ARGH!: Automated Rumor Generation Hub

Larry Huynh*
University of Western Australia
Australia
larry.huynh@uwa.edu.au

Thai Nguyen
University of Western Australia
Australia
thain7127@gmail.com

Joshua Goh
University of Western Australia
Australia
joshgoh7@gmail.com

Hyoungshick Kim
Sungkyunkwan University
Republic of Korea
hyoung@skku.edu

Jin B. Hong
University of Western Australia
Australia
jin.hong@uwa.edu.au

ABSTRACT

It is still challenging to effectively identify rumors due to rapid changes in people's interests and perceptions. To enhance rumor detectors, we first need to better understand which rumors are effective (in terms of bypassing detection) and their characteristics. In this paper, we introduce ARGH, a novel framework to automatically generate rumors using recent advancements in natural language processing, customized to target and generate specific topics. To show the effectiveness of ARGH, we conducted a user study with 212 participants and analyzed how well humans can detect the rumors generated by ARGH, and we also tested its performance against the state-of-the-art rumor detection model PLAN [17]. Surprisingly, the experimental results demonstrate that the generated rumors are significantly harder to identify as rumors than hand-written rumors, degrading the detection accuracy by both humans and machines by 18.87% and 17.62%, respectively. We believe that ARGH will be a useful tool to obtain high quality and evasive rumor datasets quickly, which is often a tedious and time consuming task. Further, our analysis results provide valuable insight into how to characterize evasive rumors and how they can be generated, which will help to enhance the existing rumor detection techniques.

CCS CONCEPTS

• **Security and privacy** → **Social aspects of security and privacy**; • **Applied computing** → Law, social and behavioral sciences.

KEYWORDS

Misinformation; Rumor Detection; Rumor Generation

ACM Reference Format:
Larry Huynh, Thai Nguyen, Joshua Goh, Hyoungshick Kim, and Jin B. Hong. 2021. ARGH!: Automated Rumor Generation Hub. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3481894

1 INTRODUCTION

Rumors are prevalent in today's society, causing harms in many ways (e.g., financial, social and even physical). For example, amidst the COVID-19 pandemic, a rumor began circulating that injecting disinfectants could cure the virus. Although the rumor was quickly disproven, there were increases in calls to poison control centers regarding the ingestion and contact with bleach products which could be correlated with the dissemination of misinformation. Hence, there is of paramount importance to monitor, control, and limit the spread of rumors, which can negatively impact our society with respect to (but are not limited to) social and economic damages.

Developing a framework to aid in the detection of rumors has become an increasingly important objective. While rumor detection has been well explored using various machine learning (ML) techniques [40, 45], existing ML classifiers have suffered from low accuracy with new and unseen topics [18]. Further, confirming the veracity of rumors involves many steps [49], including detection, tracking, stance classification, and veracity classification. The detection step is most critical; if an actual rumor gets classified as non-rumor, then it is no longer tracked for validation. Our work is motivated to address the limitation of existing detection models, which lack the capabilities to detect rumors that are out-of-domain from their training data. Hence, we must comprehensively understand the characteristics of rumors when the context changes (e.g., the topic of interest) in order to improve rumor detection when applied to out-of-domain datasets.

To improve rumor detection, we need to better understand the characteristics of rumors that can be applied to out-of-domain data sets. To do so, we propose a rumor generation framework ARGH that automatically generates rumors to exploit the weaknesses of current rumor detectors.[1] In turn, this can be used to support rumor detection classifiers by providing better, pre-labeled training sets, as well as a better understanding of how evasive rumors are structured. Moreover, ARGH can generate customized rumors through fine-tuned features as parameters, producing text that looks like a legitimate, convincing *fact* (under the eyes of rumor detection

*Huynh, Nguyen and Goh are equally first authors.

[1] A pre-trained ARGH model is available from https://github.com/argh-rumor-detection/ARGH-Rumor-Generation

CIKM '21, November 1–5, 2021, Virtual Event, Australia.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481894

models and humans) tailored to the input parameters, while still containing misinformation. [2]

To evaluate ARGH, we conducted a user study with 212 participants to understand human perception on our generated rumors, and against a state-of-the-art rumor detection model PLAN [17] to understand how well existing solutions withstand rumors generated by ARGH. The results show that humans had significant difficulty in identifying generated rumors compared with written rumors with a drop in accuracy by 18.87%. Similarly, PLAN also struggled to detect our generated rumors with a drop in relative accuracy by 17.62%, highlighting potential limitations in current rumor detection models. Hence, ARGH could be useful for enhancing existing rumor detection solutions by providing evasive, synthetic rumor datasets. The contributions of our paper are as follows.

- We develop the first, to the best of our knowledge, automated rumor generation framework that can fine-tune features to generate highly customized rumors;
- We implement a set of metrics to characterize and evaluate generated rumors;
- We conduct a user survey with 212 participants to analyze how well humans can(not) differentiate rumors;
- We evaluate the performance of state-of-the-art rumor detection model PLAN [17] against our generated rumors and reveal its weakness.

2 RELATED WORK

2.1 Rumor Detection

We define a rumor for our context as unverified pieces of information in the short text format of social media posts, while rumor detection is defined as determining the veracity value of a rumor using automated machine learning systems [24]. Previous work explored the development of rumor detection solutions, which can be divided into two main categories: machine learning (ML) approaches and deep learning (DL) approaches.[3] Many ML approaches demonstrate deteriorated performance on unseen instances, and capturing these may require retraining on new datasets [18]. This poses a problem as rumors on unseen contexts can appear spontaneously. Hence, we focus more on DL approaches in this paper. These approaches typically tackle rumor detection as a classification problem (i.e., previous studies focused on developing solutions that can label unseen pieces of text as either rumor or non-rumor). We would also like to highlight that the rumor itself for ARGH is in the text, not the shared article. Hence, we cannot ensure that techniques for detecting fake news articles work well for rumor-like tweets due to size and style differences (i.e., existing fake news generation techniques have not been verified in their effectiveness when used for generating short-text rumors such as tweets).

DL frameworks have the advantage over ML frameworks in that they can learn hidden representations from raw inputs without over-reliance on manually crafted features [3]. For example, Chen *et al.* [7] presented an RNN model for early rumor detection, which focused on temporal relations between sequential social media posts for rumor classification. Comparisons between it and other state-of-the-art detection models showed promising performance (e.g., 0.87 F1 score). Ma *et al.* [28] proposed a generative adversarial network (GANs) approach in which two generative neural models were developed, one for distorting rumors to make them seem like non-rumors and one for the reverse task, in order to complicate the detection task and force the discriminator into learning stronger rumor indicative representations. However, as highlighted by Fedus *et al.* [11], GANs may find discrete language generations challenging since they were perform best on outputting differentiable values.

Ruchansky *et al.* [35] employed a model dubbed CSI, a hybrid DL model using LSTM for fake news detection. Unlike other DL approaches, CSI does not require the use of temporal, user, or structural features (i.e., the main focus is to use contents only), which is comparable to ARGH that focuses on the text itself for identifying rumors. The LSTM is fed with temporal data about patterns of user activity. Two datasets were tested, obtaining accuracies of 0.89 and 0.95, respectively. While these approaches showed promising results, only a few previous works reported on detecting unseen rumors. Furthermore, DL models can require huge datasets that may lead to the cost and availability of data issues [1]; problem we aim to address by proposing an effective rumor generator.

2.2 Natural Language Generation

The synthesis of generating rumors relies on a combination of computational tasks orientated around natural language generation (NLG). Hence, this subsection will explore existing machine learning architectures geared towards NLG.

The deep learning transformer model proposed by Vaswani *et al.* [41] has become the basis of contemporary NLP architectures, superseding previously proposed models. Notable transformers include the Bidirectional Embedding Representations from Transformers (BERT) model developed by Google [10] that uses pre-trained language representations from a large, unsupervised text corpus and can achieve state-of-the-art results on various NLP tasks. Wang and Cho [42] demonstrated that BERT could produce reasonably fluent and diverse unsupervised NLGs, comparable to the GPT language model. While we see the generative texts are of good quality, these samples show misuse of punctuation and an overall lack of semantic coherence. The GPT-2 model [33], developed by OpenAI, superseded the GPT model and outperformed BERT in various NLP tasks, including NLG. GPT-2 can learn NLP tasks without any explicit supervision. Radford *et al.* [34] further demonstrated GPT-2's ability to complete unseen contexts with high levels of quality and fluidity. Its performance is attributed to its training dataset size of 8 million web pages and its parameterization size of 1.5 billion.

2.3 Rumor Generation

While many ML models have been proposed for rumor detection, little work has been done in implementing generative models that produce convincing rumors.

The previously mentioned GANs generator by Ma *et al.* [28] somewhat incorporated this idea, where text generators were developed as a component of the GANs to feed in synthetically generated texts. The given examples of the generator text posts did show syntagmatic and paradigmatic coherence, albeit weakly. Another paper

[2] Here, the term *feature* refers to the intrinsic characteristics of a rumor, which can include topic, degree, user's sentiment, network structural and content features [9].

[3] Although DL falls under ML in context, many rumor detection surveys separate them as DL models do not require hand-crafted features [3, 30].

features the “rumor mill” developed by Inie *et al.* [14]. A physical, interactive “mill” is presented, which can take topics, degrees of believability, and tense as user parameters and output an automatically generated text rumor. Lastly, Grover was proposed by Zellers *et al.* [44], which generates fake news to improve the detection mechanism, a similar goal to our paper. However, Grover’s generated texts are raw outputs from GPT-2. where our user study later shows that using random outputs from GPT-2 is easily detected by humans (Section 5.2). This implies only “good” generations should be used, which was not explored in their study. The authors also stated poor performance on new, unseen rumors, as well as significantly degraded performance in handling short-form text. Grover is also very expensive both in cost ($35K USD for training each model vs. free GPT-2) and in training time (two weeks of training), making it impractical for multiple topics. Further, Grover’s rumor detection method is focused on whether the text is generated by human or machine, not the context – leading to low accuracy as machine-generated texts get better. Lastly, Grover needs extensive human labor to direct generation, as stated by the authors, whereas our approach has been set up to be autonomous and can be scaled up easily. Hence, Grover’s limitations highlighted above are addressed in this paper.

3 ARGH: THE RUMOR GENERATOR

Despite recent notable advancements in the NLG field, its use in rumor generation is mostly under-explored. Rumor generation is not simply auto-generating false information; our proposed framework can customize the rumor in terms of the topic/event of interest and the fitness of generated rumors. For example, OpenAI, the creator of GPT-3 and many other ML/DL-based frameworks, has stated that “tweaking language model outputs to consistently generate convincing template messages without significant human oversight is still difficult.” [39]. Hence, this work focused on how to generate convincing template messages (i.e., rumors) without significant human oversight, rather than advances in technical details such as performance. To do this, we propose ARGH, a rumor generation pipeline that generates rumors of content conforming to user input in microblog posts. It consists of the four main steps: (1) data collection, (2) feature extraction, (3) rumor generation, and (4) metrics and filtering for measuring the goodness of generated rumors. We allow further customization of output rumors with respect to topics and/or fitness of information to generate rumors for targeted purposes. These steps and their relations are depicted, as shown in Figure 1, the following subsections describe each step in detail.

3.1 Data Collection

Using Twitter’s API, we collected tweets about various real-world events, including the COVID-19 pandemic, the Beirut explosion, and the 2020 US presidential election, and collated each of them into their own topic dataset. We used 5000 tweets for each. This relatively low number of tweets was utilised to simulate real world scenarios wherein users have limited data available, and demand datasets describing newly emerging topics for use in early rumor detection models. Each topic was chosen due to its recency (from the time of data collection) and overall popularity on Twitter, as confirmed by Twitter’s trending results page and search query analysis tools such as Google Trends[4]. Further, we obtained roughly two million tweets relating to various news events during the period of 2016 – 2017. Proposed by Shwartz *et al.* [38] for the purpose of predicate paraphrase extraction, we utilize this large-scale Twitter corpus in the fine-tuning of ARGH to provide it with the general syntactic form of a tweet. These datasets are used in fine-tuning the rumor generator (Section 3.3).

3.2 Feature Extraction

We create the processed dataset by calculating the term frequency-inverse document frequency (TF-IDF), and the word frequency per n-gram in each tweet in the Twitter dataset [23]. The TF-IDF score determines the importance of a word in a document, which is proportional to the number of times a word appears in a corpus but is offset by the frequency of the word in the document. This gives an indicator of the most topical words within the tweet. The word frequency is the same but without the offset, capturing the more diverse and unique word choices within tweets. This forms the basis of feature extraction; we build a list of keywords that are strongly representative of each tweet.

We noticed the use of monograms allowed for the best generalizations from the model compared to bigrams or above. This may be true to the typical short length of a tweet; we cannot extract enough higher-order n-grams without overlapping their internal monograms, causing an over-representation of those monograms. Thus, we used these monograms as keywords for fine-tuning.

Words are then chosen from the keywords list, half of which are chosen based on highest word frequency, the other half based on highest TF-IDF. The number of keywords used can be varied; for our experiments, eight were used to reflect the average number of non-stop words found in the two million tweets dataset. We use the combination of these words as the context in our prefix in order to build the most encapsulating representation of the source tweet. The context keywords are randomly shuffled to remove word order as a feature. We then concatenate them with the source tweet in addition to special tokens for delimiters and the start/end of the text. Keyword extraction methods, keyword order, and the number of keywords can be further explored for their effect on generation. Finally, we use the processed tweets as training data for fine-tuning GPT-2 [34]. Similar feature extraction is used for generating from a reference tweet; however, instead of TF-IDF and word frequency extraction, we use parts of speech (POS) tagging to extract the most relevant words from the tweet. The generated output is hence a continuation of the keywords. Users are also able to specify their own keywords to adjust the linguistic attributes of the output.

3.3 Rumor Generation

For rumor generation, we use a single input model adaptation [12], utilising GPT-2 as the pre-trained language model [33]. GPT-2 was chosen for its well-regarded ability to generate coherent text, and accessibility [34]. Fine-tuning and generation experiments with GPT-2 were done using free public resources. We train GPT-2 to generate output rumors whose content is transferred from the given input domain. First, we create our generic model by fine-tuning GPT-2 on the two million tweets dataset using the output from

[4] https://trends.google.com/trends/

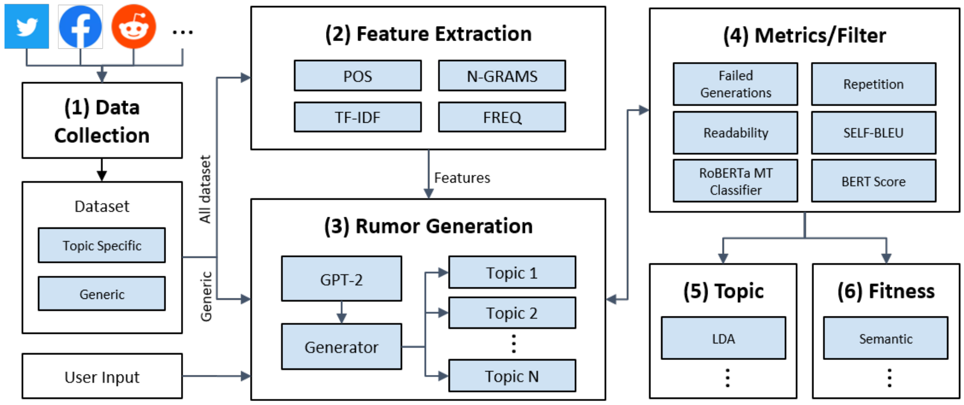

Figure 1: ARGH framework.

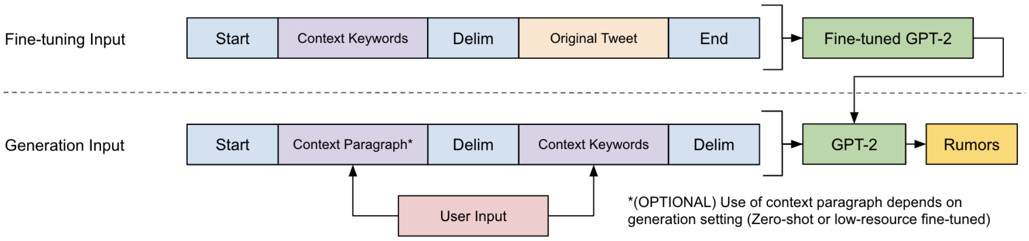

Figure 2: ARGH input format.

the feature extraction module as the fine-tuning input shown in Figure 2. The input enables GPT-2 to learn to generate rumors based on the keywords given, learning the semantic representation through the keywords and the syntactic form through the tweets. As shown in Figure 2, the generation input is the incomplete fine-tuning input where the subsequent generated tweet is decoded as a continuation by GPT-2 [33]. This generic model can be used in a zero-shot setting[5], where in addition to the context keywords, a contextual paragraph is prepended to be inserted into GPT-2 context window to generate tweets that are out of domain with what GPT-2 was fine-tuned on. The generic model can also be further fine-tuned on a few in-domain tweets to be used in a low resource fine-tuning setting.

3.4 Metrics and Filter

We require the generator output to be of acceptable quality before we consider its evaluation on rumor detection. We employ six NLG metrics covering various aspects to assess the performance of ARGH.

3.4.1 Failed Generations. A “failed” generation refers to any generation that is: (1) still contains the special or class tokens supplied to the generator in fine-tuning and input modeling, or (2) is uncharacteristically lengthy for a microblog post. In our test cases using tweets, we consider a failed generation to be over 280 characters [6]. These conditions can be adjusted for the various character limits and other conformity regulations of other social media outlets. Here, we define the proportion of failed generations PFG as $PFG = \frac{FG}{TG}$, where FG is the number of failed generations within the batch and TG is the total number of generations. Within a batch, this proportion gives a measure of the efficiency of the generator (i.e., minimizing this metric means better generations come out more often).

3.4.2 Repetition. Repetition refers to the generator’s propensity to output text ending with repeating tokens and phrases. Such texts are considered to be of poor quality due to their reduced coherence. An example of a repetitive text would be “people infected with coronavirus coronavirus coronavirus…” The proportion of repetitions within a corpus of generations PRG is defined as $PRG = \frac{RG}{TG}$, where RG is the number of generations that show repetition within the batch and TG is the total number of generations.

3.4.3 Readability. Readability refers to the sum of the text elements that affect a reader’s understanding, reading speed, and interest level in the text material [8]. We use the Flesch Reading Ease Score (FRES) [8] to determine the readability of our generations due to its wide availability, ease of use, and ability to be used in more generalized contexts compared to other tests. Typically, a higher FRES indicates the given text material is easier to read, while lower numbers indicate more challenging to read passages.

3.4.4 Self-BLEU. We use Self-BLEU [48] as a metric (denoted as Self-BLEU Score, or SBS), to both evaluate the diversity of the generations and to benchmark generations relative to other generation

[5] This is to generate rumors on topics and events it never was trained on.
[6] The character limit for Twitter.

sets. Self-BLEU is an extension of the BLEU score[7], which assesses the n-gram similarity between two sentences. Self-BLEU is calculated by computing the BLEU score of each generation against every other generation as references, then averaging the outcomes. SBS is measured on a 0 to 1 scale; a lower score indicates greater diversity within the corpus.

3.4.5 RoBERTa Detection Model. The RoBERTa Detection Model (RoBERTa) [39] is a masked and non-generative language model developed and fine-tuned by OpenAI for the task of classifying machine and human-written text. We use the RoBERTa base model (125 million parameters), which has shown a classification accuracy of 96.6%[8] over GPT-2 medium, to classify our generations. RoBERTa score is given in between 0 and 1; values above 0.5 return the $human - written$ label, else the $machine - written$ label is returned. We term this score as the *Individual RR*. Since every text in the generation corpus is machine-written, the performance can be measured by the number of generations correctly labeled as machine-written over the total number of generations, calculated as $RR = \frac{MLB}{TG}$, where RR is the RoBERTa rate, MLB is the machine labeled generated texts, and TG is the total number of generations. Thus a lower RoBERTa rate is a desirable outcome (i.e., generated rumors are more human-written like).

3.4.6 BERT Score. BERT Score [46] is an automatic evaluation metric for text generation, computing the semantic cosine similarity between sentences using contextual embeddings. This allows it to correlate better with human judgements compared to existing metrics. BERT Score is given from -1 to 1; 1 being an exact semantic match. We use it as an assessment of the conformity of a generation to the user input, where it is desirable for a generation to conform to the meaning of the input. We also use BERT Score as an indicator of the quality of a generation; a higher BERT Score is more likely to correlate with generations of sound coherence, while a low BERT Score captures generations of poor construction or even linguistic noise.

3.4.7 Use of Metrics. First, Readability (Section 3.4.3), SELF-BLEU (Section 3.4.4) and RoBERTa (Section 3.4.5) metrics are used for fine-tuning the generator. Once fine-tuned, we generate rumors using the metrics as follows.

(1) Remove failed (Section 3.4.1) and repetitive (Section 3.4.2) generation texts.
(2) Rank generations using BERT Score (Section 3.4.6).
(3) Filter generated rumors using Readability (Section 3.4.3), SELF-BLEU (Section 3.4.4) and RoBERTa (Section 3.4.5).

Then, we rank generations from best to worst performing using the BERT Score, acting as a filter for successful (i.e., not failed) but under-performing generations. The combination of our other metrics allows us to corroborate the BERT Score's indications for quality generations. The use of Readability, SELF-BLEU, and RoBERTa metrics can further refine the output rumors to enhance the evasiveness. The threshold value for each metric can be defined by the users depending on their requirements (e.g., the individual RR value to be greater than 0.9).

3.5 Topic

For further customization of generated rumors, we implement a topic module that scopes the topic of rumors generated using ARGH. A trained Latent Dirichlet Allocation (LDA) model [2] is used because it is currently considered the state-of-the-art solution for the topic modelling due to its proven ability to reliably generate topics as well as infer topics from unseen documents [31]. When a generation corpus has been prepared, the corpus is used to train the LDA model, and its topic distribution is returned. This distribution can be analyzed to gain insight into what influence topic selection has over the output of ARGH.

3.6 Fitness

We implement a fitness module to analyze the similarities between our generated tweets and real tweets that exist within the Twitter ecosystem. The goals of this analysis are two-fold; (i) to improve our generated output's evasiveness, and (ii) discover flaws in the generations that inform better rumor and machine-text detection strategies. We use the metrics from previous modules (i.e., Section 3.4.6), linguistic features (i.e., sentiment, punctuation), and content features (i.e., hashtag use) to map their distribution on sets of real and generated tweets. These distributions are used to find inputs and configurations that produce the most evasive rumors given specific requirements (i.e., topic, conveyed message). Additionally, they can reveal the biases that are currently plaguing language models [36], allowing for methods to be developed to mitigate them [37]. Overall this module provides insight into how the input influences the features present within the generated rumors, subsequently informing approaches for calibrating the input to produce desired outputs.

3.7 Combining Steps

Our focus is on the rumor detection application, which is a critical issue in the field of social media. Some may point out that many of the components are existing work, but to the best of our knowledge, there is no existing work that provides a customizable rumor generation framework that is effective, fast, and cost-saving (the most similar current work being the Grover generative model; cost and time-performance comparisons have been made in Section 2. We suspect this is because it is challenging to combine the above steps to produce effective results, strongly indicating the difficulty in training and fine-tuning the models. This is further validated by our user study (Table 2), which has not been conducted before, as well as generating datasets and experimental methods for the evaluation of rumor detection (i.e., synthetic dataset generation, which are scarce and rare).

3.8 Validity of Generated Rumors

To validate that our generated texts are indeed rumors, we employ both semantic similarity measures and manual inspection. Previous work demonstrated the ability of semantic similarity in correlating annotated labels across both reference and candidate texts (i.e., if two texts show substantial semantic similarity, they are highly likely to be classified with the same rumor/non-rumor label) [13]. Similarly, we evaluated the outputs from ARGH, and the generated texts achieved a high semantic similarity score (i.e., BERT Score),

[7] BLEU score is a well-known NLP metric; the full scope of its computation can be found in [32].
[8] As shown in https://openai.com/blog/gpt-2-1-5b-release

thus inherently preserving the rumor label across our generations similarly as found in [13]. However, this approach assumes that semantic similarity score is accurate, where there are cases with texts of opposite meaning that may still exhibit high semantic similarity. An example is shown below, which is evaluated to a high BERT Score of 0.9182, but as a human reader, the context is vastly different (the latter is generated).

- #BREAKING Armed man takes hostage in kosher grocery in Paris: source
- BREAKING: Armed man held hostage at kosher grocery store in #Paris. #Grocery #Hostage

To overcome this issue, we additionally inspected 500 randomly selected sample generations manually, each sourced from rumor-labeled inputs. In our random sampling inspection, none of these were found to be facts (i.e., they were all rumors). Using the Clopper-Pearson Exact test, we found a mean to be between 0 - 0.010541 at the 99% confidence level (i.e., we may find one non-rumor (fact) generated out of 50,000 generated texts from ARGH).

4 CASE STUDIES: SYDNEY SIEGE (FROM PHEME) & COVID-19

Examples of generated outputs and their associated metrics are shown in Table 1. Two sets of examples are given: one *Human Written Rumor*, one *Good Generated Rumor*, and one *Bad Generated Rumor* are shown each for the topics of the Sydney Siege (first row) and COVID-19 (second row).

Human-Written Rumor: We observe a *PRG* of 0% as there is no trailing repetition present (as with all other examples in Table 1). We see acceptable *FRES* scores (78.59 and 62.04), indicating both texts are relatively easy to read. The *BERT Scores* are 1.0 as they are calculated relative to the ground-truth *Human Written Rumor*. Note, however, that the *Individual RR* fails to classify these texts as human-written correctly.

Good Generated Rumor: These generations demonstrate the ability of ARGH to generate high-quality outputs. The *FRES* scores are lower than the human-written cases above, indicating greater complexity. The high *RR* value in the COVID-19 shows the generation's strong evasiveness against the RoBERTa (i.e., they appear to be human-written), while the Sydney Siege generation performed poorly. Despite this, both the Sydney Siege and COVID-19 generations were able to evade detection from both PLAN and human evaluators, respectively. Finally, the high *BERT Scores* show a strong semantic similarity between the original and generated output (i.e., aligns well with the given topic of interest). These metrics indicate that these are high-quality texts in terms of good writings done by humans.

Bad Generated Rumor: On the other hand, ARGH can also generate poor quality texts if not filtered using the described metrics. The lower *FRES* in the Sydney Siege generation indicates the text is complicated to read. However, both the *BERT Scores* are relatively high, meaning they are still relatively within the topic being presented originally. In both examples, their contents can easily be detected by human readers as a misinformation/rumor.

This case study shows that ARGH can generate rumors that contain human-like writings (as shown by metric values), but has to be of acceptable quality (i.e., be a *Good Generated Rumor*). Moreover, individual metrics are not robust enough to capture all aspects of a text to determine whether it is *good* or *bad* generated texts. An in-depth analysis of ARGH is presented in the next section to evaluate its performance under various conditions.

5 EXPERIMENTAL ANALYSIS

For the analysis, we conduct (1) a user study to understand human perception and ability to identify rumors (Section 5.2), and (2) a test against the state-of-the-art rumor detection model to observe the change in bypass rate (Section 5.3).

5.1 Experimental Setup

We used the OpenAI's 345M GPT-2 model [33]. For fine-tuning and generating with GPT-2, we utilized Google Colab [9] free Jupyter notebook environment with an NVIDIA Tesla P100 GPU with 16GB of memory. Colab's free tier was sufficient to liberally experiment with different datasets and configurations. For calculating the metrics and training the rumor detection model PLAN, we use Gradient Paperspace [10] which provides a cloud environment for deep learning. We hired an NVIDIA Quadro P5000 for our experiments at an hourly rate of $0.78 USD. Given that the trained model can be used on free-tier Colab, mass-generation of rumors using ARGH without any cost is feasible.

5.2 Public Perception on Rumors: A User Study

In this study, we examine three different topics (i.e,. COVID19, US politics and the 2020 Beirut explosion) that have been trending keywords globally over time. An anonymous survey utilizing Amazon Mechanical Turk (MTurk) was carried out, where responses were restricted to workers from the United States[11] with over 10,000 Human Intelligence Tasks (HITs) completed on MTurk and at least a 97% approval rate. For each topic, there were 11 tweets split into four categories: three rumors generated by ARGH, three written rumors, three non-rumors, and two bad rumors generated by ARGH. Generated rumors were considered "bad" if it was clear they were not coherent (as shown in Section 4 bad generated rumors). For each tweet, workers had three options to choose from: "This is a rumor," "This is NOT a rumor," and "I'm not sure." We received 212 valid responses, totalling 2,332 unique responses to our 11 questions. The validity of the survey responses was determined using three main methods: (i) survey completion time should be longer than five minutes (ii) a valid MTurk worker ID must be provided, and (iii) a provided validation code must be entered into MTurk by the worker after completing the survey. Responses not meeting the above were deemed invalid and not approved. With an average completion time of about 10 minutes, we paid workers above the US minimum wage at $1.25 USD per HIT.

For the analysis, we only considered the number of correct and incorrect responses for each tweet organized by topic and category. All "I'm not sure" responses were not included in the analysis to compare only the responses that participants were confident about. The results are shown in Table 2, where data is grouped by category

[9] Can be found at https://colab.research.google.com/

[10] Can be found at https://gradient.paperspace.com/

[11] One particular topic was related to the US politics, so we restricted participants from the US only.

Table 1: Metrics applied to examples of human written rumors, and good and bad quality generations.

Type	Text	PRG	FRES	Individual RR	RR Predicted Label	BERT Score
Human-Written Rumors	#breaking We have now heard that all airspace in the #Sydney city has been shut down. #martinplace #Lindt cafe	0	77.57	0.671586	human-written	1.0
	Our current federal government is dysfunctional and incompetent. It couldn't fight off the virus. In fact, it didn't even see it coming. The European virus infected the Northeast while the White House was still fixated on China.	0	62.04	0.377284	machine-written	1.0
Good Generated Rumors	I'm in Sydney. I hear that all the #Sydney City airspace has been shut down. #MartinPlace	0	80.28	0.126336	machine-written	0.892171
	A coronavirus vaccine, successfully tested in clinical trials, has been registered by Russia and given to 8,000 people in the country	0	33.24	0.678400	human-written	0.924138
Bad Generated Rumors	Sorry Lindt Sydney! #MartinPlace #Australia #Sydney #Airspace #Australian #News	0	28.50	0.215004	machine-written	0.833439
	Coronavirus has failed to register as a Cure in the World Health Organization's coronavirus database, according to an analysis by Vaccine World	0	40.69	0.569893	human-written	0.834257

Table 2: Comparing *Correct* and *Incorrect* responses per *Category* per *Topic*.

	Generated Rumors			Written Rumors			Non-rumors		
	Correct	**Incorrect**	**Accuracy**	**Correct**	**Incorrect**	**Accuracy**	**Correct**	**Incorrect**	**Accuracy**
COVID-19	340	180	65.38%	413	151	73.22%	340	181	65.26%
US Politics	298	222	57.31%	367	199	64.84%	312	145	68.27%
Beirut Explosion	199	289	40.78%	447	96	82.32%	252	197	56.12%
All	837	691	54.56%	1227	446	73.43%	904	523	63.25%

(generated, written, or non-rumors) and topic. On average, the generations achieved 18.87% better accuracy than written rumors and 8.69% better than non-rumors, implying that if our generated rumors were released to the public, they potentially have a higher likelihood of propagation than both written rumors and non-rumors due to their higher believability.

We also can see the clear differences between rumors of different topics, as shown in Figure 3. Across all three topics, our generated rumors performed substantially better than written rumors; however, it is evident that there are large differences depending on the topic. For example, with COVID-19, we can see that a higher number of generated rumors were identified compared to the other two topics. This is likely because COVID-19 is a popular and established topic that people are more aware of, which could means that rumors about COVID-19 are more likely to be detected by humans. On the other hand, US Politics could be more susceptible to rumors where people's personal interests can promote biased decisions, resulting in a lower score. The Beirut Explosion was an emerging topic of less direct impact to participants compared with the others, which could mean they were less informed about the event's circumstances. Not being aware of the specific details could mean rumors of this topic are harder to identify. Statistical analysis of the results seemed to corroborate the above conjecture; we used Fisher's Exact Test [29] with Bonferroni correction applied to eliminate Type I errors (i.e., at a significance level of 0.05, we only accepted P values less than 0.017).

The result for all pairs (i.e., generated vs. written, generated vs. non-rumors and written vs. non-rumors) were significantly different (i.e., the p values were less than 0.01) apart from COVID-19 generated vs. non-rumors and US Politics written vs. non-rumors.

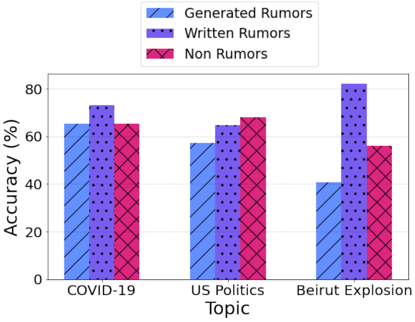

Figure 3: Survey scores per topic.

The result indicates that the generated rumors from ARGH are more difficult to detect than hand-written rumors by humans regardless of the topic, recency or popularity.

5.3 Bypassing Automated Rumor Detection Model PLAN

We evaluated the performance of ARGH to bypass the state-of-the-art rumor detection model PLAN [17]. PLAN is selected for evaluation since it has reportedly outperformed existing state-of-the-art rumor detection models, including PTK by Ma *et al.* [26], RvNN by Ma *et al.* [27], MTL2 by Kochkina *et al.* [20], and Tree LSTM by Kumar and Carley [21]. Further, we use the dataset that PLAN performed the best with (i.e., Twitter16 [38]) so that ARGH is evaluated against the best rumor detection model currently available.

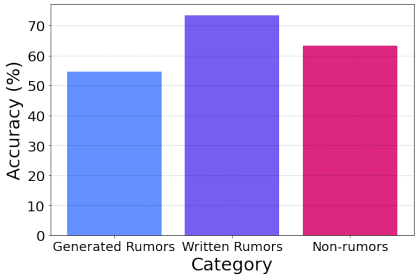

Figure 4: Survey scores per category.

5.3.1 Dataset: We use two publicly available datasets for the experiment; PHEME [20] and Twitter16 [38]. To only examine the effectiveness of our generated rumors, we test using only the original rumors in the datasets. PHEME contains a total of 1972 rumor tweets across 5 events, and Twitter16 contains a total of 498 rumor tweets across various events collected between March to December in 2015.

5.3.2 Metadata Population: Since ARGH only outputs rumors without any metadata or replies, we designed a dataset swapping technique for evaluations. For this data swapping to be legitimate when testing on the detector, we needed to use generated tweets close in meaning to the original tweet, whereby the context for the replies stays the same or is relatively similar. Firstly, we fine-tuned our generic model on the respective datasets and used each processed tweet we were swapping as the user input to generate similar tweets. For each tweet, a set of 100 successful generations were produced. We then chose the generation with the highest BERT Score against the reference tweet as the generation to swap in and use the generation alongside the original tweet's metadata as the input for the detector. As an example, two randomly selected swaps are shown in Table 3. Since ARGH is a rumor generator, we only swap on tweets labeled rumors. This technique's main limitation is the reliance on BERT Score as the semantic similarity measure between the two tweets, which is not completely robust. This can lead to mismatching of the generated rumor and the metadata it is using. However, since the generation is still a rumor, the crowd signals that PLAN uses for rumor detection are still relevant, which is proven by the detection results on just the metadata. Overall, by testing our generations with metadata from similar contexts, we can simulate how well they would perform against existing rumor detection.

5.3.3 Evaluation: We test our rumor generator against the post-level attention (PLAN) model presented in [17]. While they also produced other variants (StA-PLAN, sTA-HiTPLAN, etc.), PLAN performed the best on the Twitter16 dataset, which we used for training the model. Unfortunately, PLAN requires each tweet to have a corresponding response to determine whether it could be a rumor or not (i.e., post-level).

Table 4 shows the result of rumor detection using PLAN [17]. For both datasets (i.e., PHEME and Twitter16), we observed an improvement in lowering the detection rate using our generated rumors. We also confirmed a statistical difference between the original and generated rumors using Fisher's Exact Test on all datasets (the statistic value is 0.0465, so the result is significant at $p < .05$). The decrease in accuracy for generated rumors may seem marginal (within 3% for each dataset and 3.19% for all datasets), but this is due to metadata being part of the rumor detection. To distinguish the effects of metadata affecting the rumor detection accuracy, we also tested *Empty* datasets, which are generated by removing all data (e.g., tweets) information while retaining all metadata information (i.e., the rumor detection carried out only on metadata information). These results are presented under *Empty* columns in the table, which shows high accuracy that leaves a small gap for improvements. To find the relative improvement of our generated rumors without considering the effects of metadata to tweets we replaced, we apply the equation (1) to find the relative detection reduction. Here, g is the relative detection reduction, and $acc(x)$ represents the detection accuracy for the input dataset x. We then calculate that the relative detection reductions for generated rumors are 10%, 6%, and 18% for PHEME, Twitter16, and All datasets, respectively. That is, roughly 1 in 5 generated rumor was able to bypass the PLAN's rumor detection. Given a mass amount of rumors can be generated using ARGH (e.g., in thousands), this proportion is still alarmingly high.

$$g = \frac{acc(original) - acc(generated)}{1 - acc(empty)} \tag{1}$$

6 DISCUSSION

Improving rumor detection: ARGH provides a novel solution for filling known gaps in rumor detection research. As rumor detection is currently confined to established rumor datasets, supervised detection methods suffer from data scarcity issues, especially for obscure events. ARGH allows users to populate more samples for these events, augmenting existing datasets.

We demonstrated in Section 3.8 that we can preserve the rumor labels of inputs to ARGH. If a user can provide even one labeled rumor input of an unknown topic, they would be able to extend their rumor dataset by orders of magnitude. As such, for any new trending topic, we can use our generator to feed in new potential rumors and reinforce detectors, providing rich training sets quickly for unforeseen rumors.

Metrics for evaluating generated text: Existing evaluation methods for NLG consist of three categories; human-centric evaluation, untrained automatic metrics, and machine-learned metrics [6]. For our experiments, we utilize all three categories to evaluate generated rumors from ARGH. Each of these methods has its own limitations, which make it difficult to validate our results extensively. Human-centric evaluation is expensive, time-consuming to run, and recruitment systems, such as MTurk, have issues maintaining quality control. Although we tried to minimize these issues through careful screening of the survey results, human evaluation is inconsistent due to the individuals varying exposure to the context of rumor detection [47]. We have also demonstrated a lack of robustness with both our untrained automatic metrics (FRES, SBS) and machine-learned metrics (RoBERTa, BERT Score), as discussed in Section 4. To the best of our knowledge, a sound and robust metric for text quality that highly correlates with the human judgement within the NLP field does not yet exist; automated assessment of quality is subject to error until one is devised.

Table 3: Random samples of generated rumors for swapping.

Original Rumor	Generated Rumor	BERT Score
White House: Obama awaiting chance to speak with Harper about Ottawa shooting	White House: Obama will speak with Prime Minister Harper about the shooting. Hope he will speak with the PM	0.930525
STORY: Prince to reportedly play a pop up show in Toronto TONIGHT	"Prince reportedly plays a show tonight in Toronto."	0.929208

Table 4: Comparing *Original*, *Generated* and *Empty* datasets for rumor detection.

	PHEME			Twitter16			All		
	Original	**Generated**	**Empty**	**Original**	**Generated**	**Empty**	**Original**	**Generated**	**Empty**
Correct	1784	281	1623	130	128	118	1914	409	1741
Incorrect	188	36	349	24	26	36	212	62	385
Accuracy	90.47%	88.64%	84.42%	84.42%	83.12%	76.62%	90.03%	86.84%	81.89%

Machine text detector: Machine-generated text is difficult to distinguish from human-written text [15], which has driven the need for machine text detectors such as RoBERTa (Section 3.4.5). However, its performance over shorter and noisy text such as microblog posts is limited [39]. This diminished accuracy results in a reduced ability for these models to meet the required task, allowing rumor generators such as ours to proliferate misinformation with an inexhaustible capacity compared to human actors. Alongside developing better rumor detectors, developing better machine text detectors should also become an urgent priority.

Other transformers: GPT-2 has been used widely in the research community due to its robust performance in high-quality text generations [5, 19, 22, 25]. However, other transformers exist, such as CTRL [16] and XLNet [43]. While those alternative transformers could be used, we found GPT-2 to be more widely accessible while providing agnostic fine-tuning capabilities. In addition, the GPT-3 transformer was made publicly available in June 2020 [4], which is an extension of GPT-2 with enhanced NLP-related tasks.[12]

Lack of rumor detection models accessibility: There are various rumor detection models in contemporary literature. However, their models are often difficult to implement and replicate their results (e.g., dataset transformation is missing or unclear, provided code does not compile, etc.), not to mention their lack of accessibility in some cases. Each section of the implementation pipeline is often highly-specified, where one missing or failed component renders the whole model unusable. Thus, we have only been able to evaluate our results using PLAN [17]; future work will involve exploring other rumor detection models as they become available.

Ethical Concerns and Implications: As evidenced by our results, our generative model is able to outperform the state-of-the-art PLAN detection model and subvert the human capacity for rumor recognition. As described in our discussion, a superior generative model has the potential to accelerate training and development of superior detection models. The creation of performant generative models entails important ethical considerations as these models can be utilised in a malicious capacity by bad actors; intentionally producing rumors for the purpose of spreading misinformation. This is an inherent risk posed by a generative model that outperforms detection models. However, we believe that the ability to rapidly generate training data in an automated fashion enables a degree of performance increase in detection models that justifies this risk. In consideration of these implications, while we have made a demonstration available via GitHub, we have not published the codebase and datasets used in producing the generative model.

7 CONCLUSION

The spread of rumor is becoming more prevalent, causing a significant impact on our daily lives. Many rumor detection techniques are presented but fail to achieve high performance when out-of-domain rumors are presented. Further, our user study found that humans are more susceptible to believing rumors, causing higher polarization and antisocial behaviors. In this paper, we proposed an automated rumor generator ARGH that can be customized to generate highly evasive rumors. This has achieved an overall 18.87% accuracy drop by humans, and 17.62% accuracy drop by the state-of-the-art rumor detection model PLAN, respectively. Our results highlighted some key issues: (1) there is a need for improving existing rumor detection models in their performance on out-of-domain rumors, (2) we cannot rely on the public to distinguish rumors effectively, and (3) as generated rumors successfully penetrate human perceptions; rumor detection models relying on people’s stance may be affected significantly (e.g., a false rumor may be labeled as true rumor or unverified). To overcome these issues, ARGH efficiently generates highly evasive synthetic rumor datasets with customizable topics to align with currently trending topics that can improve the performance of rumor detection models and better understand how evasive rumors are generated and spread. Our findings provide useful insights to construct a more robust rumor detection framework that could effectively work against out-of-domain and highly evasive rumors.

ACKNOWLEDGEMENT

This research was supported by the High-Potential Individuals Global Training Program (2020-0-01550) funded by the Korea government.

[12] But it is more costly for public adoption. A better estimate can be found from beta.openai.com/pricing

REFERENCES

[1] Mohammed Al-Sarem, Wadii Boulila, Muna Al-Harby, Junaid Qadir, and Abdullah Alsaeedi. 2019. Deep learning-based rumor detection on microblogging platforms: A systematic review. *IEEE Access* 7 (2019), 152788–152812.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Alessandro Bondielli and Francesco Marcelloni. 2019. A survey on fake news and rumour detection techniques. *Information Sciences* 497 (2019), 38–55.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[5] Paweł Budzianowski and Ivan Vulić. 2019. Hello, It's GPT-2-How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. In *Proc. of the 3rd Workshop on Neural Generation and Translation (WNGT 2019)*. 15–22.

[6] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of Text Generation: A Survey. arXiv:2006.14799 [cs.CL]

[7] Tong Chen, Xue Li, Hongzhi Yin, and Jun Zhang. 2018. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2018)*. Springer, 40–52.

[8] Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics* 165, 2 (2014), 97–135.

[9] Anh Dang, Abidalrahman Moh'd, Aminul Islam, and Evangelos Milios. 2019. Early detection of rumor veracity in social media. In *Proc. of the 52nd Hawaii International Conference on System Sciences (HICSS 2019)*.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019)*.

[11] William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. MaskGAN: Better text generation via filling in the_. *arXiv preprint arXiv:1801.07736* (2018).

[12] Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. 2019. Large-scale transfer learning for natural language generation. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*. 6053–6058.

[13] Sooji Han, Jie Gao, and Fabio Ciravegna. 2019. Neural Language Model Based Training Data Augmentation for Weakly Supervised Early Rumor Detection. In *Proc. of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2019)*. 105–112.

[14] Nanna Inie, Jeanette Falk Olesen, and Leon Derczynski. 2020. The Rumour Mill: Making the Spread of Misinformation Explicit and Tangible. In *Proc. of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA 2020)*. 1–4. https://doi.org/10.1145/3334480.3383159

[15] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. arXiv:1911.00650 [cs.CL]

[16] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).

[17] Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable Rumor Detection in Microblogs by Attending to User Interactions. In *Proc. of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*.

[18] Younghwan Kim, Huy Kang Kim, Hyoungshick Kim, and Jin B. Hong. 2020. Do Many Models Make Light Work? Evaluating Ensemble Solutions for Improved Rumor Detection. *IEEE Access* 8 (2020), 150709–150724.

[19] Tassilo Klein and Moin Nabi. 2019. Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds. arXiv:1911.02365 [cs.CL]

[20] Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task Learning for Rumour Verification. In *Proc. of the 27th International Conference on Computational Linguistics (COLING 2018)*. 3402–3413.

[21] Sumeet Kumar and Kathleen M Carley. 2019. Tree LSTMs with convolution units to predict stance and rumor veracity in social media conversations. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. 5047–5058.

[22] Jieh-Sheng Lee and Jieh Hsiang. 2019. Patent Claim Generation by Fine-Tuning OpenAI GPT-2. arXiv:1907.02052 [cs.CL]

[23] Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md. Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. 2011. Twitter Trending Topic Classification. In *Proc. of the 11th IEEE International Conference on Data Mining Workshops (ICDMW 2011)*. 251–258. https://doi.org/10.1109/ICDMW.2011.171

[24] Quanzhi Li, Qiong Zhang, Luo Si, and Yingchi Liu. 2019. Rumor detection on social media: datasets, methods and opportunities. *arXiv preprint arXiv:1911.07199* (2019).

[25] Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. GPT-based Generation for Classical Chinese Poetry. arXiv:1907.00151 [cs.CL]

[26] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

[27] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1980–1989.

[28] Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *Proc. of the Web Conference (WebConf 2019)*. 3049–3055.

[29] John H McDonald. 2014. *Handbook of Biological Statistics (3rd ed.)*. Sparky House Publishing Baltimore, Maryland, 77–85.

[30] Priyanka Meel and Dinesh Kumar Vishwakarma. 2019. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications* (2019).

[31] Robertus Nugroho, Cecile Paris, Surya Nepal, Jian Yang, and Weiliang Zhao. 2020. A survey of recent methods on deriving topics from Twitter: algorithm to evaluation. *Knowledge and Information Systems* (2020), 1–35.

[32] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[33] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

[34] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.

[35] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A hybrid deep model for fake news detection. In *Proc. of the ACM on Conference on Information and Knowledge Management (CIKM 2017)*. 797–806.

[36] Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2019. The Woman Worked as a Babysitter: On Biases in Language Generation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*. 3398–3403.

[37] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2020. Towards Controllable Biases in Language Generation. arXiv:2005.00268 [cs.CL]

[38] Vered Shwartz, Gabriel Stanovsky, and Ido Dagan. 2017. Acquiring predicate paraphrases from news tweets. In *Prof. of the 6th Joint Conference on Lexical and Computational Semantics (SEM 2017)*. 155–160.

[39] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203* (2019).

[40] Tetsuro Takahashi and Nobuyuki Igata. 2012. Rumor detection on twitter. In *Proc. of the 13th IEEE International Symposium on Advanced Intelligence Systems (ISIS 2012)*. 452–457.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of the 30th Advances in Neural Information Processing Systems (NIPS 2017)*. 5998–6008.

[42] Alex Wang and Kyunghyun Cho. 2019. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. In *Proc. of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. 30–36.

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. 5753–5763.

[44] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *arXiv preprint arXiv:1905.12616* (2019).

[45] Qiao Zhang, Shuiyuan Zhang, Jian Dong, Jinhua Xiong, and Xueqi Cheng. 2015. Automatic detection of rumor on social network. In *Natural Language Processing and Chinese Computing*. Springer, 113–122.

[46] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proc. of the 9th International Conference on Learning Representations (ICLR 2020)*.

[47] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proc. of the 24th International Conference on World Wide Web (WWW 2015)*. 1395–1405.

[48] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *Proc. of the 41st ACM SIGIR International Conference on Research & Development in Information Retrieval (SIGIR 2018)*. 1097–1100.

[49] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and Resolution of Rumours in Social Media: A Survey. *ACM Comput. Surv.* 51, 2, Article 32 (2018), 36 pages.