Judge the Judges: A Large-Scale Evaluation Study of Neural Language Models for Online Review Generation

Cristina Gârbacea[1], **Samuel Carton**[2], **Shiyan Yan**[2], **Qiaozhu Mei**[1,2]
[1]Department of EECS, University of Michigan, Ann Arbor, MI, USA
[2]School of Information, University of Michigan, Ann Arbor, MI, USA
{garbacea, scarton, shiyansi, qmei}@umich.edu

Abstract

We conduct a large-scale, systematic study to evaluate the existing evaluation methods for natural language generation in the context of generating online product reviews. We compare human-based evaluators with a variety of automated evaluation procedures, including discriminative evaluators that measure how well machine-generated text can be distinguished from human-written text, as well as word overlap metrics that assess how similar the generated text compares to human-written references. We determine to what extent these different evaluators agree on the ranking of a dozen of state-of-the-art generators for online product reviews. We find that human evaluators do not correlate well with discriminative evaluators, leaving a bigger question of whether adversarial accuracy is the correct objective for natural language generation. In general, distinguishing machine-generated text is challenging even for human evaluators, and human decisions correlate better with lexical overlaps. We find lexical diversity an intriguing metric that is indicative of the assessments of different evaluators. A post-experiment survey of participants provides insights into how to evaluate and improve the quality of natural language generation systems [1].

1 Introduction

Recent developments in neural language models (Mikolov and Zweig, 2012), (Reiter and Belz, 2009), (Mikolov et al., 2011b), (Mikolov et al., 2011a) have inspired the use of neural network based architectures for the task of natural language generation (NLG). Despite fast development of algorithms, there is an urgency to fill the huge gap in evaluating NLG systems. On one hand, a rigorous, efficient, and reproducible evaluation procedure is critical for the development of any machine learning technology and for correct interpretation of the state-of-the-art. On the other hand, evaluating the quality of language generation is inherently difficult due to the special properties of text, such as *subjectivity* and *non-compositionality*. Indeed, *"there is no agreed objective criterion for comparing the goodness of texts"* (Dale and Mellish, 1998), and there lacks a clear model of text quality (Hardcastle and Scott, 2008).

Conventionally, most NLG systems have been evaluated in a rather informal manner. (Reiter and Belz, 2009) divide existing evaluation methods commonly employed in text generation into three categories: *i)* evaluations based on task performance, *ii)* human judgments and ratings, where human subjects are recruited to rate different dimensions of the generated texts, and *iii)* evaluations based on comparison to a reference corpus using automated metrics. *Task based evaluation* considers that the value of a piece of functional text lies in how well it serves the user to fulfill a specific application. It can be expensive, time-consuming, and often dependent on the good will of participants in the study. Besides that, it is hard to toss out the general quality of text generation from the special context (and confounds) of the task, or to generalize the evaluation conclusions across tasks. *Human annotation* is able to assess the quality of text more directly than task based evaluation. However, rigorously evaluating NLG systems with real users can be expensive and time consuming, and it does not scale well (Reiter et al., 2001). Human assessments also need to be consistent and repeatable for a meaningful evaluation (Lopez, 2012). Alternative strategies which are more effective in terms of cost and time are used more frequently.

Automated evaluation compares texts generated by the candidate algorithms to human-written texts.

[1] The experimental setup, data, and annotations are publicly available at: `https://github.com/Crista23/JudgeTheJudges`

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3968–3981,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

Word overlap metrics and more recent automated adversarial evaluators are widely employed in NLG as they are cheap, quick, repeatable, and do not require human subjects when a reference corpus is already available. In addition, they allow developers to make rapid changes to their systems and automatically tune parameters without human intervention. Despite the benefits, however, the use of automated metrics in the field of NLG is controversial (Reiter and Belz, 2009), and their results are often criticized as not meaningful for a number of reasons. First, these automatic evaluations rely on a high-quality corpus of references, which is not often available. Second, comparisons with a reference corpus do not assess the usefulness and the impact of the generated text on the readers as in human-based evaluations. Third, creating human written reference texts specifically for the purpose of evaluation could still be expensive, especially if these reference texts need to be created by skilled domain experts. Finally and most importantly, using automated evaluation metrics is sensible only if they correlate with results of human-based evaluations and if they are accurate predictors of text quality, which is never formally verified at scale.

We present a large-scale, systematic experiment that evaluates the *evaluators* for NLG. We compare three types of evaluators including human evaluators, automated adversarial evaluators trained to distinguish human-written from machine-generated product reviews, and word overlap metrics (such as BLEU and ROUGE) in a particular scenario, generating online product reviews. The preferences of different evaluators on a dozen representative deep-learning based NLG algorithms are compared with human assessments of the quality of the generated reviews. Our findings reveal significant differences among the evaluators and shed light on the potential factors that contribute to these differences. The analysis of a post experiment survey also provides important implications on how to guide the development of new NLG algorithms.

2 Related Work

2.1 Deep Learning Based NLG

Recently, a decent number of deep learning based models have been proposed for text generation. Recurrent Neural Networks (RNNs) and their variants, such as Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) models, Google LM (Jozefowicz et al., 2016), and Scheduled Sampling (SS) (Bengio et al., 2015) are widely used for generating textual data.

Generative Adversarial Networks (Goodfellow et al., 2014), or GANs, train generative models through an adversarial process. Generating text with GANs is challenging due to the discrete nature of text data. SeqGAN (Yu et al., 2017) is one of the earliest GAN-based model for sequence generation, which treats the procedure as a sequential decision making process. RankGAN (Lin et al., 2017) proposes a framework that addresses the quality of a set of generated sequences collectively. Many GAN-based models (Yu et al., 2017), (Lin et al., 2017), (Rajeswar et al., 2017), (Che et al., 2017), (Li et al., 2017), (Zhang et al., 2017) are only capable of generating short texts. LeakGAN (Guo et al., 2018) is proposed for generating longer texts.

Deep learning architectures other than LSTM or GAN have also been proposed for text generation. (Tang et al., 2016) study NLG given particular contexts or situations and proposes two approaches based on the encoder-decoder framework. (Dong et al., 2017) address the same task and employ an additional soft attention mechanism. Pre-training enables better generalization in deep neural networks (Erhan et al., 2010), especially when combined with supervised discriminative fine-tuning to learn universal robust representations (Radford et al., 2018), (Devlin et al., 2018), (Radford et al., 2019). (Guu et al., 2018) use a prototype-then-edit generative language model for sentences.

2.2 Automated Evaluation Metrics

The variety of NLG models are also evaluated with various approaches. Arguably, the most natural way to evaluate the quality of a generator is to involve humans as judges, either through some type of Turing test (Machinery, 1950) to distinguish generated text from human input texts, or to directly compare the texts generated by different generators (Mellish and Dale, 1998). Such approaches are hard to scale and have to be redesigned whenever a new generator is included. Practically, it is critical to find automated metrics to evaluate the quality of a generator independent of human judges or an exhaustive set of competing generators.

Perplexity (Jelinek et al., 1977) is commonly used to evaluate the quality of a language model, which has also been employed to evaluate generators (Yarats and Lewis, 2018), (Ficler and Goldberg, 2017), (Gerz et al., 2018) even though it is commonly criticized for not being a direct measure of the quality of generated text (Fedus et al., 2018). Perplexity is a model dependent metric, and "how likely a sentence is generated by a given model" is not comparable across different models. Therefore we do not include perplexity in this study.

Discriminative Evaluation is an alternative way to evaluate a generator, which measures how likely its generated text can fool a classifier that aims to distinguish the generated text from human-written texts. In a way, this is an automated approximation of the Turing test, where machine judges are used to replace human judges. Discriminative machine judges can be trained either using a data set with explicit labels (Ott et al., 2011), or using a mixture of text written by real humans and those generated by the model being evaluated. The latter is usually referred to as *adversarial evaluation*. (Bowman et al., 2016) proposes one of the earliest studies that uses adversarial error to assess the quality of generated sentences. Notably, maximizing the adversarial error is consistent to the objective of the generator in generative adversarial networks. (Kannan and Vinyals, 2017) propose an adversarial loss to discriminate a dialogue model's output from human output. The discriminator prefers longer output and rarer language instead of the common responses generated. There however lacks evidence that a model that obtains a lower adversarial loss is better according to human evaluations.

Automatic dialogue evaluation is formulated as a learning problem in (Lowe et al., 2017), who train an RNN to predict the scores a human would assign to dialogue responses. RNN predictions correlate with human judgments at the utterance and system level, however each response is evaluated in a very specific context and the system requires substantial human judgments for training. (Li et al., 2017) employ a discriminator (analogous to the human evaluator in the Turing test) both in training and testing and define adversarial success. Other work finds the performance of a discriminative agent (e.g., attention-based bidirectional LSTM binary classifier) is comparable with human judges at distinguishing between real and fake dialogue excerpts (Bruni and Fernández, 2017). However, results show there is limited consensus among humans on what is considered as coherent dialogue passages.

Word Overlap Metrics, such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), are commonly used to measure the similarity between the generated text and human written references. (Liu et al., 2016) find that word overlap metrics present weak or no correlation with human judgments in non-task oriented dialogue systems and thus should be used with caution or in combination with user studies. In contrary, it is reported in (Sharma et al., 2017) that text overlap metrics are indicative of human judgments in task-oriented dialogue settings, when used on datasets which contain multiple ground truth references. (Dai et al., 2017) find text overlap metrics too restrictive as they focus on fidelity of wording instead of fidelity of semantics. (Callison-Burch et al., 2006) consider an increase in BLEU insufficient for an actual improvement in the quality of a system and posit in favor of human evaluation.

BLEU and its variants (e.g., Self-BLEU) are used to evaluate GAN models (Caccia et al., 2018; Zhu et al., 2018). (Shi et al., 2018) compare frameworks for text generation including MLE, SeqGAN, LeakGAN and Inverse Reinforcement Learning using a simulated Turing test. A benchmarking experiment with GAN models is conducted in (Lu et al., 2018); results show LeakGAN presents the highest BLEU scores on the test data. Similarly, BLEU and METEOR present highest correlations with human judgements (Callison-Burch et al., 2008), (Graham and Baldwin, 2014). However, evaluation metrics are not robust across conditions, and no single metric consistently outperforms other metrics across all correlation levels (Przybocki et al., 2009).

Conventional neural language models trained with maximum likelihood can be on par or better than GANs (Caccia et al., 2018), (Semeniuta et al., 2018), (Tevet et al., 2018). However, log-likelihood is often computationally intractable (Theis et al., 2016). Models with good likelihood can produce bad samples, and vice-versa (Goodfellow, 2016). Generative models should be evaluated with regards to the task they are intended for over the full quality-diversity spectrum (Cífka et al., 2018), (Hashimoto et al., 2019), (Montahaei et al., 2019).

While many generators are proposed and evaluated with various metrics, no existing work has systematically evaluated the different evaluators at scale, especially in the context of online review generation. Our work fills in this gap.

3 Experiment Design

We design a large-scale experiment to systematically analyze the procedures and metrics used for evaluating NLG models. To test the different *evaluators*, the experiment carefully chooses a particular application context and a variety of natural language generators in this context. Ideally, a sound automated evaluator should be able to distinguish good generators from suboptimal ones. Its preferences (on ordering the generators) should be consistent to humans in the exact application context.

3.1 Experiment Context and Procedure

We design the experiment in the context of generating online product reviews. There are several reasons why review generation is a desirable task for the experiment: 1) online product reviews are widely available, and it is easy to collect a large number of examples for training/ testing the generators; 2) Internet users are used to reading online reviews, and it is easy to recruit capable human judges to assess the quality of reviews; and 3) comparing to tasks like image caption generation or dialogue systems, review generation has minimal dependency on the conversation context or on non-textual data, which reduces possible confounds.

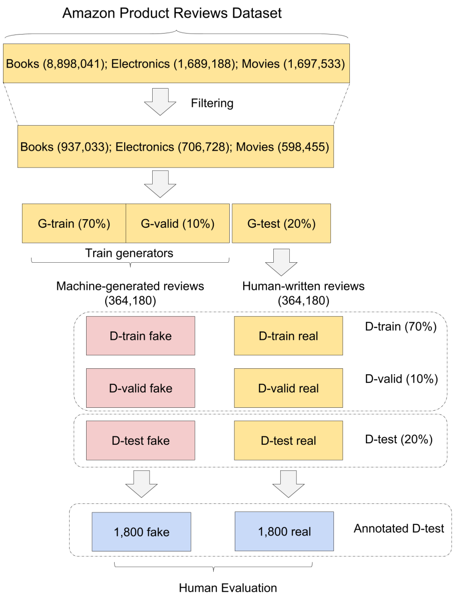

Figure 1: Overview of the Experiment Procedure.

The general experiment procedure is presented in Figure 1. We start from the publicly available Amazon Product Reviews dataset [2] and select three most popular domains: *books*, *electronics*, and *movies*. After filtering rare products, inactive users, and overly long reviews, the dataset is randomly split into three parts, to train, to validate, and to test the candidate review generators (denoted as *G-train*, *G-valid*, and *G-test*). Every generative model is trained and validated using the same datasets, and then charged to generate a number of product reviews (details are included in the next section). These generated reviews, mixed with the real reviews in *G-test*, are randomly split into three new subsets for training, validating, and testing candidate (discriminative) evaluators, denoted as *D-train*, *D-valid*, and *D-test*. Finally, a random sample of reviews from *D-test* are sent for human evaluation.

3.2 Review Generators

Although our goal is to evaluate the evaluators, it is critical to include a wide range of text generators with various degrees of quality. A good evaluator should be able to distinguish the high-quality generators from the low-quality ones. We select a diverse set of generative models from recent literature. The goal of this study is *not* to name the best generative model, and it is unfeasible to include all existing models. Our criteria are: (1) the models are published before 2018, when our experiment is conducted; (2) the models represent different learning strategies and quality levels; (3) the models have publicly available implementations, for reproducibility purposes. In Table 1 we list the candidate generators. It is not an exhaustive list of what are currently available. For implementation details of these models please see Appendix A.1.

Table 1: Candidate models for review generation.

Generative Model	Adversarial Framework
Word LSTM temp 1.0 (Hochreiter and Schmidhuber, 1997)	No
Word LSTM temp 0.7 (Hochreiter and Schmidhuber, 1997)	No
Word LSTM temp 0.5 (Hochreiter and Schmidhuber, 1997)	No
Scheduled Sampling (Bengio et al., 2015)	No
Google LM (Jozefowicz et al., 2016)	No
Attention Attribute to Sequence* (Dong et al., 2017)	No
Contexts to Sequences* (Tang et al., 2016)	No
Gated Contexts to Sequences* (Tang et al., 2016)	No
MLE SeqGAN (Yu et al., 2017)	Yes
SeqGAN (Yu et al., 2017)	Yes
RankGAN (Lin et al., 2017)	Yes
LeakGAN (Guo et al., 2018)	Yes

* indicates that review generation using these models are conditional on context information such as product ids; other models are context independent.

Every generator (except Google LM) is trained and validated on *G-train* and *G-valid* datasets, and used to generate the same number of machine-generated (a.k.a., fake) reviews (see Table 2). We follow the best practice in literature to train these models, although it is possible that the performance of models might not be optimal due to various con-

[2] http://jmcauley.ucsd.edu/data/amazon/

straints. This will not affect the validity of the experiment as our goal is to evaluate the **evaluators** instead of the individual generators. Google LM was not trained on reviews, but it provides a sanity check for the experiment - a reasonable evaluator should not rank it higher than those trained for generating reviews.

Table 2: Number of generated reviews by each model.

Generative Model	**Total**	**D-Train**	**D-Valid**	**D-Test**
∀ model in Table 1 except Google LM	32,500	22,750	3,250	6,500
Google LM	6,680	4,676	668	1,336

3.3 Evaluators

We include a comprehensive set of evaluators for the quality of the aforementioned generators: *i)* human evaluators, *ii)* discriminative evaluators, and *iii)* text overlap evaluators. The evaluators are the main subjects of the experiment.

3.3.1 Human evaluators

We conduct a careful power analysis (Christensen, 2007), which suggests that at least 111 examples per generative model should be human annotated to infer that the machine-generated reviews are comparable in quality to human-written reviews, at a minimal statistically significance level of 0.05. Per this calculation, we sample 150 examples for each of the 12 generators for human evaluation. This totals 1,800 machine-generated reviews, to which we add 1,800 human-written reviews, or a total of 3,600 product reviews sent for human annotation. We markup out-of-vocabulary words in *both* human-written and machine-generated reviews to control for confounds of using certain rare words. There is no significant difference in proportion of the markup token between the two classes (2.5%-real vs. 2.2%-fake). We recruit 900 human annotators through the Amazon Mechanical Turk (AMT) platform. Each annotator is presented 20 reviews, a mixture of 10 real (i.e., human written) and 10 fake (i.e., machine generated), and they are charged to label each review as real or fake based on their own judgment. Clear instructions are presented to the workers that markup tokens are present in both classes and cannot be used to decide whether a review is real or fake. Each page is annotated by 5 distinct human evaluators. The 5 judgments on every review are used to assemble two distinct **human evaluators**: *H1* - **individual votes**, treating all human annotations independently, and *H2* - **majority votes** of the 5 human judgments. For every *annotated* review, the human evaluator ($H1$ or $H2$) makes a call which can be either right or wrong with regard to the ground truth. A generator is high quality if the human evaluator achieves low accuracy identifying the reviews as fake.

3.3.2 Discriminative evaluators

The inclusion of multiple generators provides the opportunity of creating **meta-adversarial evaluators**, trained using a *pool* of generated reviews by *many* generators, mixed with a larger number of "real" reviews (*D-train* and *D-valid* datasets). Such a "pooling" strategy is similar to the standard practice used by the TREC conferences to evaluate different information retrieval systems (Harman and Voorhees, 2006). Comparing to individual adversarial evaluators, a meta-evaluator is supposed to be more robust and fair, and it can be applied to evaluate new generators without being retrained. In our experiment, we find that the meta-adversarial evaluators rank the generators in similar orders to the best individual adversarial evaluators.

We employ a total of 7 meta-adversarial evaluators: 3 deep, among which one using LSTM (Hochreiter and Schmidhuber, 1997), one using Convolutional Neural Network (CNN) (LeCun et al., 1998), and one using a combination of LSTM and CNN architectures; 4 shallow, based on Naive Bayes (NB) (Rish, 2001), Random Forest (RF) (Liaw et al., 2002), Support Vector Machines (SVM) (Cortes and Vapnik, 1995), and XGBoost (Chen and Guestrin, 2016), with unigrams, bigrams, and trigrams as features and on balanced training sets. We find the best hyper-parameters using random search and prevent the models from overfitting by using early stopping. For every review in *D-test* (either annotated or not), a meta-adversarial evaluator makes a judgment call. A generator is considered high quality if the meta-adversarial evaluator makes more mistakes on reviews it generated.

3.3.3 Word-overlap evaluators

We include a set of 4 text-overlap metrics used for NLG evaluation: BLEU and METEOR (specific to machine translation), ROUGE (used in text summarization), and CIDEr (Vedantam et al., 2015) (used in image description evaluation). These metrics rely on matching n-grams in the target text (i.e., generated reviews) to the "references" (i.e., human-written reviews). The higher the overlap (similarity), the higher the quality of generated text. For every generated review in *D-test Fake*, we assemble the set of references by retrieving the top-10 most similar human-written reviews in *D-test Real* using a simple vector space model. We compute 600-dimensional vector representation of reviews using Sent2Vec (Pagliardini et al., 2018), pretrained on English Wikipedia, and retrieve the top-k nearest neighbors for each review based on cosine similarity of the embedding vectors. The rationale of using nearest neighbors of each generated review as references is that to appear "real", a generated review just need to be similar to *some* real reviews instead of *all*. A generator is considered high quality if its generated reviews obtain a high average score by a text overlap evaluator. In total, we analyze and compare 13 candidate evaluators (2 human evaluators, 7 discriminative evaluators, and 4 text-overlap metrics), based on the *D-test* dataset.

4 Results

First, we are interested in the accuracy of individual evaluators - how well they can distinguish "fake" (machine-generated) reviews from "real" (human-written) reviews. Second, we are interested in how an evaluator assesses the quality of the 12 generators instead of individual reviews. The absolute scores an evaluator gives to the generators are not as informative as how it ranks them: a good evaluator should be able to rank good generators above bad generators. Last but not least, we are interested in how the rankings by different evaluators correlate with each other. Intuitively, an automated evaluator that ranks the generators in similar orders as the human evaluators is more reasonable and can potentially be used as the surrogate of humans.

4.1 Results of Individual Evaluators

4.1.1 Human evaluators

Every review is annotated by 5 human judges as either "fake" or "real." The inter-annotator agreement (Fleiss-Kappa score (Fleiss et al., 2013)) is $k = 0.2748$. This suggests that *distinguishing machine-generated reviews from real in general is a hard task even for humans*; there is limited consensus on what counts as a realistic review. The low agreement also implies that any automated evaluator that mimics human judges is not necessarily the most "accurate."

In Figure 2 we present the accuracy of two human evaluators on individual annotated reviews, based on either all 5 annotations or their majority votes for each review. Comparing to the ground-truth (of whether a review is machine-generated or collected from Amazon), individual human decisions are 66.61% accurate, while their majority votes can reach 72.63%. Neither of them is close to perfect. *We observe that human evaluators generally do better at correctly labelling human-written reviews as real (true positive rate of 78.96% for* $H1$ *and 88.31% for* $H2$*), and they are confused by machine-generated reviews in close to half of the cases (true negative rate of 54.26% for* $H1$ *and 56.95% for* $H2$*).* This trend reassures previous observations (Tang et al., 2016).

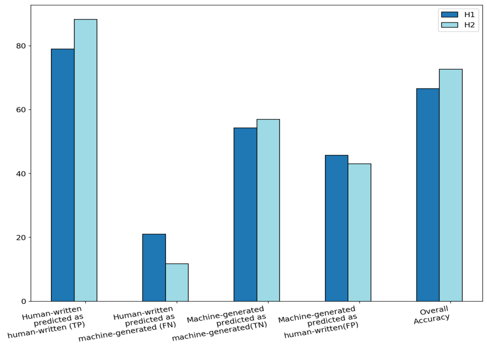

Figure 2: Accuracy of human evaluators on individual reviews: *H1* - individual votes; *H2* - majority votes.

We then look at how the human evaluators rank the 12 generators, according to the accuracy of human evaluators on all (fake) reviews generated by each of the generators. The lower the accuracy, the more likely the human evaluator is confused by the generated reviews, and thus the better the generator. We observe a substantial variance in the accuracy of both human evaluators on different generators, which suggests that human evaluators are able to distinguish between generators. The generator ranked the highest by both human evaluators is *Gated Contexts to Sequences*. Google LM is ranked on the lower side, which makes sense as the model is not trained to generate reviews. Interestingly, humans tend not to be fooled by reviews generated by the GAN-based models (MLE SeqGAN, SeqGAN, RankGAN and LeakGAN), even though their objective is to confuse fake from real. GAN-generated reviews tend to be easily distinguishable from the real reviews by human judges.

4.1.2 Discriminative evaluators

We then analyze the 7 meta-adversarial evaluators. Different from human evaluators that are applied to the 3,600 annotated reviews, the discriminative evaluators are applied to *all* reviews in *D-test*.

Meta-adversarial Evaluators. On individual reviews, the three deep learning based and the one SVM based evaluators achieve higher accuracy than the two human evaluators, indicating that adversarial evaluators can distinguish a single machine-generated review from human-written better than humans (Figure 3 and Table 4 in Appendix A.3.2). Their true positive rates and true negative rates are more balanced than human evaluators. Meta-discriminators commonly rank GAN-based generators the highest. This makes sense as the objective of GAN is consistent to the (reversed) evaluator accuracy. Interestingly, by simply setting the temperature parameter of Word LSTM to 1.0, it achieves comparable performance to the GANs.

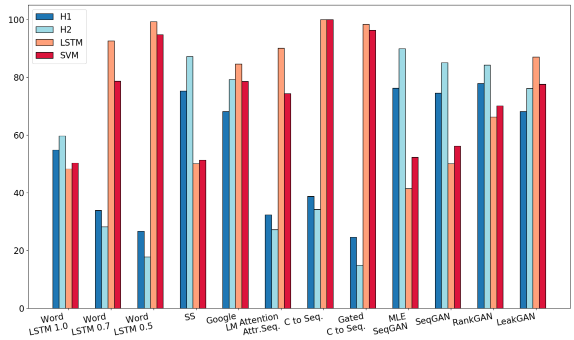

Figure 3: Accuracy of human (H1, H2) and meta-adversarial evaluators (LSTM, SVM) on reviews generated by individual generators. **The lower the accuracy, the better the generator.**

4.1.3 Word-Overlap Evaluators

The generators are ranked based on the average scores of their generated reviews. In Figure 4 we present the average scores of the 12 generators by each evaluator. Different word-overlap evaluators also tend to rank the generators in similar orders. Interestingly, the top-ranked generator according to three evaluators is *Contexts to Sequences*, while CIDEr scores highest the *Gated Contexts to Sequences* model. GAN-based generators are generally ranked low; please also see Appendix A.3.3.

4.2 Comparing Evaluators

To what degree do the evaluators agree on the ranking of generators? We are more interested in how the automated evaluators compare to the human evaluators, and whether there is any suitable automated surrogate for human judges at all. To do this, we compute the correlations between $H1$, $H2$ and each discriminative evaluator and correlations between $H1$, $H2$ and the text-overlap evaluators, based on either their decisions on individual reviews, their scores of the generators (by Pearson's coefficient (Fieller et al., 1957)), and their rankings of the generators (by Spearman's ρ (Spearman, 1904) and Kendall's τ (Daniel et al., 1978)). Patterns of the three correlation metrics are similar; please see Figure 5 and Table 5 in Appendix A.3.4.

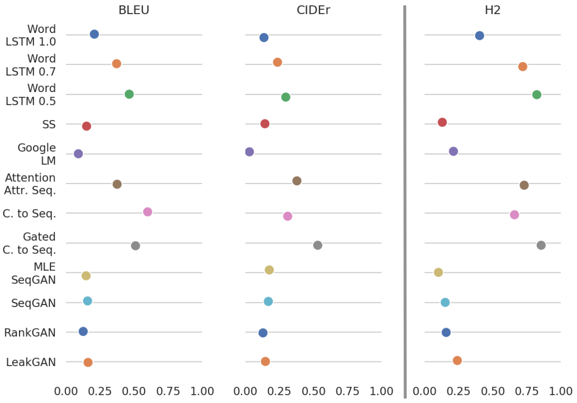

Figure 4: Text-Overlap Evaluators (BLEU and CIDEr) scores for individual generators. **The higher the better.** The rankings are overall similar, as GAN-based generators are ranked low.

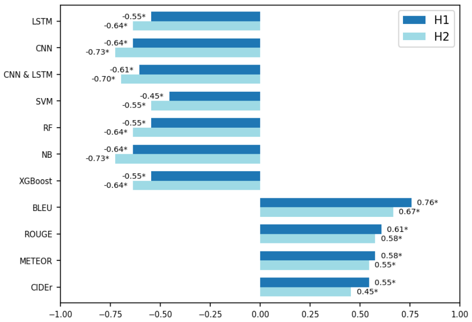

Figure 5: Kendall τ-b between human and automated evaluators. Human's ranking is positively correlated to text-overlap evaluators and negatively correlated to adversarial evaluators (* is $p \leq 0.05$).

Surprisingly, none of the discriminative evaluators have a positive correlation with the human evaluators. That says, *generators that fool machine judges easily are less likely to confuse human judges, and vice versa. Word-overlap evaluators tend to have a positive correlation with the human evaluators in ranking the generators.* Among them, BLEU appears to be closer to humans. This pattern is consistent in all three types of correlations. These two observations are intriguing, which indicates that when identifying fake reviews, humans might focus more on word usage rather than trying to construct a “decision boundary” mentally.

In summary, we find that 1) human evaluators cannot distinguish machine-generated reviews from real reviews perfectly, with significant bias between the two classes; 2) meta-adversarial evaluators can better distinguish individual fake reviews, but their rankings at the generator level tend to be negatively correlated with human evaluators; and 3) text-overlap evaluators are highly correlated with human evaluators in ranking generators.

5 Discussion

We carried a systematic experiment that evaluates the evaluators for NLG. Results have intriguing implications to both the evaluation and the construction of natural language generators. We conduct in-depth analysis to discover possible explanations.

5.1 Granularity of Judgments

We charged the Turkers to label individual reviews as either fake or real instead of evaluating each generator as a whole. Comparing to an adversarial discriminator, a human judge has not seen many “training” examples of *fake* reviews or generators. That explains why the meta-adversarial evaluators are better at identifying fake reviews. In this context, humans are likely to judge whether a review is real based on how “similar” it appears to the true reviews they are used to seeing online.

This finding provides interesting implications to the selection of evaluation methods for different tasks. In tasks that are set up to judge individual pieces of generated text (e.g., reviews, translations, summaries, captions, fake news) where there exists human-written ground-truth, it is better to use word-overlap metrics instead of adversarial evaluators. When judgments are made on the agent/system level (e.g., whether a Twitter account is a bot), signals like how similar the agent outputs are or how much the agent memorizes the training examples may become more useful than word usage, and a discriminative evaluator may be more effective than word-overlap metrics. Our finding also implies that adversarial accuracy might not be the optimal objective for NLG if the goal is to generate documents that humans consider as real. Indeed, a fake review that fools humans does not necessarily need to fool a machine that has seen everything. In Appendix B.2 we provide more details.

5.2 Imperfect Ground Truth

One important thing to note is that all discriminative evaluators are trained using natural labels (i.e., treating all examples from the Amazon review dataset as positive and examples generated by the candidate models as negative) instead of human-annotated labels. Some reviews posted on Amazon may have been generated by bots, and if that is the case, treating them as human-written examples may bias the discriminators. To verify this, we apply the already trained meta-discriminators to the human-annotated subset (3,600 reviews) instead of the full *D-test* set, and we use the majority vote of human judges (whether a review is fake or real) to surrogate the natural “ground-truth” labels (whether a review is generated or sampled from Amazon).

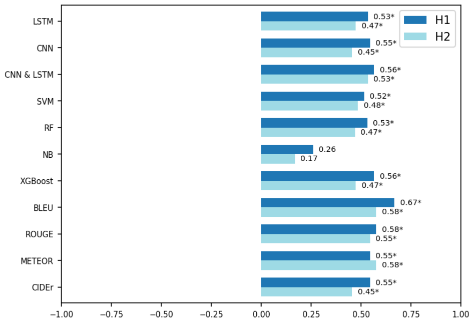

Figure 6: Kendall τ-b correlation coefficients between human evaluators and automated evaluators, tested on the **annotated subset of *D-test*** with *majority votes* as ground-truth (* denotes $p \leq 0.05$).

When the meta-adversarial evaluators are tested using human majority-votes as ground-truth, the scores and rankings of these discriminative evaluators are more inline with the human evaluators, although still not as highly correlated as BLEU; please see Figure 6. Indeed, discriminative evaluators suffer the most from low-quality labels, as they were directly trained using the imperfect ground-truth. Word-overlap evaluators are more robust, as they only take the most relevant parts of the test set as references (more likely to be high quality). Our results also suggest that when adversarial training is used, selection of training examples must be done with caution. If the “ground-truth” is hijacked by low quality or “fake” examples, models trained by GAN may be significantly biased. This finding is related to the recent literature of the robustness and security of machine learning models (Papernot et al., 2017). Appendix B.3 contains further details.

5.3 Role of Diversity

We assess the role diversity plays in rankings the generators. Diversity of a generator is measured by either the lexical diversity (Bache et al., 2013) or Self-BLEU (Zhu et al., 2018) of the samples produced by the generator. Results obtained (see Figure 7) indicate generators that produce the least diverse samples are easily distinguished by the meta-discriminators, while they confuse humans the most. This confirms that adversarial discriminators capture limitations of generative models in lack of diversity (Kannan and Vinyals, 2017).

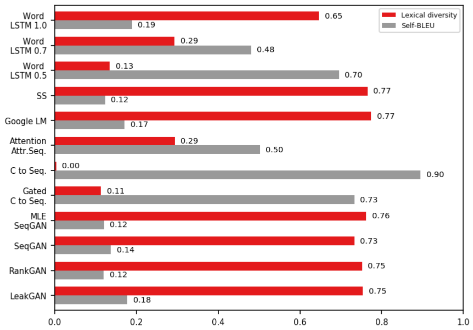

Figure 7: Self-BLEU scores (the lower the more diverse) and lexical diversity scores (the higher the more diverse) are highly correlated in ranking the generators.

Similarly, we measure to what extent the generators are memorizing the training set *G-train* as the average BLEU scores of generated reviews using their nearest neighbors in *G-train* as references. We observe the generators do not memorize the training set, and GAN models generate reviews that have fewer overlap with *G-train*; this finding is in line with recent theoretical studies on memorization in GANs (Nagarajan et al.).

The effects of diversity indicate that when humans are distinguishing individual reviews as real or fake, whether or not a fake review is sufficiently different from the other generated reviews is not a major factor for their decision. Instead, they tend to focus on whether the review looks similar to the reviews they have seen in reality. A discriminative evaluator is more powerful in making decisions at a system level (e.g., a dialog system or a bot account), where diversity becomes a major factor. In Appendix B.4 we provide more details.

5.4 User Study

Finally, we are interested in the reasons why human annotators label certain reviews as fake (machine-written). After annotating a batch of reviews, workers are asked to explain their decisions by filling in an optional free-text comment. This enables us to have a better understanding of what differentiates machine-generated from human-written reviews from human's perspective. Analyzing their comments, we identify the main reasons why human evaluators annotate a review as machine-written. These are mainly related to the presence of grammatical errors in the review text, wrong wording or inappropriate choice of expressions, redundant use of specific phrases or contradictory arguments in the review. Interestingly, human evaluators' innate biases are also reflected in their decisions: they are likely to categorize a review as fake if it is too formal, lacks emotion and personal pronouns, or is too vague and generic. Please see Appendix B.1.

5.5 Summary

In summary, our findings represent a preliminary foundation for proposing more solid and robust evaluation metrics and objectives of natural language generation. The low inter-rater agreement we observe suggests that judging *individual* pieces of text as machine- or human-generated is a difficult task even for humans. In this context, discriminative evaluators are not as correlated with human judges as word-overlap evaluators. That implies that adversarial accuracy might not be the optimal objective for generating individual documents when realism is the main concern. In contrast, GAN based models may more easily pass a Turing test on a *system* level, or in a conversational context. When the judges have seen enough examples from the same generator, the next example had better be somewhat different.

Our results also suggest that when adversarial evaluation is used, the training examples must be carefully selected to avoid false-positives. We also find that when humans are distinguishing fake reviews from real ones, they tend to focus more on the usage of words, expressions, emotions, and other details. This may affect the design of objectives for the next generation of NLG models.

Acknowledgement

We thank Wei Ai for his help on the power analysis, and Yue Wang and Teng Ye for helpful discussions. This work is in part supported by the National Science Foundation under grant numbers 1633370 and 1620319 and by the National Library of Medicine under grant number 2R01LM010681-05.

References

Kevin Bache, David Newman, and Padhraic Smyth. 2013. Text-based measures of document diversity. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 23–31. ACM.

Philip Bachman and Doina Precup. 2015. Data generation as sequential decision making. In *Advances in Neural Information Processing Systems*, pages 3249–3257.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Elia Bruni and Raquel Fernández. 2017. Adversarial evaluation for open-domain dialogue generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 284–288.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2018. Language gans falling short. *arXiv preprint arXiv:1811.02549*.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the third workshop on statistical machine translation*, pages 70–106. Association for Computational Linguistics.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, and Yoshua Bengio. 2017. Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv:1702.07983*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

Erik Christensen. 2007. Methodology of superiority vs. equivalence trials and non-inferiority trials. *Journal of hepatology*, 46(5):947–954.

Ondřej Cífka, Aliaksei Severyn, Enrique Alfonseca, and Katja Filippova. 2018. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *arXiv preprint arXiv:1804.07972*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. 2017. Towards diverse and natural image descriptions via a conditional gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2970–2979.

Robert Dale and Chris Mellish. 1998. Towards evaluation in natural language generation. In *In Proceedings of First International Conference on Language Resources and Evaluation*.

Wayne W Daniel et al. 1978. *Applied nonparametric statistics*. Houghton Mifflin.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 623–632.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better text generation via filling in the _. *arXiv preprint arXiv:1801.07736*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *EMNLP 2017*, page 94.

Edgar C Fieller, Herman O Hartley, and Egon S Pearson. 1957. Tests for rank correlation coefficients. i. *Biometrika*, 44(3/4):470–481.

Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. 2013. *Statistical methods for rates and proportions*. John Wiley & Sons.

Daniela Gerz, Ivan Vulić, Edoardo Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. *Transactions of the Association of Computational Linguistics*, 6:451–465.

Ian Goodfellow. 2016. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Yvette Graham and Timothy Baldwin. 2014. Testing for significance of increased correlation with human judgment. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 172–176.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450.

David Hardcastle and Donia Scott. 2008. Can we evaluate the quality of generated text? In *LREC*. Citeseer.

Donna K Harman and Ellen M Voorhees. 2006. Trec: An overview. *Annual review of information science and technology*, 40(1):113–155.

Tatsunori B Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. *arXiv preprint arXiv:1904.02792*.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ferenc Huszár. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity?a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Anjuli Kannan and Oriol Vinyals. 2017. Adversarial evaluation of dialogue models. *arXiv preprint arXiv:1701.08198*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Andy Liaw, Matthew Wiener, et al. 2002. Classification and regression by randomforest. *R news*, 2(3):18–22.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Adam Lopez. 2012. Putting human assessments of machine translation systems in order. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 1–9. Association for Computational Linguistics.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.

Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Neural text generation: Past, present and beyond. *arXiv preprint arXiv:1803.07133*.

Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep photo style transfer. *CoRR, abs/1703.07511*.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. 2016. Semantic segmentation using adversarial networks. *arXiv preprint arXiv:1611.08408*.

Computing Machinery. 1950. Computing machinery and intelligence-am turing. *Mind*, 59(236):433.

Chris Mellish and Robert Dale. 1998. Evaluation in the context of natural language generation. *Computer Speech & Language*, 12(4):349–373.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011a. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. 2011b. Rnnlm-recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. *SLT*, 12:234–239.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. *arXiv preprint arXiv:1904.03971*.

Vaishnavh Nagarajan, Colin Raffel, and Ian J. Goodfellow. Theoretical Insights into Memorization in GANs.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of NAACL-HLT*, pages 528–540.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Mark A. Przybocki, Kay Peterson, Sebastien Bronsart, and Gregory A. Sanders. 2009. The NIST 2008 metrics for machine translation challenge - overview, methodology, metrics, and results. *Machine Translation*, 23(2-3):71–103.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. 2017. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Ehud Reiter, Roma Robertson, A Scott Lennox, and Liesl Osman. 2001. Using a randomised controlled clinical trial to evaluate an nlg system. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 442–449. Association for Computational Linguistics.

Irina Rish. 2001. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242.

Jürgen Schmidhuber and Sepp Hochreiter. 1997. Long short-term memory. *Neural Comput*, 9(8):1735–1780.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. 2018. On accurate evaluation of gans for language generation. *arXiv preprint arXiv:1806.04936*.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward diverse text generation with inverse reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4361–4367. AAAI Press.

Nasim Souly, Concetto Spampinato, and Mubarak Shah. 2017. Semi and weakly supervised semantic segmentation using generative adversarial network. *arXiv preprint arXiv:1703.09695*.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101.

Akash Srivastava, Lazar Valkoz, Chris Russell, Michael U Gutmann, and Charles Sutton. 2017. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3310–3320.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. *arXiv preprint arXiv:1611.09900*.

Guy Tevet, Gavriel Habib, Vered Shwartz, and Jonathan Berant. 2018. Evaluating text gans as language models. *arXiv preprint arXiv:1810.12686*.

L Theis, A van den Oord, and M Bethge. 2016. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)*, pages 1–10.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057.

Denis Yarats and Mike Lewis. 2018. Hierarchical text generation and planning for strategic dialogue. In *International Conference on Machine Learning*, pages 5587–5595.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4006–4015. JMLR. org.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*.

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 *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100. ACM.