International Journal of Data Science and Analytics (2022) 13:363–383
https://doi.org/10.1007/s41060-021-00299-5

REGULAR PAPER

Detecting computer-generated disinformation

Harald Stiff[1] · Fredrik Johansson[1]

Received: 30 March 2021 / Accepted: 3 December 2021 / Published online: 23 December 2021

Abstract

Modern neural language models can be used by malicious actors to automatically produce textual content looking as it has been written by genuine human users. Due to progress in the controllability of computer-generated text, there is a risk that state-sponsored actors may start using such methods for conducting large-scale information operations. Various detection algorithms have been suggested in the research literature to identify texts produced by language model-based generators, but these are often mainly evaluated on test data from the same distribution as they have been trained on. We evaluate promising Transformer-based detection algorithms in a large variety of experiments involving both in-distribution and out-of-distribution test data, as well as evaluation on more realistic in-the-wild data. It is shown that the generalizability of the detectors can be questioned, especially when applied to short social media posts. Moreover, the best performing (RoBERTa-based) detector is shown to be non-robust also to basic adversarial attacks, illustrating how easy it is for malicious actors to avoid detection by the current state-of-the-art detection algorithms.

Keywords Computer-generated text · Detection algorithms · Information operations · Language models

1 Introduction

Progress in the research field of neural language generation has in recent years resulted in a variety of generative models able to produce texts of high quality. Current state-of-the-art generative models for text, also referred to as neural language models, produce textual output that often is so grammatically correct, fluent, and coherent that it is hard to tell apart from text written by humans [5,48]. As with many other technologies, there are several ways in which such models can be used for malicious purposes. Examples of use cases include targeted bot attacks [47], fake news generation [43,48], and fake reviews generation [1].

An even more worrying threat is the potential weaponization of such techniques by state actors and state-sponsored groups [5]. Open societies are already today challenged with malicious actors deliberately creating and spreading disinformation in social media for everything from economical gain to increasing divide and sowing distrust in the political system in democratic countries, using a combination of bots, sockpuppets, and hijacked accounts [4]. Although it is hard to estimate the real-world impact of such online information operations, it is clear that some actors spend a considerable amount of money to orchestrate the spreading of lies and disinformation that follows specific narratives of interest [3,13]. Hence, there is more than a hypothetical risk that malicious actors will attempt to make use of generative models of various modality, including the high-quality texts generated by powerful language models such as GPT-2 [35] and Grover [48]. Potentially, this will lower the existing barriers for state actors and other malicious users to efficiently produce misinformation [5] in the social media landscape.

Although there lately has been some reports of language models used to abuse governmental process [47] and creating blog posts more or less automatically [24], there has so far not been any reports of state actors actively using language models for information operation purposes. One explanation for this may be that it until recently has been complicated to control language models to follow a specific narrative, while at the same time keep producing varied text of high quality. However, controlling language models has become a popular research topic, resulting in several new ways to better steer what is generated by a language model, in addition to previous coarser methods like fine-tuning and priming. Examples

✉ Fredrik Johansson
fredrik.johansson@foi.se

Harald Stiff
harald.stiff@foi.se

[1] Swedish Defence Research Agency (FOI), Stockholm, Sweden

of such methods include adding additional metadata to the training [28,48], and the use of attribute classifiers [12,29] for guiding the text generation process. As noted by Brown et al. [5], this may lead to an increased future risk of language models being misused by, e.g., state-sponsored groups. For this reason, it is of interest to find out to which degree a malicious actor can achieve control of the content being generated using such techniques. Furthermore, it is becoming increasingly important to investigate to which extent existing detection methods suggested in the research literature can be used to distinguish between human-generated text and text being produced using neural language models, especially as there exists a growing amount of research indicating that this task is very challenging for humans [1,20,26,48]. Initial research on the large-scale state-of-the-art neural generative model GPT-3 even suggests that human abilities to distinguish between real texts and texts generated by the largest GPT-3 models are not better than random guessing [5].

The main contribution of the work presented in this article is that we evaluate the performance of promising machine learning-based detection models suggested in the available research literature on a wide variety of datasets, covering several types of texts, including news articles, product reviews, forum posts, and tweets. The texts are generated using existing language models such as GPT-2 and Grover, while more fine-grained control of the topic of the generated texts is achieved using the control mechanisms PPLM and GeDi. Such control of the generated texts is likely to be utilized by actors wanting to misuse language models for information operation purposes. The generalizability of the detection methods is studied in both in-distribution and out-of-distribution experiments, as well as on in-the-wild data. Lastly, the detectors' robustness toward adversarial attacks is investigated.

All in all, it is shown that detectors based on RoBERTa [30] demonstrate reasonable generalizability to out-of-distribution data, but that the detectors are not accurate enough for practical use, other than in constrained scenarios where pretrained generative models are likely to be used out of the box. Furthermore, active countermeasures such as the use of adversarial attacks can cause the detection algorithms to perform worse than random guessing. This calls for future research into more robust detection methods than are available today.

The rest of this article is structured as follows. In Sect. 2, it is explained how neural language models work, how they are trained, and how they can be used by attackers to automatically generate novel text content on a specific topic and with a specific sentiment. In Sect. 3, various detection algorithms suggested in the existing research literature for distinguishing between real and computer-generated text are reviewed, together with related work on bot detection and adversarial attacks. The actual task definition studied in this work is specified in Sect. 4. Next, the experimental setup is described in Sect. 5, including datasets used for training or fine-tuning the detectors, as well as evaluating their performance on in-distribution, out-of-distribution, and in-the-wild data produced by various neural language models being controlled in different ways. The obtained results are presented and analyzed in Sect. 6, while their potential implications are discussed in Sect. 7, together with ideas for future work. Finally, conclusions are presented in Sect. 8.

2 Neural text generation

The idea of text generation methods is not new. For example, various n-gram language models have been around for a long time [34]. Earlier text generation methods usually relied on extracting and storing statistical frequencies from large text corpora, and used these to estimate probability distributions from which new text sequences could be sampled. However, text produced by such models tends to be ungrammatical and incoherent [18], and hence, easy for humans to tell apart from its real human-generated counterpart. Neural RNN-based language models [40] took a step forward in terms of quality of the generated text, but the quality reached another level when large-scale language models based on the Transformer architecture [44] were introduced. Unlike RNNs which have to process data sequentially, Transformer models allow for significantly better parallelization thanks to their attention mechanism which allows them to selectively focus on segments of input text they predict to be the most relevant. Since Transformer-based language models such as GPT-2 [35] require large quantities of text and compute to train, much of their popularity has, at least until recently, been relying on being trained and publicly released by large companies or research organizations. However, it has become easier for other actors to both fine-tune and train such models from scratch due to factors such as release of open-source code, developments of new hardware accelerators, and new research on how to fine-tune existing language models to other languages. Hence, although this is not feasible for the average user, it is without doubt accomplishable for state actors under the premise that they can find large enough representative datasets for the domain and language for which they are interested in generating text.

2.1 Language modeling

On a high level, neural language models can be described as being trained to predict the next token (such as a word or a word-piece) in a text sequence, given the previous tokens. This is an example of a self-supervised learning task for which no human-annotated texts are required as training data. Instead, only large quantities of unstructured and unlabeled text are required, where tokens can be masked out automatically. As a concrete example of a single training example, the language model can be asked to predict the next word in the text sequence: "Barack Obama is a former," where continuations like "US," "American," or "president" can be expected. More formally, given a corpus of texts $D = \{\mathbf{x}^i\}_{i=1}^{|D|}$, where each text $\mathbf{x}^i$ is composed of a sequence of tokens $(x_1^i, \ldots, x_N^i)$, a left-to-right neural language model P_θ is trained using a language modeling objective to learn the distribution:

$$P(\mathbf{x}) = \prod_{i=1}^{N} P(x_i | x_{<i}). \tag{1}$$

The chain rule decomposition of Eq. 1 follows from how the texts are generated in an autoregressive manner. The parameters θ of the language model P_θ are obtained by optimizing the language modeling loss function:

$$L = - \sum_{\mathbf{x} \in D} \log P_\theta(x_i | x_{<i}). \tag{2}$$

Once a neural language model has been trained, it can be used to estimate the probability of a text sequence, but also to generate new text.

2.2 Generating text

A trained neural language model can be used to create text conditioned on some input, such as the beginning of a sentence, or just an empty start token in the case of unconstrained text generation. New texts can then be generated by sampling tokens repeatedly from the conditional distribution $P_\theta(x_i | x_{<i})$ until an end token is generated or other stopping criteria are fulfilled, such as a pre-specified maximum sequence length being reached. Although, in theory, it is possible to simply greedily select the most probable token at each step, this leads to repetitive and highly non-varied text [25]. Hence, some kind of non-deterministic sampling strategy is needed. One such strategy could be to let each token have a chance of being generated that is directly proportional to its estimated probability, as expressed by the language model. However, this tends to lead to texts that significantly deviate from human-generated text, as the probability distribution often contains a long tail of tokens that individually are assigned low probabilities, but which cumulatively are assigned a high probability mass [27]. It is therefore more common in practice to sample from a truncated part of the probability distribution. One common strategy is top-k sampling [17], where the probability distribution is reassigned to only include the k most probable tokens. Nucleus sampling [25] is a dynamic version of top-k sampling that dynamically truncates the distribution to the smallest set of tokens with a total probability mass reaching above a fixed threshold $p \in [0, 1]$.

```
<|begintitle|>Bots are Flooding the Internet With
Fake Reviews<|endoftitle|><|begindomain|>New
York Times<|endofdomain|<|beginauthor|>John
Smith<|endofauthor|><|begindate|>09-08-2018<|endofdate|>
<|beginarticle|>There are worrying reports of bots...
```

Fig. 1 An example of how GROVER is conditioned on article fields in order to generate a news article. The desired characteristics of the text, e.g., the chosen title, is added as an initial string that GROVER will be conditioned on. The generation begins after the $< |\text{beginarticle}| >$ token

2.2.1 GPT-2

GPT-2 [35] is a Transformer-based language model trained to predict the next token in a sequence as described in Sect. 2.1. It has originally been trained on a dataset (WebText) containing 40 GB of text scraped from the internet. The relatively large training data size and its powerful architecture makes it capable of generating diverse coherent text in a multitude of domains. GPT-2 can easily be adapted to generate text in more restrictive domains (e.g., reviews and social media comments) with additional fine-tuning on datasets several orders of magnitude smaller than the WebText dataset.

2.2.2 GROVER

GROVER [48] is a language model with the same architecture as GPT-2. However, it has been trained on the REALNEWS dataset, containing articles from a broad range of news domains. Unlike GPT-2, it is trained to generate texts conditioned on a $headline$, $date$, $author$, and $domain$, adding the possibility of steering the generated text more closely toward a desired style and topic. When generating an article, the text is initialized with the desired article attributes enclosed in their corresponding start and end tokens as illustrated in Fig. 1, whereafter the rest of the text is generated auto-regressively as described in Sect. 2.2. There are three different model sizes of GROVER, ranging from a 117M parameter (Base) model to the largest 1.5B parameter (Mega) model.

2.3 Controllable text generation

Although the output of neural language models can be controlled to some degree by conditioning (probing) them on an input sequence or fine-tuning them on a more domain-specific dataset, this level of control is in general not enough for a malicious actor who wants to utilize it within the context of information operations. Instead, there are more sophisticated ways in which the generated text can be controlled. One way is to incorporate metadata in the form of control tokens as additional information during the training, so that these later on can be utilized for finer-level control during text generation. Examples of this kind of class-conditional language models are Grover [48] and CTRL [28]. Another, more flexible, way to achieve control over what is being generated is to make use of attribute classifiers. An early variant of this was suggested by Adelani et al. [1]. In their approach, texts generated by a language model were used as input to a separate discriminator model to ensure that all texts with an unwanted sentiment could be discarded. While such a filtering mechanism is not impacting the actual text generation (and thereby may require generating large amounts of texts before producing a text that is in line with what its user wants), more modern approaches incorporate the attribute classifier into the generation process, so that the text generation can be more directly guided. Examples of methods that use such attribute classifiers are Plug and Play Language Models (PPLM) [12] and generative discriminator-guided sequence generators (GeDi) [29].

2.3.1 PPLM

PPLM relies on an external attribute model in addition to a pre-trained neural language model in order to generate text with a desired characteristic. The attribute model is typically implemented as a standard text classifier. This makes it several orders of magnitude smaller than the original language model, but still allow for effective steering of the output [12]. This is achieved by sampling text using the language model and feeding the generated text into the attribute model. This results in a probability of the text to be of the correct class, according to the attribute model. Gradients from the attribute model are utilized in a backward pass that updates the internal latent representations so that a new distribution over the vocabulary can be generated from the updated latent. This process is repeated at every generation step, leading to a gradual transition of the generated text toward the desired attribute.

2.3.2 GeDi

GeDi uses generative discriminators to, with the help of a control code, guide (larger) language models toward generating text with a desired attribute, or alternatively, away from generating text with undesired attributes. GeDi drastically reduces the required computation time per generated sample compared to PPLM [29] (as it unlike PPLM does not require performing multiple forward passes per generation step), but on the other hand is more computationally expensive and difficult to train since it requires training a separate (but smaller) language model using hybrid generative-discriminative training. In essence, GeDi guides the text generation process by at each step efficiently compute classification probabilities for all possible next tokens at once using Bayes rule. This is accomplished by normalizing over two class-conditional distributions, where the first is conditioned on the desired attribute (e.g., positive sentiment) and the other on the undesired attribute (e.g., negative sentiment). The computed likelihoods can then efficiently guide the generation of text from the original (large) language model using various heuristics.

3 Detection models

Detection of text being generated by language models has received increasing attention [1,20,43,48] since the advent of large-scale language models such as GPT-2. However, compared to detection of images [11,31,45,46] and videos [2,7,32] being synthesized or manipulated using generative models, its text counterpart is under-researched.

Among the suggested approaches for predicting whether a text sequence has been machine generated or not, different classes of methods can be identified. Some of these make direct use of the probability distribution expressed by neural language models, while others rely on machine learning-based classifiers trained using supervised learning. Within the first class of methods, the total probability method introduced by Solaiman et al. [39] is a representative example. It simply computes the total probability of the text sequence of interest, based on a pre-trained GPT-2 language model. If the computed probability is closer to the mean likelihood over a set of known machine-generated sequences than the corresponding mean likelihood over a set of human-written texts, the text sequence is classified as machine generated. This idea can easily be expanded upon to also incorporate other pre-trained language models.

A related detection method is GLTR [20]. GLTR relies on that text generation methods tend to sample from a truncated head of the full probability distribution. In addition to calculate the probability of each word in the text sequence of interest according to a pre-trained language model, it also computes its absolute rank of the word. After binning the ranks into a smaller number of buckets, the text can be overlayed with colors corresponding to the chosen buckets. In this way, a human can more easily spot if probable words are being overrepresented in the text sequence. Averages over the calculated values can also be used as input features to shallow classifiers, which has been tested with limited success [26]. Other detectors based on shallow classifiers have also been proposed, such as a baseline logistic regression model representing texts using TF-IDF features on unigram and bigram level [39].

Together with the public release of the largest GPT-2 model (consisting of 1.5B parameters), OpenAI released a sequence classifier based on a pre-trained RoBERTa [30] model, fine-tuned to distinguish between texts being generated from the GPT-2 model and real texts [39]. The detector was trained on 250,000 samples from the WebText dataset [35] and an equal amount of texts synthesized with GPT-2 using a mixture of sampling methods.

In a similar vein, Zellers et al. [48] have proposed adding a linear classification layer on top of their powerful GROVER language model. They argue that the capability of GROVER to generate text also makes it a strong detector. Their largest GROVER-Mega detector has been trained on an equal amount of human-written articles and articles generated by GROVER-Mega, using top-$p = 0.94$. In their experimental results, it is shown to outperform other detectors (including a detector based on BERT [15]), although later work [43] questions the generalizability of using GROVER as a detector when other potential generators are taken into consideration.

Although there is a growing amount of research on detection of text being generated by language models, there is still a lack of understanding of which detection models that perform the best, especially when they have to generalize to data from other distributions than being trained on. In a real-world scenario, a sophisticated attacker is unlikely to generate text straight from a publicly available language model on which a detection model can be trained. Instead, it can be expected that such language models are retrained on other types of (non-public) text data before use, and that some kind of controlled text generation method is used to steer the content of the text being generated. The use of alternative sampling mechanisms, or even adversarial attacks aimed at confusing specific detection models, can also cause the generated text to deviate significantly from what a public language model would generate using the default settings. Hence, there still exists many research questions to address within this area of research.

Although the focus in this article is on detection of computer-generated text, it is highly related to the more well-researched problem of bot detection. Bot detection has been an active field of research for more than a decade [9], i.e., much longer than there have been widespread discussions on the impact of social bots on polarization and spread of disinformation [37]. Early machine learning-based approaches to detecting automation of social media accounts were often based on relatively simple measurements related to posting behavior, posted content, and account properties of individual accounts [6], but such approaches are not working as well today, due to newer generations of bots that are often far more sophisticated than previous generations [10]. As demonstrated in [42], bot detection systems are often not robust enough to generalize to social bot scenarios that are not part of the training data. Moreover, the false positive and false negative rates of such systems on real-world data can be questioned [36]. Graph-based approaches that take coordination and synchronization among groups of accounts into consideration when classifying the accounts are therefore becoming increasingly popular [9]. For coming generations of social bots, it is more than likely that controllable text generation using language models will be utilized for generating the textual content, making the bot detection problem even more challenging than it already is today.

Both bot detection and detection of computer-generated text can be seen as an arms race where improved detectors may cause increasingly sophisticated attack strategies designed for bypassing the defense. For this reason, it becomes relevant to study the developed machine learning-based detection methods' robustness against adversarial attacks, i.e., slight modifications of the input designed to be difficult for the machine learning-based models to classify accurately [22]. To the best of our knowledge, adversarial attacks have not previously been studied in the context of detection of language model-generated text, but the relatively immature research field of adversarial examples is quickly evolving. In white-box attacks in which an attacker has perfect knowledge of the classification model used for detection, it is in many cases a rather straightforward optimization problem to modify input in a way such that a minimal perturbation causes the input to be misclassified by the model, e.g., changing a few pixels in an input image. This can, for example, be accomplished using gradient-based search algorithms such as FGSM [23] or L-BFGS [41]. In black-box scenarios, in which the used detection model is unknown to the attacker, it becomes more challenging to carry out adversarial attacks. However, it has been demonstrated that adversarial examples transfer surprisingly well [41], so that an attack optimized for a substitute model to which the adversary has access is likely to misclassified also by the target detection model unavailable to the attacker. Suggested defenses against adversarial attacks include adversarial training [23,41] and defensive distillation [33], but these defenses may often be broken using black box-attacks or more expensive iterative optimization attacks [22].

For textual input, adversarial attacks are less studied. Such attacks are somewhat different as they have to be carried out on the level of individual characters or words rather than on the level of pixels. While a slight change of the intensity level of a single pixel rarely changes the overall content of an image, adding or changing a single token in a sentence may change its meaning completely. Despite this, various attacks for NLP applications are continuing to emerge in the research literature [49], which emphasizes the need to also evaluate the robustness against adversarial attacks for machine learned-based models aimed at detecting text being generated by language models.

4 Task definition

We assess the performance of promising machine learning-based detection algorithms suggested in the research literature for distinguishing between real and machine-generated texts with respect to:

1. their ability to *generalize* to different domains, generators, and control mechanisms, and
2. the extent to which they are *robust* against adversarial attacks.

The generalizability aspect is important since it in practice is most likely that an adversary who generates text in an information operations context will do this using a method that generates data deviating from what the detection model has been trained on originally. This can, e.g., be a result of the attacker using a new or fine-tuned language model, an alternative sampling strategy, or by steering the generated text toward a specific narrative of interest.

The robustness aspect becomes relevant in situations where the attacker knows that a defender may use machine learning-based detection models to automatically identify use of machine-generated text. If there are publicly available detection models, the attacker may design adversarial examples specifically targeted for being misclassified by these models. If the defender instead uses a non-public detection model, black-box attacks may still be a valid threat. For these reasons, it is of interest to evaluate both the generalizability and robustness of the detectors. The performance of the detectors is evaluated on a binary classification task, i.e., predicting whether individual texts have been machine generated or not.

5 Experimental setup

In this section, the experimental setup used to investigate the generalizability and robustness of promising detectors is described. When evaluating the performance of such detectors, representative data become highly important. Section 5.1 describes the datasets used for evaluation of the models, while Sect. 5.2 presents the actual detection models that have been tested on this data. Section 5.3 describes the methodology for investigating the detection models' generalizability, while Sect. 5.4 describes how the robustness has been evaluated using white-box and black-box adversarial text attacks. In the experiments, machine-generated text is treated as the positive class. The performance is measured in terms of accuracy, precision, recall, and F1-score. A high-level overview of the conducted experiments is given in Fig. 2. Details on the hyperparameters used for each generation strategy are described in further detail in "Appendix A.2."

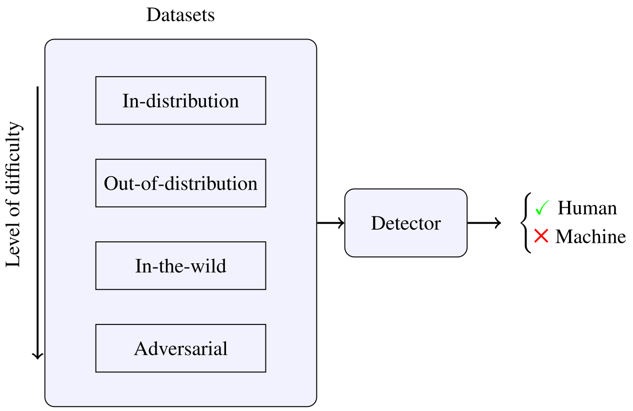


Fig. 2 An illustration of the conducted experiments. The robustness of the detection models is evaluated using four different groups of datasets with increasing levels of difficulty. The evaluation begins with texts generated with nucleus sampling, continuing with out-of-distribution texts and in-the-wild datasets generated with novel models and sampling strategies. Finally, the models are evaluated on the most challenging dataset of adversarial examples that have been optimized to fool the detectors

5.1 Generators and datasets

The experiments have been conducted on two very different types of textual domains: news articles and social media texts. On a finer scale, the social media texts that have been included in this research can be divided into tweets, Reddit comments, Yahoo answers, and Yelp user reviews. For each domain of interest, representative datasets have been required, covering both real human-written texts and language model-generated texts. Further details about these datasets and their generator models are presented below and summarized in Table 1.

5.1.1 News articles

For news articles, it is well known that GROVER language models are able to produce highly realistic articles. For this reason, two GROVER models of different size have been included as language models used to create machine-generated news articles, while data instances from the REALNEWS dataset [48] (originally used for training GROVER) have been utilized as real texts. As GROVER allows for conditioning on metadata relating to *headline*, *domain*, *author*, and *date*, this information was extracted from genuine news articles, sampled randomly from the REALNEWS dataset [48]. The generated text was thereafter sampled auto-regressively from the generator, conditioned on the sampled metadata.

Table 1 Generator models used to synthesize the different texts

Model	#Params	Pre-training data	Fine-tuning data
GROVER-Base	117M	REALNEWS	
GROVER-Mega	1.5B	REALNEWS	
GPT-2 Go Emotions	345M	WebText	GoEmotions
GPT-2 Sentiment140	345M	WebText	Sentiment140
GPT-2 Yahoo Answers	345M	WebText	Yahoo Answers
GPT-2 Yelp Polarity	345M	WebText	Yelp Polarity

5.1.2 Social media texts

The GPT-2 language model has (unlike GROVER) partly been pre-trained on social media data, and is therefore better suited for generating such data. We fine-tune four separate generative models, all based on a pre-trained medium-size version of GPT-2. The fine-tuning was performed on the following social media datasets, from which we also have extracted the real social media texts:

- **Sentiment140** [21]: A dataset of Twitter posts originally created for sentiment analysis. Fine-tuning was carried out for one epoch on all of the 1,599,502 texts belonging to the training split of the dataset.
- **GoEmotions** [14]: A dataset containing Reddit comments, originally used for fine-grained emotion classification. All of the 43,410 comments in the training split were used for fine-tuning the GPT-2 model for ten epochs.
- **Yahoo! Answers (nfL6)** [8]: A dataset of 87,362 questions and their corresponding answers. The first 82,363 answers of the dataset were used to fine-tune a pre-trained GPT-2 model for ten epochs. None of the questions in the dataset were used.
- **Yelp Polarity Reviews** [50]: A dataset containing an equal number of positive and negative Yelp reviews. The GPT-2 model was fine-tuned for one epoch on the training split containing 560,000 reviews.

The texts from each dataset that were not used for training the generators were later utilized as real texts when evaluating the various detectors. Each dataset has been obtained from Huggingface Datasets.[1]

5.1.3 Controlled text generation

While the data described so far cover the different domains and generators used in the experiments, some further complexity arises when taking the controllability into account. As explained earlier, an attacker may want to be able to control the content of what is being generated, which potentially can have an impact on the detectors' performance. In addition to unconditioned text generation (and conditioning on sampled metadata for the GROVER generator) as described above, we have also controlled a subset of the generated news articles and social media texts on a more fine-grained level using PPLM and GeDi.

For PPLM, two different attribute models were used. For the first attribute model, a simple bag-of-words (BoW) model was utilized. A list of military-related terms[2] was used to represent a military topic. According to this straightforward model, the likelihood of a text containing a military topic is given by the sum of likelihoods of each word in the bag. As the second (slightly more complex) attribute model, a single linear layer was trained on top of the last hidden state of each generator model on the task of classifying sentiment (based on data from the Stanford Sentiment Treebank [38]). Once trained, gradients from the attribute models were used to steer the generated texts to be (1) positive or negative, or (2) military-related, respectively, while simultaneously taking gradient steps in the direction of high likelihood as expressed by the underlying text generation model.

For GeDi, the generated text was also steered toward a specific sentiment (negative) or topic (food). For this purpose, pre-trained generative discriminators[3] were utilized. The parameters used for steering the generation using PPLM and GeDi are found in "Appendix A.2."

5.2 Detection models

Although many different types of detection models have been suggested in the research literature for the task of discriminating between real and machine-generated texts (as described in more detail in Sect. 3), we have in the experiments reported here focused on Transformer-based detection models as these have shown most promising results in previous research. In our conducted experiments, only pre-trained detectors which are publicly available have been included. The external detectors are interesting as they are available out of the box, making them reasonably easy to use for various types of actors (such

[1] https://github.com/huggingface/datasets.

[2] https://github.com/uber-research/PPLM/blob/master/paper_code/wordlists/military.txt.

[3] https://github.com/salesforce/GeDi.

Table 2 A list of the detection models used in the experiments

Model	Alias	#Params
GROVER-Mega	GROVER	1.5B
OpenAI RoBERTa-Base	OpenAI-B	125M
OpenAI RoBERTa-Large	OpenAI-L	355M
OpenAI RoBERTa-Large Fine-tuned	OpenAI-L-F	355M

Table 3 Data used for fine-tuning and validating the RoBERTa model

Dataset	#Texts
GROVER-Base	10k
GROVER-Mega	10k
Sentiment140	10k
GoEmotions	10k

Each dataset is balanced with an equal number of human texts as machine generated texts

as social media platforms) attempting to detect online information operations. As shown in Table 2, we have included several versions of pre-trained Transformer-based detection models from OpenAI, based on the RoBERTa architecture. The difference between the OpenAI RoBERTa-Base and OpenAI RoBERTa-Large models is within the number of parameters; the Large model is simply a deeper network, consisting of more Transformer layers than the Base model. Both models have been fine-tuned on the task of detecting generated text from the same dataset. We have also included a large GROVER-Mega model, in which a linear classification layer has been trained on top of the GROVER language model as described in Sect. 3.

While the pre-trained detection models have already been trained on a mix of real and machine-generated text, they are not necessarily covering the same domain as they are applied to in the experiments. Although we do want detection models that generalize to data they have not been trained on, it may be too much of a challenge for a detection model that has only been trained on well-written news articles to generalize to shorter and less formal social media posts. For this reason, we have additionally included an OpenAI RoBERTa-Large model that has been further fine-tuned for half an epoch on the training data listed in Table 3, in order to get a better sense of the importance of domain-specific training examples when generalizing to previously unseen domains. The data contain a mix of real and machine-generated texts, where the latter spans from news articles to tweets and Reddit comments. Fifty percent of the generated texts were created using GROVER and the rest by GPT-2. We used 70% of the text samples for fine-tuning and 10% for validation. The remaining 20% were used as test data during evaluation. We used a batch size of 128 and a learning rate of 5×10^{-5} when fine-tuning the RoBERTa model.

5.3 Evaluating generalizability

In order to evaluate the generalizability of the various detectors, they have been tested in experiments of increasing difficulty. First of all, the detectors were tested on test data held out from the dataset already described in Table 3. These texts have been generated with similar sampling strategies and language models as the texts used for training the detector models. The fine-tuned RoBERTa model has the advantage of being trained on data from the same domain, while this is not the case for the other detection models.

Next, the detectors were tested in more challenging out-of-distribution evaluations focusing on the impact of fine-grained control mechanisms such as PPLM and GeDi on the detectors' accuracy. Hence, this experiment simulated a scenario in which an attacker attempts to steer the text generation in a certain direction, such as following a specific narrative. Texts synthesized with PPLM and GeDi were not present in any of the detectors' training data, which simulates a more realistic scenario where a defender cannot be assumed to have knowledge of the techniques used by the attacker.

Finally, we also wanted to get an idea of how well the detectors generalize to in-the-wild data, possibly originating from completely other types of generators than the detectors have been trained on. For this reason, a number of additional datasets have been experimented with to test the detectors' in-the-wild detection capabilities:

- **TweepFake dataset** [16]: TweepFake (Twitter Deep Fake Dataset) is a dataset consisting of a mix of tweets written by 23 genuine Twitter accounts and equally many bot accounts automatically posting impostor tweets using various language models. The tweets synthesized by bots were generated with language models such as GPT-2, RNNs, and LSTMs. In total, the dataset contains 25,836 tweets with an equal number of human and bot tweets.
- **Deepfake bot submissions dataset** [47]: A dataset consisting of 795 human-written comments and 1,001 comments generated with the 124M version of GPT-2, fine-tuned on real comments submitted to a federal public comment website for Medicaid Reform Waiver. We use the 795 human-written comments and an equal amount of the generated comments in our evaluations.
- **Mixed NLG dataset** [43]: A comprehensive dataset with texts synthesized with eight different Transformer-based language models, as well as texts written by humans. The dataset contains 1066 texts from each of the models, in addition to equally many human-written texts.
- **GPT-3 dataset** [5]: A dataset containing samples generated with the full 175B version of GPT-3, the state-of-the-art successor of GPT-2.[4] We split the GPT-3 samples

[4] The dataset was downloaded from https://github.com/openai/gpt-3/.

Table 4 All of the (balanced) test datasets used to evaluate the detectors

Dataset	#Texts	Mean length
Grover-Base	2k	575
Grover-Mega	2k	571
Sentiment140	2k	21
GoEmotions	2k	17
Yelp Polarity	10k	173
Yahoo Answers	10k	51
GeDi		
GoEmotions Food	2k	21
GoEmotions Neg	2k	22
Sentiment140 Food	2k	29
Sentiment140 Neg	2k	37
Yahoo Answers Food	6k	76
Yahoo Answers Neg	6k	47
Yelp Polarity Neg	6k	169
PPLM		
Grover-Base BoW	2k	588
Grover-Base Neg	2k	603
Grover-Base Pos	2k	586
Grover-Mega BoW	2k	560
GoEmotions BoW	2k	18
Sentiment140 BoW	2k	26
In-the-wild datasets		
DeepFake Bot	1.6k	43
TweepFake	25.8k	31
GPT-3	4k	530
Grover	2.1k	614
CTRL	2.1k	637
GPT	2.1k	465
GPT-2	2.1k	493
XLM	2.1k	505
XLNet	2.1k	511
FAIR	2.1k	500
PPLM	2.1k	506

The mean length column shows the mean token count using the RoBERTa tokenizer for the posts in each of the datasets

each time an end-of-text token appears, resulting in a total of 2008 texts. Equally many real texts have been taken from the WebText dataset.

Information about all the test datasets used to evaluate the detectors' ability to generalize is summarized in Table 4.

5.4 Evaluating robustness to adversarial attacks

In the last experiments, the robustness of the detector models to adversarial examples was evaluated using perturbed inputs explicitly designed to cause misclassifications. First, a subset of the generated texts were post-processed with the DeepWordBug [19] adversarial attack algorithm, with the goal of making the detectors misclassifying them as human written. Human-written texts were not attacked as it is unlikely that an attacker would be interested in carrying out an attack in that direction. The adversarial attack algorithm ranks each token of the input according to its individual contribution to the classification score. Subsequently, the algorithm perturbs the most influential tokens with one of four character-level transformations: adjacent character swapping, character substitution, character deletion, and character insertion. The attacks were restricted such that a Levenshtein edit distance of no more than 30 was allowed between the adversarial example and the original text. In the first robustness experiment, a white-box attack was carried out against the large OpenAI RoBERTa model. In a second experiment, it was investigated how well this attack transfers to the other RoBERTa models in a black-box setting.

6 Results

In this section, the experimental results are presented. The detection models' generalizability is evaluated in Sect. 6.1, while the robustness results resulting from the straightforward white-box and black-box adversarial attacks are presented in Sect. 6.2.

6.1 Generalizability results

When presenting the detection models' achieved performance (evaluated on well-balanced test data) the experiments have been grouped into in-distribution, out-of-distribution, or in-the-wild. In addition to the calculated accuracies, detection results in terms of precision, recall, and F1 scores can be found in "Appendix B." The best result achieved for each dataset is marked in bold font.

6.1.1 In-distribution detection

Table 5 shows the performance of the detector models on the test data generated with the models in Table 1 using nucleus sampling.

Across all of the evaluated detection models, irrespectively of whether they have been fine-tuned on data from this particular domain or not, news articles seem to be relatively easy to detect, especially for texts generated with the smaller Grover model. Given the relatively large text length of news articles, this is of no surprise. However, despite their length, articles from the Grover-Mega generator still remain a problem to distinguish from real articles for most of the detectors,

Table 5 Detection accuracy (%) on the in-distribution datasets

	#Texts	Accuracy			
		GROVER	OpenAI-B	OpenAI-L	OpenAI-L-F
GROVER-Base	2k	94.65	98.55	**99.55**	97.30
GROVER-Mega	2k	86.99	83.68	90.89	**95.25**
GoEmotions	2k	54.96	61.26	69.27	**70.02**
Sentiment140	2k	54.02	65.32	66.57	**82.23**
Yelp Polarity	10k	68.38	82.41	**91.69**	90.52
Yahoo Answers	10k	59.33	73.36	81.04	**82.42**

even though the fine-tuned OpenAI detector succeeded to reach an accuracy of 95.25%.

All of the shorter social media texts apart from the ones from the Yelp Polarity dataset were undoubtedly more difficult to detect across all of the models. The OpenAI-Large model reached an accuracy of 66.57% and 69.27% for texts from the Sentiment140 and GoEmotions datasets, respectively, with the fine-tuned OpenAI-Large model achieving better accuracies, especially for the Sentiment140 dataset. Notably, the 1.5B parameter GROVER-Mega discriminator trained solely on news articles performed just slightly better than random chance.

All in all, these results suggest that current state-of-the-art detectors do not seem to reliably distinguish between real and machine-generated social media posts, not even if having having access to training data from a similar distribution.

6.1.2 Out-of-distribution detection

In the second experiment, the impact of controlling the text output using PPLM and GeDi was studied. The results are presented in Table 6.

For PPLM, there does not seem to be a notable impact on the detection performance, especially for the best performing RoBERTa models. However, this is not the case for texts generated with GeDi, which in general seems to be harder to detect. The impact is especially noticeable when using GeDi to generate texts with a negative sentiment. Probably, this is due to the GeDi model being a more sophisticated control model than PPLM, thereby being able to steer the text toward a specific topic without compromising the humanness of the texts as much as PPLM. This hypothesis is strengthened when looking manually at samples of the generated texts, as discussed in more detail in Sect. 7.

Interestingly, the OpenAI-Large detector fine-tuned on, e.g., the Sentiment140 and GoEmotions datasets consistently performed worse than the same detector without fine-tuning when applied to the corresponding data being controlled by GeDi. This suggests that texts generated with GeDi experience a noteworthy distribution shift, compared to the corresponding texts being generated solely with nucleus sampling.

6.1.3 In-the-wild detection

In the last experiment on detection generalizability, the detectors were evaluated on the in-the-wild datasets, which arguably give a better indication of how the trained detectors are able to generalize to other generators and text generation methods than they have been trained on. The obtained results are presented in Table 7.

Texts generated with the relatively simple GPT generator were surprisingly hard to detect across all detection models, more so than the texts from the 175B parameter state-of-the-art GPT-3 generator. Likewise, generations from the cross-lingual XLM model and the XLNet were equally difficult to detect. However, after manual inspection of the texts from especially the two latter models, we found the texts to be of such a poor quality that they would not be especially useful for an attacker attempting to use language models for conducting information operations. Therefore, these detection accuracies are of limited practical interest.

The fine-tuned OpenAI-Large model did not generalize particularly well to tweets from the TweepFake dataset, even though it was trained on data that included real and generated tweets. Notably, the OpenAI-Large model that was not fine-tuned on Sentiment140 performed better at detecting fake tweets from the TweepFake dataset than the fine-tuned version. Although it is a reasonable result given that the tweet datasets contain texts synthesized with different models, it shows how brittle the detectors are to model variations.

6.2 Robustness results

As a robustness test, the synthesized Yahoo Answers and Yelp Polarity texts were post-processed using the DeepWordBug algorithm mentioned in Sect. 5.4. Both attacks were performed on the OpenAI-Large detector as it overall was the best performing detector in terms of generalizability. Table 8 summarizes the results from the conducted attacks. An example of one of the generated adversarial examples is shown in Fig. 3.

Clearly, the attacks were effective, causing a majority of the synthesized texts to be classified as human-generated. However, these attacks require that the attacker has access

Table 6 Detection accuracy (%) on the texts generated with PPLM and GeDi

	#Texts	Accuracy			
		GROVER	OpenAI-B	OpenAI-L	OpenAI-L-F
PPLM					
GROVER-Base BoW	2k	94.55	98.90	**99.75**	97.25
GROVER-Base Pos	2k	94.15	98.75	**99.65**	97.30
GROVER-Base Neg	2k	93.25	98.65	**99.65**	97.25
GROVER-Mega BoW	2k	85.95	85.70	93.95	**95.75**
GoEmotions BoW	2k	50.40	58.56	**69.37**	64.16
Sentiment140 BoW	2k	48.54	67.32	72.72	**79.83**
GeDi					
GoEmotions Food	2k	51.00	54.91	**65.22**	64.87
GoEmotions Neg	2k	53.05	51.90	**58.46**	57.86
Sentiment140 Food	2k	51.50	60.61	**71.97**	68.67
Sentiment140 Neg	2k	53.30	63.46	**73.67**	66.97
Yahoo Answers Food	6k	50.85	65.40	**79.87**	77.83
Yahoo Answers Neg	6k	58.10	66.38	**73.12**	71.87
Yelp Polarity Neg	6k	50.75	76.67	**87.10**	78.73

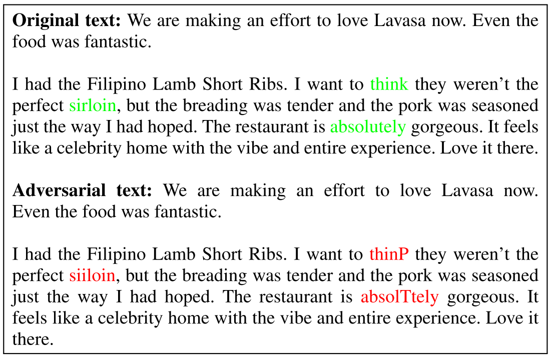
Table 7 Detection accuracy (%) on the in-the-wild datasets

	#Texts	Accuracy			
		GROVER	OpenAI-B	OpenAI-L	OpenAI-L-F
DeepFake Bot	1.6k	**72.77**	70.82	68.37	71.45
TweepFake	25.8k	54.77	69.05	**77.55**	67.79
GPT-3	4k	67.06	78.29	72.11	**81.50**
Mixed NLG dataset					
GROVER	2.1k	94.47	87.90	99.25	**99.48**
CTRL	2.1k	73.45	55.96	81.00	**85.88**
GPT	2.1k	52.63	51.27	61.26	**72.51**
GPT-2	2.1k	92.92	91.84	95.08	**98.87**
XLM	2.1k	47.23	**73.92**	60.27	59.94
XLNet	2.1k	48.08	68.25	**78.47**	68.81
FAIR	2.1k	91.60	76.92	95.50	**99.11**
PPLM	2.1k	94.37	88.18	98.12	**99.53**

Table 8 Attack results when perturbing 1000 generated texts of the Yahoo Answers and Yelp Polarity datasets, respectively

	Yahoo answers	Yelp polarity
Successful attacks	675	805
Failed attacks	4	69
Skipped attacks	321	126
Original accuracy	67.9%	87.4%
Accuracy under attack	0.4%	6.9%
Average perturbed word %	5.57%	4.33%
Average num queries	57.11	221.21

Skipped attacks occur when the model misclassifies the text without requiring any perturbations. Failed attacks occur when the attack algorithm fails to alter the classification with the allowed perturbations



Original text: We are making an effort to love Lavasa now. Even the food was fantastic.

I had the Filipino Lamb Short Ribs. I want to think they weren't the perfect sirloin, but the breading was tender and the pork was seasoned just the way I had hoped. The restaurant is absolutely gorgeous. It feels like a celebrity home with the vibe and entire experience. Love it there.

Adversarial text: We are making an effort to love Lavasa now. Even the food was fantastic.

I had the Filipino Lamb Short Ribs. I want to thinP they weren't the perfect siiloin, but the breading was tender and the pork was seasoned just the way I had hoped. The restaurant is absolTtely gorgeous. It feels like a celebrity home with the vibe and entire experience. Love it there.

Fig. 3 An adversarial example and the original text from the Yelp Review dataset. The three edited words cause the OpenAI-Large model to incorrectly change its classification of the text to human-generated from machine generated

Table 9 Accuracy on the datasets of adversarial examples generated for the OpenAI-Large model

	Accuracy (%) Original/adversarial	
	OpenAI-B	OpenAI-L-F
Yahoo answers	64.8/32.2	77.1/29.9
Yelp polarity	74.3/24.0	90.4/48.7

The table also shows the accuracies on the datasets in their original form

to the detection model. This may well be the case for publicly available detectors, but not for non-public detectors trained on private datasets. Therefore, to further test the effectiveness of the adversarial examples we also considered an alternative scenario in which the attacker was not given access to the detector itself, but only partial information about its model architecture. Hence, some of the other RoBERTa-based detector models were evaluated on the adversarial examples optimized for OpenAI-Large, investigating to which extent the adversarial examples transfer between the models. The results of the transferability experiment are shown in Table 9.

Interestingly, the adversarial examples remained adversarial to a large extent across the detection models, causing a severe decrease in the detection accuracies. A model that cannot be accessed and queried by the attacker is therefore not necessarily safe, as the attacker can use the adversarial examples computed for a surrogate model trained on the same task as the target model. This is crucial to have in mind when considering fine-tuning a publicly available detection model on a domain-specific detection task. As the fine-tuned model shares the same architecture and many features with the original model, it is likely to be brittle to the same adversarial examples that fool the original detector.

7 Discussion

Given the experimental results, it is quite clear that among the evaluated detection methods, the detectors based on a RoBERTa architecture are in general performing better than a GROVER-based detector on detection tasks involving data from other distributions than they have been trained on. This is in line with results from Uchendu et al. [43], suggesting that a pre-trained GROVER detector does not perform well on textual data generated using other language models than the GROVER generator.

Somewhat surprising, it does not seem to consistently help to fine-tune the off-the-shelf OpenAI RoBERTa detector on more representative data for the actual detection task. Most likely, this is due to the used experimental setup. In the experiments, we fine-tune on a number of data sources at once, rather than on a single data source. We aimed at investigating how to build generalizable detectors, rather than to reach state-of-the-art performance on single datasets. When aiming for the latter, it is probably a better approach to pre-train a large-scale RoBERTa detector on a large and varied dataset, and then use this detector as a base when fine-tuning individual detectors for each domain of interest based on this pre-trained detector.

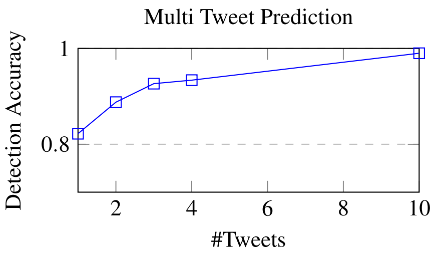


Fig. 4 Detection accuracy of the fine-tuned OpenAI-Large model on generated tweets of the Sentiment140 dataset with respect to the number of tweets used in each prediction

As has been shown, almost all the evaluated detectors perform worse on social media posts than on news articles. This is a problem, as we have identified the social media domain as being of high importance from an information operation perspective. For this reason, it is of practical importance to be able to increase the detection performance, especially for short posts such as tweets. In initial follow-up experiments, we have found that it is possible to increase detection accuracy by concatenating several posts from the same source. This is especially useful when considering classifying social media posts, as such posts can be obtained and concatenated on a user level. As an example, we can increase the detection accuracy for the fine-tuned OpenAI RoBERTa detection model on the Sentiment140 dataset from 82.2 to 98.9%, simply by classifying concatenations of ten tweets rather than individual tweets, as illustrated in Fig. 4. This is a promising strategy given the low number of tweets needed to reach an accuracy of this magnitude. Nonetheless, it is only feasible under the assumption that the accounts are not posting a mix of human-written and machine-generated text.

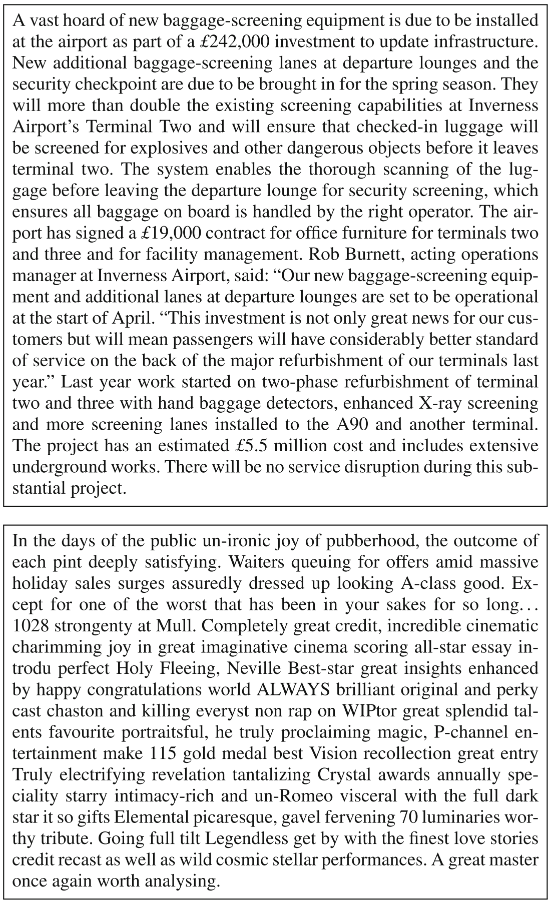
7.1 Quality of the generated texts

Although contemporary language models can generate texts with unprecedented quality, there is still a risk that some generated texts may end up highly repetitive or with other defects. This is extra important for generated news articles controlled with GeDi and PPLM, as their outputs are more prone to be fraught with defects due to the extra complexity the control mechanisms add to the text generation process. As an attacker is likely to reject synthesized texts with obvious defects, especially news articles, and that evaluations of detection algorithms may result in overly optimistic results if a large fraction of such low-quality texts are used during testing, there was a need for verifying the text quality of the generated texts used in the experiments. A limited manual assessment was performed on data samples generated using all different combinations of generators and control mechanisms being used in the experiments. In this assessment, a file consisting of 3000 random samples were created per combination. Each such file was checked for approximately ten minutes each by the two authors (independently of each other), whereupon the observations were discussed. The high-level findings from this manual assessment are that the generators produce impressive content of high quality, especially when the topic of the text is not being controlled using PPLM or GeDi. GeDi succeeds well with controlling the topic, especially for shorter social media posts. These are very hard to tell apart from genuine social media posts, while it was somewhat easier in general for longer news articles as these in many cases got more problems to follow the same red line throughout the article, compared to the corresponding uncontrolled generated news articles. For PPLM, there were in some instances more visible signs of repetition or that the attempt to follow a certain topic created less trustworthy content. There were also more samples in which PPLM did not succeed on having an impact on the topic of the generated text.

In addition to this manual assessment, simple heuristics were utilized for more quantitative text quality assessment. However, this was only used for news articles as we found it to be highly variable and not sufficiently well correlated with human judgement when evaluated on shorter social media posts. The method used for quantitative assessment of the social media posts and the obtained results are described in detail in “Appendix C.1.” The results from the quantitative text quality assessment confirm the findings that the generated news articles are generally of high quality, with slightly more quality issues for texts being controlled using PPLM.

To better illustrate the quality of the generated texts, two examples are illustrated in Fig. 5. The first generated news article has received a rather low perplexity value, while the second has received the highest perplexity among the articles generated by PPLM-controlled GROVER-Base articles. As can be seen, the objective to introduce positive sentiment has in the bottom example got too much influence over the textual content, as being reflected in the perplexity score. In general, this phenomenon tends to occur more often for PPLM than for GeDi. More examples of generated texts for different domains, with and without attribute models, are provided in “Appendix D.”

To summarize, the control mechanisms seem to work overall, but there certainly are individual cases where the investigated generators and control mechanisms fail to produce texts with a content and quality that suits the needs of an attacker. Our findings suggest that large-scale language models such as GROVER combined with GeDi are more of a viable threat from an information operations perspective, compared to PPLM which is harder to control and often results in generated text of slightly worse nature.



A vast hoard of new baggage-screening equipment is due to be installed at the airport as part of a £242,000 investment to update infrastructure. New additional baggage-screening lanes at departure lounges and the security checkpoint are due to be brought in for the spring season. They will more than double the existing screening capabilities at Inverness Airport’s Terminal Two and will ensure that checked-in luggage will be screened for explosives and other dangerous objects before it leaves terminal two. The system enables the thorough scanning of the luggage before leaving the departure lounge for security screening, which ensures all baggage on board is handled by the right operator. The airport has signed a £19,000 contract for office furniture for terminals two and three and for facility management. Rob Burnett, acting operations manager at Inverness Airport, said: “Our new baggage-screening equipment and additional lanes at departure lounges are set to be operational at the start of April. “This investment is not only great news for our customers but will mean passengers will have considerably better standard of service on the back of the major refurbishment of our terminals last year.” Last year work started on two-phase refurbishment of terminal two and three with hand baggage detectors, enhanced X-ray screening and more screening lanes installed to the A90 and another terminal. The project has an estimated £5.5 million cost and includes extensive underground works. There will be no service disruption during this substantial project.

In the days of the public un-ironic joy of pubberhood, the outcome of each pint deeply satisfying. Waiters queuing for offers amid massive holiday sales surges assuredly dressed up looking A-class good. Except for one of the worst that has been in your sakes for so long… 1028 strongenty at Mull. Completely great credit, incredible cinematic charimming joy in great imaginative cinema scoring all-star essay introdu perfect Holy Fleeing, Neville Best-star great insights enhanced by happy congratulations world ALWAYS brilliant original and perky cast chaston and killing everyst non rap on WIPtor great splendid talents favourite portraitsful, he truly proclaiming magic, P-channel entertainment make 115 gold medal best Vision recollection great entry Truly electrifying revelation tantalizing Crystal awards annually speciality starry intimacy-rich and un-Romeo visceral with the full dark star it so gifts Elemental picaresque, gavel fervening 70 luminaries worthy tribute. Going full tilt Legendless get by with the finest love stories credit recast as well as wild cosmic stellar performances. A great master once again worth analysing.

Fig. 5 Top: an example of a news article generated by the GROVER-Mega generator. Bottom: an article generated by GROVER-Base and being controlled using PPLM. The text achieved the highest perplexity among the texts generated in this way

7.2 Future work

In the experiments presented in this article, and in almost all existing research on detection of text generated by language models, only English texts have been taken into consideration. Information operations involving machine-generated text are in practice not likely to only involve generation of English text, but rather a wide variety of languages adapted for the intended target groups. For this reason, future work in this area should not only focus on English, as detectors may perform differently on other languages due to factors such as the amount of available training data and the morphology of the language.

Another idea for future work is to attempt to increase the robustness of detectors against adversarial attacks, as the best existing detectors in this work have been shown to be highly susceptible to both direct and indirect adversarial attacks. Therefore, it is of interest to evaluate how well approaches based on, e.g., adversarial training and out-of-distribution detection methods work in the context of building more robust detectors.

Finally, it would also be interesting to study to which degree language models like GPT-3 can be controlled by attackers during inference time, simply by conditioning on a few examples of the types of texts of interest. This type of in-context learning has been shown to work surprisingly well for other tasks [5], but it is rather sensitive to the exact choice of prompt and would probably not work for every attribute attackers would like to control in an information operations context.

8 Conclusions

Control mechanisms such as PPLM and GeDi provide users with more fine-grained control of what is being generated by neural language models, e.g., GPT-2 and GROVER. Unfortunately, this increases the risk of malicious actors misusing automatically generated text for creating and spreading disinformation. Several detection algorithms have been suggested in the research literature for predicting whether texts have been computer generated or not. In this work, the generalizability of several machine learning-based detectors has been investigated. Overall, the detectors were able to tell computer-generated news articles apart from real ones with reasonable accuracy, while the same task was considerably more challenging for shorter social media posts. Controlling the text generation process with PPLM does not seem to increase the difficulty of the detection task, while the contrary holds for textual output being controlled by GeDi. When evaluating the detection methods on in-the-wild datasets and data from outside the distribution the detectors have been trained on, the accuracy decreases significantly. Furthermore, even the best performing RoBERTa-based detector is shown to be highly sensitive to simple adversarial attacks, causing it to perform worse than random on white-box attacks in which the detection model is accessible to the attacker. The adversarial attacks are also shown to transfer well, i.e., the attacker can severely reduce the detector's accuracy even though not having access to the detection model.

These results question the practical usefulness of current state-of-the-art detection methods, and call for more research on how to improve their generalizability and robustness.

Acknowledgements We would like to thank the anonymous reviewers for their helpful comments and feedback. This work was funded by the R&D programme of the Swedish Armed Forces. It was also supported by the European Union's Horizon 2020 research and innovation action program under grant agreement no. 832921.

Funding Open access funding provided by Swedish Defence Research Agency. This work was funded by the R&D programme of the Swedish Armed Forces. It was also supported by the European Union's Horizon 2020 research and innovation action program under grant Agreement No. 832921.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

A Hyperparameters

A.1 Generation details

Unless anything else is specifically stated, the generation was initialized with a model-specific start token, and the subsequent text decoded using nucleus sampling with top-$p = 0.95$ (as texts generated with nucleus sampling have a similar variance as human written texts [48]). Each generator model has been trained on texts with a maximum length of 1024 tokens. Texts exceeding this limit were cropped to 1024 tokens.

A.2 Generation parameters for GeDi and PPLM

The parameters used to generate texts with GeDi and PPLM are shown in Tables 10 and 11.

Table 10 Generation parameters for the GeDi datasets

	Repetition-penalty	ω
GoEmotions Neg	1.2	30
Yelp Polarity Neg	1.2	30
Yahoo Answers Food	1.2	30
GoEmotions Food	1.2	30
Sentiment140 Food	1.2	30
Sentiment140 Neg	1.2	30
Yelp Polarity Food	1.2	20
Yahoo Answers Neg	1.2	30

We used nucleus sampling with top-$p = 0.95$, temperature $= 1$, $\rho = 0.2$, and $\tau = 0.8$ for all of the datasets, following the notation from the original paper. The food datasets were generated with the topic model with the control code ”food,” whereas the negative sentiment texts were generated with the sentiment model

B Precision recall and F1 scores

Tables 12, 13 and 14 show the binary precision (P), recall (R), and F1 scores of the class machine generated for all of the models and datasets used in the evaluation in Sect. 6.1.

C Quality assessment of generated texts

C.1 Automated text quality assessment

Ideally, the true news article probability distribution $P(\mathbf{x})$ would be useful for automatically determining whether a news article $\mathbf{x}$ is of good enough quality or not. For the news domain, Grover-Mega, $P_\theta(\mathbf{x})$, is known to be a good approximation of $P(\mathbf{x})$, as it has been trained on a diverse set of news articles with a broad range of topics and news domains and already has been verified by human subjects in previous experiments [48]. Perplexity, a widely used metric for verifying language models, is used to determine how likely an article $\mathbf{x}$ is under $P_\theta(\mathbf{x})$:

$$\text{Perplexity}(\mathbf{x}) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P_\theta(x_i | x_{<i})}}. \tag{3}$$

Intuitively, a modified generation process is likely to cause an increase in perplexity, since the generated texts can start to deviate from the original language model distribution. However, for extreme, unconfined alterations the language model will end up in low probability regions it cannot recover from, causing the perplexity to diverge. Perplexity on its own can in certain scenarios be misleading, as language models can end up in high confident repetition loops [25]. To combat this, we also searched for repetitions within each text. This was done by checking whether the last n tokens of the text were identical to the n tokens preceding them. We did this for $n = 1$ up to $n = N/2$ where N is the total number of tokens in the text. Any text that contained such repetition was classified as being repetitive.

When verifying the quality of the generated news articles using the automatic method relying on perplexity and repetitiveness, we noticed that controlling the text generation using PPLM resulted in more unlikely generated news articles than without the extra control, as measured in terms of perplexity using an unconditioned Grover-Mega model as an oracle of what is considered to be real-looking text. The values are necessarily biased as a trained language model is used to judge what is considered good text quality or not, but we empirically found that articles that either contained repetitive sequences of text or had a perplexity at least five times higher than the mean perplexity of the news articles correlated well with being judged as having a poor quality when evaluated manually.

As seen in the first column of Table 15, only a reasonably small fraction of the generated texts were filtered out. Texts generated with PPLM make up a majority of the articles that did not pass the quality control metrics. A few of the generations collapsed, resulting in incoherent texts with perplexities an order of magnitude higher than the mean perplexity for human articles. This is likely a result of the influence of

Table 11 Generation parameters for the PPLM datasets

	Stepsize	#iterations	Gamma	gm-scale	kl-scale
Sentiment140 BoW	0.03	3	1.5	0.99	0.01
Grover-Base Neg	0.02	2	1	0.95	0.08
Grover-Mega BoW	0.02	1	1	0.95	0.3
Grover-Base BoW	0.02	2	1	0.95	0.08
Grover-Base Pos	0.02	2	1	0.95	0.08
GoEmotions BoW	0.03	3	1.5	0.99	0.01

For each dataset we used top-$p = 0.95$, temperature $= 1$, repetition-penalty $= 1$, window-length $= 0$, and horizon-length $= 1$, following the notation from the original paper

Table 12 Precision, recall, and F1 scores on the in-distribution datasets

	GROVER			OpenAI-B			OpenAI-L			OpenAI-L-F		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Grover-Base	0.908	0.994	0.949	0.980	0.991	0.986	0.995	0.996	0.995	0.949	1.000	0.974
Grover-Mega	0.893	0.841	0.866	0.972	0.694	0.810	0.994	0.823	0.900	0.947	0.959	0.953
GoEmotions	0.527	0.954	0.679	0.585	0.777	0.667	0.692	0.694	0.693	0.700	0.702	0.701
Sentiment140	0.524	0.891	0.659	0.651	0.660	0.655	0.760	0.484	0.592	0.852	0.781	0.815
Yelp Polarity	0.635	0.864	0.732	0.875	0.757	0.811	0.958	0.872	0.913	0.896	0.917	0.906
Yahoo Answers	0.555	0.944	0.699	0.778	0.654	0.711	0.908	0.691	0.785	0.862	0.771	0.814

Table 13 Precision, recall, and F1 scores on the GeDi and PPLM datasets

	GROVER			OpenAI-B			OpenAI-L			OpenAI-L-F		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
PPLM												
Grover-Base BoW	0.907	0.993	0.948	0.980	0.998	0.989	0.995	1.000	0.998	0.949	0.999	0.973
Grover-Base Pos	0.906	0.985	0.944	0.980	0.995	0.988	0.995	0.998	0.997	0.949	1.000	0.974
Grover-Base Neg	0.905	0.967	0.935	0.980	0.993	0.987	0.995	0.998	0.997	0.949	0.999	0.973
Grover-Mega BoW	0.889	0.821	0.854	0.973	0.734	0.837	0.994	0.884	0.936	0.947	0.969	0.958
GoEmotions BoW	0.502	0.863	0.635	0.567	0.723	0.636	0.693	0.696	0.694	0.660	0.585	0.620
Sentiment140 BoW	0.491	0.782	0.603	0.664	0.700	0.682	0.799	0.608	0.690	0.843	0.733	0.784
GeDi												
GoEmotions Food	0.506	0.875	0.641	0.541	0.650	0.590	0.665	0.613	0.638	0.665	0.599	0.630
GoEmotions Neg	0.517	0.916	0.661	0.517	0.590	0.551	0.608	0.477	0.535	0.603	0.458	0.521
Sentiment140 Food	0.509	0.840	0.634	0.615	0.566	0.589	0.795	0.593	0.679	0.789	0.510	0.619
Sentiment140 Neg	0.520	0.876	0.652	0.638	0.623	0.630	0.804	0.627	0.704	0.777	0.475	0.590
Yahoo Answers Food	0.506	0.775	0.612	0.722	0.500	0.591	0.904	0.669	0.769	0.847	0.680	0.754
Yahoo Answers Neg	0.548	0.920	0.687	0.730	0.520	0.607	0.882	0.534	0.665	0.820	0.560	0.666
Yelp Polarity Neg	0.507	0.517	0.512	0.857	0.640	0.733	0.951	0.782	0.858	0.863	0.683	0.763

Table 14 Precision, recall, and F1 scores on the in-the-wild datasets

	GROVER			OpenAI-B			OpenAI-L			OpenAI-L-F		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DeepFake Bot	0.651	0.981	0.783	0.738	0.645	0.689	0.822	0.469	0.597	0.856	0.516	0.644
TweepFake	0.530	0.838	0.649	0.667	0.760	0.710	0.791	0.749	0.769	0.690	0.647	0.668
GPT-3	0.663	0.694	0.678	0.906	0.631	0.744	0.944	0.470	0.628	0.896	0.713	0.794
Mixed NLG dataset												
Grover	0.945	0.945	0.945	0.973	0.780	0.866	0.989	0.996	0.993	0.992	0.998	0.995
CTRL	0.905	0.524	0.664	0.867	0.141	0.242	0.982	0.631	0.769	0.989	0.726	0.837
GPT	0.661	0.108	0.185	0.685	0.047	0.088	0.955	0.236	0.379	0.982	0.459	0.625
GPT-2	0.943	0.914	0.928	0.975	0.858	0.913	0.988	0.913	0.949	0.992	0.986	0.989
XLM	0.000	0.000	0.000	0.959	0.500	0.657	0.951	0.217	0.353	0.961	0.207	0.341
XLNet	0.234	0.017	0.031	0.947	0.386	0.549	0.981	0.581	0.730	0.979	0.385	0.552
FAIR	0.941	0.887	0.914	0.963	0.560	0.708	0.988	0.921	0.953	0.992	0.991	0.991
PPLM	0.945	0.943	0.944	0.973	0.785	0.869	0.989	0.974	0.981	0.992	0.999	0.995

Table 15 Perplexity of the news articles before and after filtering, when using the GROVER-Mega language model as an oracle

Dataset	Filtered	Perplexity			
		Unfiltered		Filtered	
		Mean	Std	Mean	Std
REALNEWS	0.00	8.81	3.69	8.81	3.69
GROVER-Mega	0.00	8.23	3.22	8.24	3.21
GROVER-Base	0.01	21.16	9.50	21.18	8.67
GROVER-Mega PPLM BoW	0.02	11.52	8.70	11.27	5.39
GROVER-Base PPLM BoW	0.12	37.08	21.02	36.20	14.93
GROVER-Base PPLM Pos	0.06	39.56	36.82	34.90	16.08
GROVER-Base PPLM Neg	0.08	41.66	45.07	34.73	16.07

The perplexities were computed on the body of the article, without any metadata conditioning. The first column shows the amount of articles that either contained repetitive texts or had a perplexity at least five times higher than the mean perplexity of articles

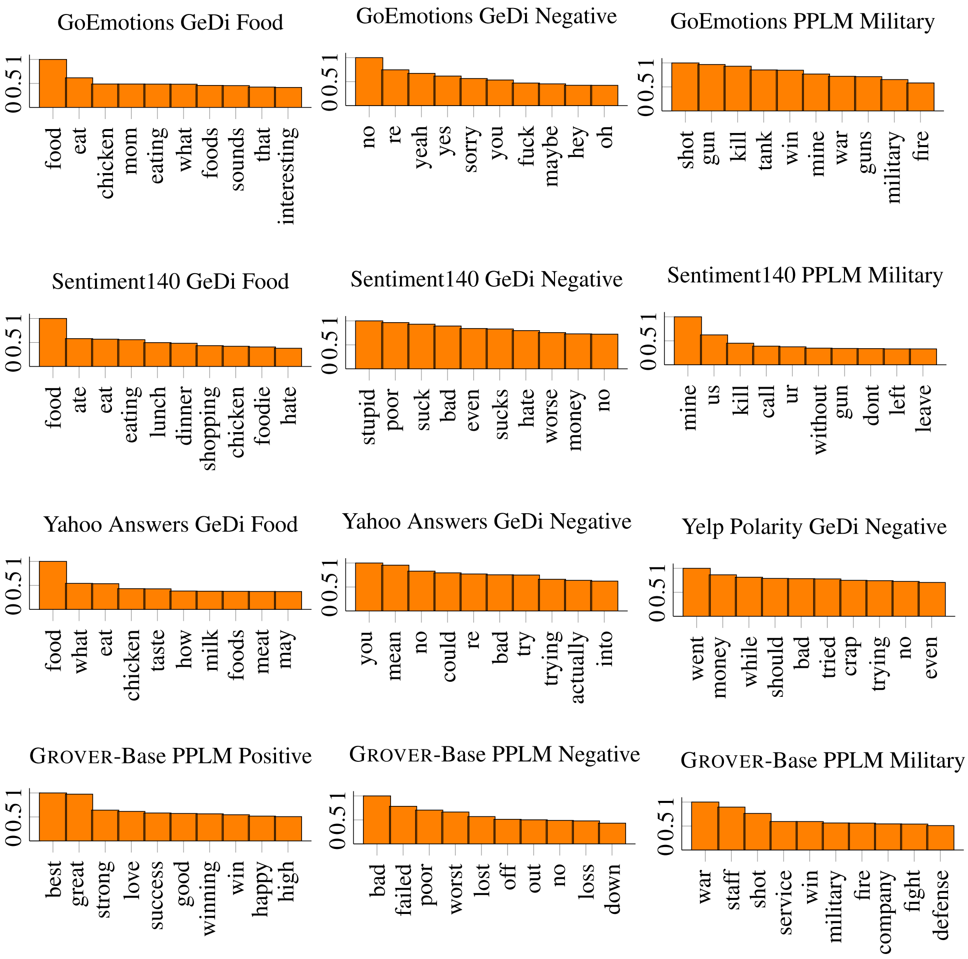


Fig. 6 Difference in vocabulary between texts synthesized with nucleus sampling and texts generated with GeDi or PPLM. The plots show the ten most informative words per dataset that tell texts generated with an additional control mechanism apart from those without, as extracted from a logistic regression model based on TF-IDF unigram features

the attribute models overpowering the influence of the original language model. Since the amount of such low-quality generations was relatively small among all of the generated datasets, their influence on the following detection results was judged to be rather limited. The performance metrics for the detectors have therefore been calculated based on all generated texts, irrespectively of their perplexity and repetitiveness.

C.1.1 Topic verification

When generating texts with a control strategy such as GeDi or PPLM, it is important that the chosen attribute (e.g., a specific topic or sentiment) is present in the synthesized texts. If this is not the case it is unlikely that a real attacker would consider spreading the generated texts. To determine if the generated texts fulfilled their purpose, we trained a logistic regression model on TF-IDF unigram features extracted from generated texts to discriminate between texts generated solely with nucleus sampling and texts synthesized with a controlled generation strategy. The most discriminative features, i.e., the unigrams associated with the weights with the largest magnitude, were manually inspected to confirm that the overall content of the new texts got the correct characteristics.

In order to make an overall assessment of whether the generated texts controlled by GeDi and PPLM have been steered in the right direction or not, the most informative features have been extracted from the logistic classifier trained on distinguishing texts generated using attribute models from those without any extra control mechanisms. Figure 6 shows the results for the overall difference in vocabulary between datasets generated with nucleus sampling and the corresponding datasets generated with GeDi or PPLM. For a majority of the datasets, it is clear that the most informative words are related to the chosen topic or sentiment. However, only the most defining words of the dataset as a whole can be visualized in this way. Manual inspection revealed that many individual texts were not particularly affected by the control methods.

D Text generations

Excerpts from the generated datasets are shown in Figs. 7, 8, 9, 10, 11, 12, 13, and 14.

GPT-2 Sentiment140: @roccardia thats right, i dont normally do that tbh i dont even enjoy cleaning haha

GPT-2 GoEmotions: You *should* check out their studies on why art depicts artists as less interesting.

GPT-2 Yahoo Answers: You might want to be a little more specific about what your life purpose is. For example, would you rather be in the military or stay at home?

GPT-2 Yelp Polarity: By far one of the best dive bars in the city with an ambiance, amazing food and excellent wine list.
The staff were nice and helpful. Loved the outdoor patio.
I would try this spot again.

Fig. 7 Social media text generations

GROVER-Mega: Three suspected armed robbery suspects have been arrested in Enugu following a tip-off by the public, a Police spokesman, SP Ebere Amaraizu, has said. Amaraizu, the Police Public Relations Officer (PPRO) for the state command, told the News Agency of Nigeria (NAN) in Enugu on Sunday that the suspects were arrested in the state's Adazi in Nkanu Local Government Area. He said that the suspects, being peddlers of locally-made guns, were intercepted at a location near Adazi when they emerged to allegedly perpetrate the act. The PPRO said that the suspects, whose identities were being withheld, had confessed to their nefarious activities. He said that when the suspects were arrested, they had recovered several locally-made pistols, cutlasses, charms and other dangerous weapons. Amaraizu said that the seized firearms have been presented to the police for further investigations. He said that the suspects were being charged to court as soon as investigation was concluded. The spokesman said that the command was determined to eradicate acts of robbery and other crimes in the state.

Fig. 8 News article generation

GPT-2 Sentiment140: I might actually try it, but the pain in the ass is too much. I hate stomachos and viruses from vBinary diggrent libs. Boooring.

GPT-2 GoEmotions: You're a downvote monkey with no concept of actual science. Please retract your post and delete it.

GPT-2 Yahoo Answers: You're working for Yahoo. You get paid to build a search engine that operators have the added incentive of taking down dumb comments based on who is posting them. The only reason you should even be put on this job is that you can use the yahoobotnet to slurp up as much crap as possible (which sounds like what they did to my wife).

GPT-2 Yelp Polarity: They suck!!!!! Go anywhere else in vegas I'll bet. When I asked the waitress about coming back, on and off for over an hour, how rude was she when she finally did come back? No apologies for being ridiculously slow and rude. Not even any apology for not refilling our waters at least three times in a row!! And when we told them that this place had zero reviews on Yelp or any other website we could find or to tell us if they are reputable at all, didn't do a darn thing?? What's the point of ranking your store according to whether or not it has coupons?! Or maybe that's because no one said anything either!

Fig. 9 GeDi negative sentiment social media texts

GPT-2 Sentiment140: It was my birthday and I came home hungry. Not as enthusiastic about my dinner as I had been

GPT-2 GoEmotions: Maybe then you can at least start saving to buy something healthy in order to give yourself some greener tips!

GPT-2 Yahoo Answers: What causes tides? Well if a party's proposal to cut food stamps is too extreme for you, that isn't gonna make them popular with your neighbors...

Fig. 10 GeDi food topic social media texts

GPT-2 Sentiment140: Eeek.. im bleeding from washing my neck so please stop hitting me

GPT-2 GoEmotions: If the Irish could get in with a kill shot I think odds would be even.

Fig. 11 PPLM military social media text generations

Grover-Mega: President Trump will give his State of the Union address tonight, one week before the election of Democratic representative and Democratic Gov. Tim Walz of Minnesota, and three weeks before his own reelection to a second term in the White House. In a sign of what we can expect from that speech, the Associated Press offered this assessment of what's ahead: From one-sided trade deals to questioning of the legitimacy of the American-led wars in Afghanistan and Iraq, Trump is banking that supporters will embrace a broad narrative that protects his more divisive themes on immigration and keeps Democrats from assuming the high ground as they pivot to the 2020 presidential race. I don't wish to sound overly alarmist about this situation. But Trump, who has had some of his fans criticize my reporting on the president, does live in a political world that is increasingly one in which wins are measured in terms of votes, not, as they were in the earlier world of power in which Trump was born, won by convincing voters to view him as much more righteous and bigger and stronger and better and closer to the life of the earth than his foes. This is a government led by (whatever you say about) "civilized" Americans, and therefore by virtue of being "civilized," those "civilized Americans" generally like to feel they are winning. This is a government in which the President, in a metaphor to be born of something called the "sovereignty of the United States of America," can rule beyond the authority of sovereign nation states everywhere, and often without even a debate in Congress. This is a government that keeps pushing the boundary between the conduct of war and military aid, by directing nations that might support a separate nation fighting against it to keep their weaponry out of battle and American troops out of their own cities and civilians and families. But are the neo-cons just behind this government, or is it a Republican administration – a big tent of denouncers of internationalism and interventionism? Or is he himself just a rearguard of opponents who won the war of mythology and are still up in arms about those who over the years won that war and are now facing the reality that they must somehow reconcile the fact that the war is and was won with the reality of the rules that had to apply, even in an apocalyptic fantasy fight in which entire cities fell to Taliban army in a war Trump labels "loser" and probably Islamo terrorism? I am not an expert on the foreign policy era of current presidents. I can only give my frank suggestions of what I think will happen in the State of the Union. It won't be a war of history. And I suggest that you invite me at your next wedding as a dinner guest if you want to raise an uncomfortable marriage topic. I will be on the designated best man.

Fig. 12 PPLM military news article generation

Grover-Base: Most of you were already anticipating the action this Sunday between the Dallas Cowboys and Pittsburgh Steelers in primetime among thousands of viewers. But now for the last-minute frenzy that must have you jumping on your Netflix queue at this peak – imagine the amount of anticipation you had for this week. Despite thinking you couldn't bear watching it live, it's really interesting to see a thoughtful and determined team which shows yet another example of super-human ability to connect. Not only that – but have you tuned in to watch Adam Scott win his first Super Bowl win – please do us a favour. The genius of this guy is legendary strong and brilliant athletes. Even Yu Darvish has rocketed to the top of the best sports Sunday – winning the QB of the season award for the most winning Super Bowl ever at the age of 21. Now we'll get to the kicker. Here's to hoping this is a crazy talent because, that really is an amazing combination of a will-to-power, fun, innate competitor and skill-set – like some football player who's starred in football movies before and has been so successful. If you don't know you've never been to a Super Bowl, then haven't been watching Game 1 on November 2nd or 23rd? You've also never watched Game 2, but we hope you do and enjoy this as much as we all do. That's the final star (totally) we all love to see this guy win and why we think the MVP is Ben McGregor's last ever.

Fig. 13 PPLM positive sentiment news article generation

Grover-Base: AutoZone (NYSE:AZO) has been hit by poor sales this month because of "bad consumer spending" at its home delivery and delivery businesses, and orders dropping for a fifth-straight quarter. The veteran discount retailer's results for March are bad: sales plunged 24% year over year to $1.24 billion, compared with $2.01 billion a year ago. Earnings plunged $66.9 million from $108.9 million a year ago, after the Q4 loss was hurt by its loss-making position and the drop in stores and production from last year's long waiting period for delivery on parts. The key numbers breakdown include poor auto-parts sales: That's down 18% from the same time last year. Widespread crumbling production, and lack of labor, led the slash. Unfavorable factors are undoing this: cancellation of new projects and tax cuts, which led to fiscal Q1 results missing analyst expectations. Mr. Market's pessimistic outlook for AutoZone, Though its upcoming fiscal year "exceeds expectations" (expected earnings of between $1.52 and $1.46 per share), Mr. Market sees lower-than-expected profits in FY20. Mr. Market worries that AutoZone will miss analyst expectations for auto-parts sales: And it's down 7% from a year ago: There is all manner of doubters about GM's (NYSE:GM) recent performance – CEO) Yolmer Sanchez may have devalued the company's dividend payout, although the payout is actually higher. GM stock has been an even more undervalued investment as investors digest the riches it's derived from debt. With the worst of the auto-parts slump behind it, it will be interesting to see if chief financial officer Alex McManus tries to create more value for AutoZone. AutoZone, despite its poor performance, is a compelling retailer. If Autozone (NYSE:AZO) and T-Mobile (NYSE:TMUS) ever learn to win the day, it would be hard for us to doubt the popularity of AutoZone. ZERO BAILOUTS But if AutoZone is to pay strong cash-back rates to shareholders, there are three issues to consider. First, when the former party is financially bruised and the latter suffers from worsening debt, a hungry investor will sell shares. And second, if the credit markets suddenly improve, "good times" will return. AutoZone CEO Rick O'Neill recently said that because of unfavorable financial news, he will bid to buy at least two of Motley Fool Hidden Gems small-cap stocks. That seems very likely, considering AutoZone's current track record. The last gusher of bad news prompted Yahoo! Finance to report last week that the shares are no longer cheap. What will happen if AutoZone and all its competitors crash? Will auto-parts sales fall and damage contracts? Maybe AutoZone's bottom line doesn't have to improve as quickly as the rest of the industry. Its stores are still filling up, and investors have much to consider before this roller-coaster gives way to better days. There will be happy days and slumps. Foolish bottom line: The funny part is that our hand is decidedly torn.

Fig. 14 PPLM negative sentiment news article generation

References

1. Adelani, D.I., Mai, H., Fang, F., Nguyen, H.H., Yamagishi, J., Echizen, I.: Generating sentiment-preserving fake online reviews using neural language models and their human-and machine-based detection. In: International Conference on Advanced Information Networking and Applications, Springer, pp. 1341–1354 (2020)
2. Agarwal, S., Farid, H., El-Gaaly, T., Lim, S.N.: Detecting deep-fake videos from appearance and behavior. In: 2020 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2020). https://doi.org/10.1109/WIFS49906.2020.9360904
3. Bastos, M., Farkas, J.: "Donald Trump is my president!": the internet research agency propaganda machine. Social Media+Society, pp. 1–13 (2019)
4. Bradshaw, S., Bailey, H., Howard, P.N.: Industrialized disinformation: 2020 global inventory of organized social media manipulation. Technical Report, Oxford Internet Institute (2021)

5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., pp. 1877–1901 (2020). https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
6. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Trans. Dependable Secure Comput. **9**(6), 811–824 (2012). https://doi.org/10.1109/TDSC.2012.75
7. Ciftci, U.A., Demir, I., Yin, L.: Fakecatcher: detection of synthetic portrait videos using biological signals. IEEE Trans. Pattern Anal. Mach. Intell. (2020). https://doi.org/10.1109/TPAMI.2020.3009287
8. Cohen, D., Croft, W.B.: End to end long short term memory networks for non-factoid question answering. In: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, pp. 143–146 (2016)
9. Cresci, S.: A decade of social bot detection. Commun. ACM **63**(10), 72–83 (2020). https://doi.org/10.1145/3409116
10. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: The paradigm-shift of social spambots: evidence, theories, and tools for the arms race. In: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp. 963–972 (2017). https://doi.org/10.1145/3041021.3055135
11. Dang, H., Liu, F., Stehouwer, J., Liu, X., Jain, A.K.: On the detection of digital face manipulation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, pp. 5780–5789 (2020). https://doi.org/10.1109/CVPR42600.2020.00582
12. Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., Liu, R.: Plug and play language models: a simple approach to controlled text generation. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=H1edEyBKDS
13. Dawson, A., Innes, M.: How Russia's internet research agency built its disinformation campaign. Political Q. **90**(2), 245–256 (2019)
14. Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A.S., Nemade, G., Ravi, S.: Goemotions: a dataset of fine-grained emotions. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, Association for Computational Linguistics, pp. 4040–4054 (2020). https://doi.org/10.18653/v1/2020.acl-main.372
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pretraining of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186 (2019)
16. Fagni, T., Falchi, F., Gambini, M., Martella, A., Tesconi, M.: Tweepfake: about detecting deepfake tweets. PLoS ONE **16**(5), e0251415 (2021)
17. Fan, A., Lewis, M., Dauphin, Y.: Hierarchical neural story generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, no. long papers, Association for Computational Linguistics, Melbourne, Australia, pp. 889–898 (2018)
18. Foster, D.: Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play, O'Reilly Media (2019)
19. Gao, J., Lanchantin, J., Soffa, M.L., Qi, Y.: Black-box generation of adversarial text sequences to evade deep learning classifiers. In: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, pp. 50–56 (2018)
20. Gehrmann, S., Strobelt, H., Rush, A.: GLTR: statistical detection and visualization of generated text. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Florence, Italy, pp. 111–116 (2019). https://doi.org/10.18653/v1/P19-3019. https://aclanthology.org/P19-3019
21. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, vol. 1, no. 12 (2009)
22. Goodfellow, I., McDaniel, P., Papernot, N.: Making machine learning robust against adversarial inputs. Commun. ACM **61**(7), 56–66 (2018). https://doi.org/10.1145/3134599
23. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, Conference Track Proceedings (2015). arXiv:abs/1412.6572
24. Hao, K.: A college kid's fake, ai-generated blog fooled tens of thousands. this is how he made it. https://www.technologyreview.com/2020/08/14/1006780/ai-gpt-3-fake-blog-reached-top-of-hacker-news/
25. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. In: 8th International Conference on Learning Representations (2020). https://openreview.net/forum?id=rygGQyrFvH
26. Ippolito, D., Duckworth, D., Callison-Burch, C., Eck, D.: Automatic detection of generated text is easiest when humans are fooled. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 1808–1822(2020)
27. Jawahar, G., Abdul-Mageed, M., Lakshmanan V.S., L.: Automatic detection of machine generated text: a critical survey. In: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, pp. 2296–2309 (2020)
28. Keskar, N.S., McCann, B., Varshney, L.R., Xiong, C., Socher, R.: CTRL: a conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858 (2019)
29. Krause, B., Gotmare, A.D., McCann, B., Keskar, N.S., Joty, S., Socher, R., Rajani, N.F.: GeDi: Generative discriminator guided sequence generation. arXiv preprint arXiv:2009.06367 (2020)
30. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
31. Liu, Z., Qi, X., Torr, P.H.S.: Global texture enhancement for fake face detection in the wild. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, Computer Vision Foundation/IEEE, pp. 8057–8066 (2020). https://doi.org/10.1109/CVPR42600.2020.00808
32. Nguyen, H.H., Fang, F., Yamagishi, J., Echizen, I.: Multi-task learning for detecting and segmenting manipulated facial images and videos. In: 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1–8 (2019). https://doi.org/10.1109/BTAS46853.2019.9185974
33. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, Los Alamitos, CA, USA, pp. 582–597 (2016). https://doi.org/10.1109/SP.2016.41

34. Pauls, A., Klein, D.: Faster and smaller n-gram language models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, pp. 258–267 (2011)
35. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
36. Rauchfleisch, A., Kaiser, J.: The false positive problem of automatic bot detection in social science research. PLoS ONE **15**(10), 1–20 (2020). https://doi.org/10.1371/journal.pone.0241045
37. Shao, C., Ciampaglia, G.L., Varol, O., Yang, K.C., Flammini, A., Menczer, F.: The spread of low-credibility content by social bots. Nat. Commun. (2018). https://doi.org/10.1038/s41467-018-06930-7
38. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
39. Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J.W., Kreps, S., et al.: Release strategies and the social impacts of language models. arXiv preprint arXiv:1908.09203 (2019)
40. Soutner, D., Müller, L.: Application of LSTM neural networks in language modelling. In: Habernal, I., Matoušek, V. (eds.) Text, Speech, and Dialogue, pp. 105–112. Springer, Berlin (2013)
41. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.), 2nd International Conference on Learning Representations, ICLR 2014 (2014). arXiv:abs/1312.6199
42. Torusdağ, M.B., Kutlu, M., Selşuk, A.A.: Are we secure from bots? Investigating vulnerabilities of botometer. In: 2020 5th International Conference on Computer Science and Engineering (UBMK), pp. 343–348 (2020). https://doi.org/10.1109/UBMK50275.2020.9219433
43. Uchendu, A., Le, T., Shu, K., Lee, D.: Authorship attribution for neural text generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 8384–8395 (2020). https://doi.org/10.18653/v1/2020.emnlp-main.673. https://aclanthology.org/2020.emnlp-main.673
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, U., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, pp. 6000–6010 (2017)
45. Wang, R., Juefei-Xu, F., Ma, L., Xie, X., Huang, Y., Wang, J., Liu, Y.: Fakespotter: a simple yet robust baseline for spotting ai-synthesized fake faces. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI) (2020)
46. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
47. Weiss, M.: Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. Technology Science (2019)
48. Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y.: Defending against neural fake news. In: Advances in Neural Information Processing Systems, pp. 9054–9065 (2019)
49. Zhang, W.E., Sheng, Q.Z., Alhazmi, A., Li, C.: Adversarial attacks on deep-learning models in natural language processing: a survey. ACM Trans. Intell. Syst. Technol. (2020). https://doi.org/10.1145/3374217
50. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.), Advances in Neural Information Processing Systems, vol. 28. Curran Associates, Inc. (2015). https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.