

7.2.2 Top-P @0.94

This necklace was for my granddaughter. I am so glad I purchased it. I think it was a good value. It was very comfortable for my 7 year old granddaughter and it definitely fits her as well.

7.2.3 Top-K @50 and Top-P @0.94

This is the perfect size for a small purse and I love it!!! My daughter loves it, is just too thin, and is almost as slim as she thought it would be. It's super cute and the color is great.

7.2.4 Beam-Search, 5-Beams

This is my second purchase from this company. I am so happy with it. I have been looking for a different brand of sunscreen in the past, and they seem to work just as well right now. This one does not disappoint at all. It's very moisturizing and doesn't leave a sticky residue on your skin. My only complaint is that you need to wipe off any excess products before putting them on or after showering.

7.3 Appendix C - Further Lexical Analysis Graphs

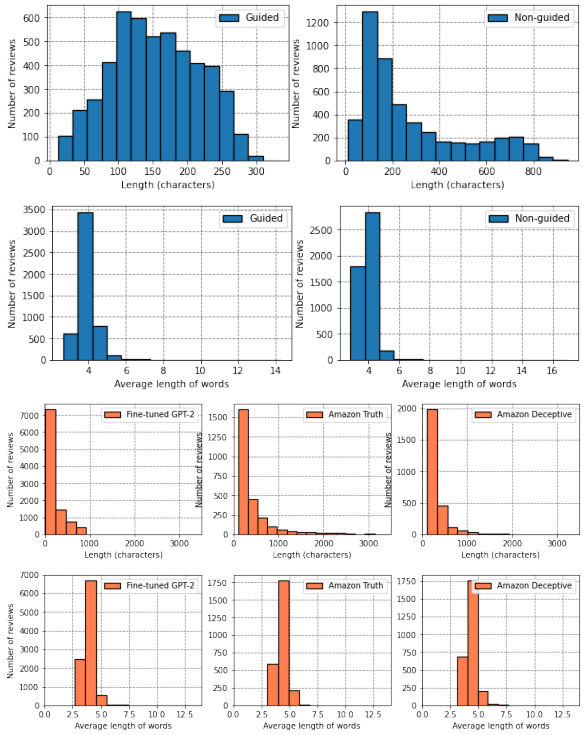

Figure 6: Length of review (characters) and average length of words per review between GPT-2 guided vs non-guided (top) and GPT-2 whole vs Amazon dataset (bottom).

7.4 Appendix D - Samples From Final Generated Dataset

These were the first 3 samples taken from the final shuffled generated with no cherry-picking of results:

- **Category:** Automotive
  Sample Type: Guided
  ”Fits well on my Subaru Crosstrek. My only gripe is that it’s hard to find a place in the back seat, so I don’t have much room for it. I’ve owned several Crosstrek sedans before and this one has been great.”

- **Category:** Electronics
  Sample Type: Non-Guided
  ”If you are looking for a sound system that is easy to set up, this may not be the case. You will need an external power supply or something like that. I have tried other similar products and they seem to work just fine.”

- **Category:** Health & Personal Care
  Sample Type: Guided
  ”This set is great for those looking to get a lot out of the box. It doesn’t have all the bells and whistles, but it’s still very functional.”