Automatic Detection of Generated Text is Easiest when Humans are Fooled

Daphne Ippolito†‡
daphnei@seas.upenn.edu

Daniel Duckworth‡*
duckworthd@google.com

Chris Callison-Burch†‡
ccb@seas.upenn.edu

Douglas Eck‡
deck@google.com

Abstract
With the advent of transformer-based generative models with a billion parameters or more, it is now possible to automatically generate vast amounts of human-sounding text. One would like both humans and automatic discriminators to be capable of detecting generated text, but humans and machines rely on different cues to make their decisions. Existing decoding methods have primarily optimized for fooling humans. Here, we perform careful benchmarking and analysis of three popular sampling-based decoding strategies: top-k, nucleus sampling, and untruncated random sampling. Though both human and automatic detector performance improve with longer excerpt length, even multi-sentence excerpts can fool expert human raters over 30% of the time. Top-k sampling is rated as most human-like by humans, but it is by far the easiest automatic detection task because of differences in its unigram distribution. Our findings reveal the importance of using both human and automatic detectors to assess the humanness of text generation systems.

1 Introduction
State-of-the-art generative models for text are now capable of producing multi-paragraph excerpts that are on the surface virtually indistinguishable from human-written content (Zellers et al., 2019; Radford et al., 2019). Often, only subtle logical fallacies or idiosyncrasies of language give away a text excerpt as machine-generated—errors that require a close reading and/or domain knowledge to detect. As this technology matures, authors, well-meaning or otherwise, will increasingly employ it to augment and accelerate their own writing.

Automatic systems trained to detect generated text are useful to combat bad actors and are a promising direction in adversarial training (Lin et al., 2017; Li et al., 2017) and in automatic evaluation of generative model quality (Novikova et al., 2017; Kannan and Vinyals, 2017). Use of human raters to evaluate generative model quality is widespread (Amidei et al., 2019). However, there has been little inquiry into the properties that cause humans to give generated text high human-like ratings compared to those that cause automatic systems to rate it highly.

Language models output a probability distribution over the next word in a sequence given the previous words, and the field has largely moved toward probabilistic sampling strategies to decode text from these distributions. However, when many low-likelihood words cumulatively contain quite a bit of probability mass, choosing one of these words can lead to odd or contradictory phrases and semantic errors. Humans are quick to notice these types of errors.

Top-k random sampling, where low-likelihood words are restricted from being generated, is an attractive way to get around this issue. A language model that is only permitted to produce high-likelihood words is less likely to make a poor choice and create the type of mistakes that are easy for humans to detect. Since humans are not proficient at identifying when a model subtly favors some utterances more often than a human author would, they don’t notice the over-representation of high-likelihood words in the generated text. In contrast, automatic systems excel at identifying statistical anomalies and struggle to build deeper semantic understanding. Top-k in particular creates text that is easy for machines to detect but very hard for humans. Thus, we observe the general trend: as the number of unlikely words available to be chosen is increased, humans get better at detecting fakes while automatic systems get worse.
In this work, we carefully examine three sampling-based decoding strategies and the ability of humans and automatic systems to detect them. We discuss the conditions under which discriminators trained on one sampling strategy transfer to detecting text from other strategies. We argue that it is necessary to study both human and automatic detection to evaluate the efficacy of new language models and decoding strategies.

In summary, our contributions are:
- A comprehensive study of generated text detection systems’ sensitivity to model structure, decoding strategy, and excerpt length.
- An analysis of human raters’ ability to identify machine-generated content, and how human raters differ from automatic detectors.

2 Related Work

Language Models With a sufficiently large training set, neural language models based on the Transformer architecture (Vaswani et al., 2017) are able to generate convincing, human-like excerpts up to several paragraphs in length. GPT-2 (Radford et al., 2019), Grover (Zellers et al., 2019), and Transformer-DMCA (Liu et al., 2018) are a few examples of large, publicly available models capable of doing so. Grover, in particular, has been shown to generate fake news that is more trustworthy than human-written fake news according to human raters.

Generation of Fake Content Fake news, whether human- or machine-generated, has entered the sphere of public concern (Cooke, 2018). It propagates quickly (Vosoughi et al., 2018), sets political agendas (Vargo et al., 2018), and influences elections (Allcott and Gentzkow, 2017). It is also difficult to identify; in the UK, a mere 2% of children and young people have the skills necessary to identify it (National Literacy Trust, 2018).

Aside from news, crowdurfing of fake reviews on websites such as Amazon and Yelp significantly undermines user trust (Wang et al., 2012; Song et al., 2015). According to Luca and Zervas (2016), approximately 16% of Yelp restaurant reviews were fraudulent as of 2012. Recently, Adelani et al. (2019) have shown that automatically generated reviews are perceived to be as fluent as human-written ones. The problem extends to the political sphere; millions of fake comments in favor of repealing Net Neutrality rules were submitted to the FCC during its public comment period (Fung, 2017; Singer-Vine and Collier, 2019).

A significant barrier to generating fake content is cost. Fake news requires writers; fake reviews, fraudulent reviewers. The Internet Research Agency, a Russian-backed troll factory, had a monthly budget of more than $1 million to interfere in the 2016 US Presidential election (Weiss, 2018). With recent developments in neural language modeling, large quantities of synthetic content can be generated automatically at nearly no cost. Identifying such content is now more important than ever.

Detecting Generated Content The rise of machine-generated content has led to the development of models to identify it. Grover was developed to not only generate convincing excerpts but to also identify them using a fine-tuned version of the generative model itself (Zellers et al., 2019). GLTR aims to make machine-generated text detectable by computing histograms over per-token log likelihoods, expecting attackers to use sampling methods that favor high-likelihood tokens (Gehrmann et al., 2019). Bakhtin et al. (2019) frame human-text detection as a ranking task and evaluate their models’ cross-domain and cross-model generalization, finding significant loss in quality when training on one domain and evaluating on another. Finally, Schuster et al. (2019) argue that language distributional features implicitly or explicitly used by these detectors are insufficient; instead, one should look to explicit fact-verification models.

Natural Language Understanding Detection of machine-generated text requires a semantic understanding of the text. Contradictions, falsehoods, and topic drift can all indicate that an excerpt was machine-generated. Encoder-only Transformer models such as BERT (Devlin et al., 2018) have been shown to do very well at tasks requiring this understanding. While we fine-tune BERT for the task of classifying whether text was machine-generated, others have used the contextual word embeddings from a pre-trained BERT model without fine-tuning to compute a quality score for generated text (Zhang et al., 2019). It is worth noting that recent work has raised questions as to whether BERT truly builds a semantic understanding or whether it merely takes advantage of spurious statistical differences between the text of different classes (Niven and Kao, 2019).
3 Dataset Methodology

We frame the detection problem as a binary classification task: given an excerpt of text, label it as either human or machine. In particular, we are interested in how excerpt length and decoding strategy impact detectability. To this end, generated text samples are drawn from GPT-2, a state-of-the-art Transformer-based generative language model that was trained on text from popular web pages (Radford et al., 2019). While we use the GPT-2 LARGE model with 774M parameters, in experiments with smaller language models, we found that similar trends to those reported here hold.

Given an autoregressive language model that defines a probability distribution over the next token given the previous tokens in a sequence, the choice of decoding strategy used to generate text from these distributions is critical. Perhaps the most straightforward method is to at each step randomly choose a token with probability proportional to its likelihood. A challenge with the random sampling approach is that these probability distributions often contain a long tail of vocabulary items that are individually low-probability but cumulatively comprise a substantial amount of probability mass. Holtzman et al. (2019) observe that choosing tokens from this tail often leads to incoherent generations.

Top-$k$ sampling, nucleus sampling, and (in the extreme) beam search have all been proposed to heuristically promote samples with higher per-token likelihoods. Top-$k$ and nucleus sampling both do so by setting the likelihood of tokens in the tail of the distribution to zero. Top-$k$ restricts the distribution to all but the $k$ most likely tokens, where $k$ is a constant (Fan et al., 2018). Nucleus sampling, or top-$p$, truncates the distribution at each decoding step to the $k$-most-likely next tokens such that the cumulative likelihood of these tokens is no greater than a constant $p$ (Holtzman et al., 2019).

We consider three different sampling strategy settings:

- Sample from the untruncated distribution
- Top-$k$, choosing $k=40$ (Radford et al., 2019).
- Nucleus sampling (aka top-$p$), choosing $p=0.96$ (Zellers et al., 2019).

For each sampling method, we construct a training dataset by pairing 250,000 generated samples with 250,000 excerpts of web text that come from the same distribution as GPT-2’s training data\(^1\). 5,000 additional paired samples are kept aside for validation and test datasets. We filter out excerpts with fewer than 192 WordPiece tokens (Wu et al., 2016) (excerpts might be quite short if the model produces an end-of-text token early on). See Appendix 1 for final dataset sizes.

A crucial question when generating text with a language model is whether or not to provide a priming sequence which the language model should continue. Unconditioned samples, where no priming text is provided, in conjunction with top-$k$ sampling, lead to pathological behavior for discriminators as the first token of the generated text will always be one of $k$ possible options. On the other hand, if long sequences of human text are used as priming, the space of possible generated sequences is larger, but the detection problem shifts from one of “how human-like is the generated text?” to “how well does the generated text follow the priming sequence?”.

Since in this study we are interested in the former simpler question, we create two datasets, one with no priming, and one with the minimum amount of priming possible: a single token of web text. This means that for every excerpt of web text in the training set, there is an excerpt of machine text that starts with the same token. We find that even with a single word of priming, detection for certain sampling strategies is strongly impacted.

To study the question of excerpt length, we construct variations of the above datasets by truncating all excerpts to ten possible lengths ranging from 2 to 192 WordPiece tokens (Wu et al., 2016). In total, we obtain sixty dataset variations: one per sampling method, truncation length, and choice of priming or no priming.

4 Automatic Detection Method

The primary discriminator we employ is a fine-tuned BERT classifier (Devlin et al., 2018). We finetune one instance of BERT per dataset variation described above. For the longest sequence length, $n=192$, we compare BERT’s performance with several simple baselines that have been proposed in other work.

Fine-tuned BERT We fine-tune cased BERT-Large on the task of labeling a sentence as human- or machine- generated. The models are trained

\(^1\)https://github.com/openai/gpt-2-output-dataset
for 15 epochs, with checkpoints saved every 1000 steps, and a batch size of 256. All results are reported on the test set using the checkpoint for which validation accuracy was highest.

**Bag-of-Words** For each sequence, we compute a bag-of-words embedding where each dimension corresponds to a token in GPT-2’s 50,000 token BPE vocabulary, and we count how many times that token appears in the generated sequence. We then train a logistic regression classifier.

**Histogram-of-Likelihood Ranks** Following GLTR (Gehrmann et al., 2019), we compute the probability distribution of the next word given the previous words in a text sequence according to a trained language model (in our case the same GPT-2 model that was used for generation). At each sequence position, we rerank the vocabulary words by likelihood, and record the rank of the ground-truth next word within this list. These ranks are then binned. While GLTR uses 4 bins (top 1, top 5, top 100, >100), we observe higher accuracy when 50 bins are spread uniformly over the possible rankings. Since there are 50,000 vocabulary words, the first bin counts the number of times the actual next word was within the 1,000 mostly likely next words, the second bin counts the 1,001-2,000th, and so on. A logistic regression classifier is trained on top of these histograms.

**Total Probability** Solaiman et al. (2019) propose a very simple baseline consisting of a threshold on the total probability of the text sequence. An excerpt is predicted as machine-generated if its likelihood according to GPT-2 is closer to the mean likelihood over all machine-generated sequences than to the mean of human-written ones.

### Human Detection Method

The human evaluation task is framed similarly to the automatic one. We ask the raters to decide whether a passage of text was written by a human or by a computer algorithm. (Full instructions are in the Appendix.) They are allowed to choose between four options: “definitely” or “possibly” machine-generated, or “definitely” or “possibly” human-written. They are first shown an excerpt of length 16 WordPiece tokens. After they make a guess, the length of the excerpt is doubled, and they are asked the same question again. This continues until the entire passage of length 192 tokens is shown. Passages are equally likely to be human-written or machine-generated, with the machine-generated excerpts being evenly split between the three sampling strategies considered in this paper.

Initially, Amazon Mechanical Turk raters employed for this task, but rater accuracy was poor with over 70% of the “definitely” votes cast for “human” despite the classes being balanced. Accuracy, even for the longest sequences, hovered around 50%. The same study was then performed with university students after walking through ten examples as a group. We will refer to this group as the “expert raters.” Among them, 52.1% of “definitely” votes were cast for human, and accuracy on the longest excerpt length was over 70%.

The human evaluation dataset consisted of 150 excerpts of web text and 50 excerpts each from the three decoding strategies. Each question was shown to at most three raters, leading to 900 total annotations from the untrained workers and 475 from the expert raters. A more detailed breakdown can be found in the Appendix.

### Automatic Detection Results

#### Simple Baselines

Table 1 shows the performance (accuracy and AUC) of fine-tuned BERT classifier and several simple baselines on detecting length-192 sequences with one word of conditioning.

<table>
<thead>
<tr>
<th>Method</th>
<th>BERT acc AUC</th>
<th>BagOfWords acc AUC</th>
<th>HistGLTRBuckets acc AUC</th>
<th>Hist50Buckets acc AUC</th>
<th>TotalProb acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>k40-1wordcond</td>
<td>0.88 0.99</td>
<td>0.79 0.87</td>
<td>0.52 0.52</td>
<td>0.69 0.76</td>
<td>0.61</td>
</tr>
<tr>
<td>p0.96-1wordcond</td>
<td>0.81 0.89</td>
<td>0.60 0.65</td>
<td>0.53 0.56</td>
<td>0.54 0.56</td>
<td>0.63</td>
</tr>
<tr>
<td>p1.0-1wordcond</td>
<td>0.79 0.92</td>
<td>0.59 0.62</td>
<td>0.53 0.55</td>
<td>0.54 0.55</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table 1: Performance (accuracy and AUC) of fine-tuned BERT classifier and several simple baselines on detecting length-192 sequences with one word of conditioning.

Reassuringly, BERT far surpasses all simple baselines, indicating that it is not possible to solve the detection problem without complex sequence-based understanding. The simplest baseline, TotalProb, which makes a decision based on the likelihood of the sequence, performs surprisingly well (over 60% accuracy for all sampling methods) relative to the methods which involve training logistic regression models.

Logistic regression on bag-of-words is the best
Accuracy of BERT Fine-tuned Discriminator

Fraction of BERT Discriminator Errors that are Machine-generated Labeled as Human-written

Figure 1: In (a), accuracy increases as the length of the sequences used to train the discriminator is increased. In (b), we see that the BERT fine-tuned discriminator predicts about the same number of false-positives as false-negatives when trained with samples generated using top-p sampling. However, for top-k, it more often mistakes machine-generated text to be human-written, while for untruncated random sampling the opposite is the case.

of the baselines, beating out the histogram-based methods. While Gehrmann et al. (2019) report an AUC of 0.87 on classifying text as real or generated using logistic regression on the four buckets of the GLTR system, we report AUC between 0.52 and 0.56 for this task. The discrepancy is likely due to the human-written text in our discriminator training set comes from the same distribution as the text used to train the language model, while in GLTR the human text comes from children’s books, scientific abstracts, and newspaper articles. The selection of training data for learned detection systems is crucial. In real-world applications, the choice ought to reflect the genres that builders of text-generation systems are trying to impersonate.

Fine-tuned BERT In Figure 1a, we begin by observing discriminator accuracy as a function of excerpt length and sampling method. As can be intuitively expected, as sequence length increases, so too does accuracy. For unconditioned text decoded with nucleus and untruncated random sampling, we find discriminator accuracy increases from 55%, near random, to about 81% for the longest sequences tested. In contrast, discriminators trained and evaluated on top-k achieve over 80% accuracy even on 16-token excerpts.

Why are top-k’s samples so easy to detect? In Figure 2b, we see the percentage of probability mass concentrated in the k most common token types for each sampling method. While random sampling and nucleus sampling are very similar to human-written texts, we see top-k concentrating up to 80% of its mass in the first 500 most common tokens. The other sampling methods as well human-written texts require at least 1,100 token types for the same. It is clear that top-k’s distribution over unigrams strongly diverges from human-written texts—an easy feature for discriminators to exploit. In fact, See et al. (2019) note that it takes setting k to 1000 to achieve about the same amount of rare word usage and fraction of non-stopword text as as human writing2 This makes it very easy for the model to pick out machine-generated text based on these distributional differences.

One way to help resolve this problem is to add priming text. Doing so causes more rare words to be incorporated into the top-k of the unigram distribution. Adding even a single random word of priming significantly reduces the performance of detectors trained with top-k random sampling. Without priming, a discriminator trained on sequences of length 2 can classify with ~90% accuracy the provenance of the text (Figure 1a). By adding one priming token, accuracy drops to ~65%. Even on the longest 192-length sequences, top-k discriminator accuracy is 6% lower on the primed dataset than the unprimed one.

When generating with nucleus or untruncated random sampling, adding a priming token is not as impactful, as these methods are already sam-

---

2when decoding from the GPT-2 small model with 117M parameters.
Figure 2: In (a), the average (over sequences in the test set) $k$ chosen at each step during generating with nucleus sampling is plotted. Adding a single word of priming strongly impacts the $k$s chosen for the first few positions, but this difference quickly dissipates. In (b), we consider the first token generated in each sequence by top-$k$, and plot what fraction of these are captured by the $k$ most common unique tokens from the vocabulary. Overall, at its first step, top-$k$ concentrates 80% of its probability mass in the 500 most common tokens from the vocabulary.

Sampling from a large fraction (or all) of the probability distribution. This is seen in Figure 2a where at the very first step of unprimed generation, nucleus sampling selects from 3075 possible vocabulary words, and at later positions selects from more than 500. Untruncated random sampling always selects from the entire 50,000 word vocabulary, whereas top-$k$ only selects from $k$.

**Transferability** In Table 2, we show how discriminators trained with samples from one decoding strategy can transfer at test time to detecting samples generated using a different decoding strategy. Unsurprisingly a discriminator trained on top-$k$ generalizes poorly to other sampling methods: accuracy drops to as low as 42.5%, worse than chance. Conversely, training the discriminator with sequences sampled from the untruncated distribution leads to little transferability to detecting top-$k$ samples. Only the discriminator trained with nucleus sampling (a compromise between unmodified sampling and top-$k$) was able to detect sequences from the other sampling strategies.

Perhaps this lack of transferability is related to each discriminator’s calibration. Indeed, the degree to which a discriminator’s average prediction deviates from 50% is a direct indicator of its accuracy. In Table 3, we observe that of the three BERT discriminators, only that trained on top-$p$ samples predicts ‘machine-generated’ on approximately 50% of in-domain examples as expected. This same discriminator’s behavior holds on datasets generated by other sampling strategies as well. In contrast, we observe that discriminators trained on top-$k$ and untruncated random samples severely underestimate the percentage of machine-generated excerpts in out-of-domain datasets. Even within domain (Figure 1b), we find both discriminators heavily favor a single class, increasingly so as the number of tokens increases.

**Human Evaluation** Overall human performance
across all sampling methods is shown in Figure 3b. Even with the multi-paragraph 192-length excerpts, human performance is only at 71.4%, indicating that even trained humans struggle to correctly identify machine-generated text over a quarter a time. However, it is worth noting that our best raters achieved accuracy of 85% or higher, suggesting that it is possible for humans to do very well at this task. Further investigation is needed into how educational background, comfort with English, participation in more extensive training, and other factors can impact rater performance.

To break up the accuracies by sampling method in a way that is comparable to the results shown for the automatic discriminators, we pair each machine-generated example with a randomly selected one of webtext to create a balanced dataset for each sampling strategy. Performance is shown in Figure 3a. Top-k produces the text that is hardest for raters to correctly distinguish, but as shown in Section 6, it is the easiest for our automatic detection systems. Samples from untruncated random sampling and nucleus sampling with $p=0.96$ are equivalently difficult for raters to classify as machine-generated. Our human evaluation results suggest that much lower $p$-values than the 0.92 to 0.98 range proposed in Zellers et al. (2019) might be necessary in order to generate text that is considered significantly more human-like to human raters than the text produced by using the untruncated distribution.

Table 4 gives several examples where human raters and our BERT-based discriminators disagreed. When raters incorrectly labeled human-written text as machine-generated, often the excerpts contained formatting failures introduced when the HTML was stripped out. In the middle two examples, topic drift and falsehoods such as Atlanta being the “information hub of the nation’s capital” allowed humans to correctly detect the generated content. However, in the bottom two examples, the high level of fluency left human raters fooled.

Overall we find that human raters—even expertly trained ones—have consistently worse accuracy than automatic discriminators for all decoding methods and excerpt lengths. In our experiments, randomly-selected pairs of raters agree with each other on a mere 59% of excerpts on average. (In comparison, raters and discriminators agree on 61% to 70% of excerpts depending on the discriminator considered). We surmise that the gap between human and machine performance will only grow as researchers inevitably train bigger, better detection models on larger amounts of training data. However, it is unclear how to go about improving human performance. GLTR proposes providing visual aids to humans to improve their performance at detecting generated-text, but it is unlikely that their histogram-based color-coding will continue to be effective as generative methods get better at producing high-quality text that lacks statistical anomalies.

7 Conclusion

In this work, we study the behavior of automated discriminators and their ability to identify machine-generated and human-written texts. We train these discriminators on balanced binary classification datasets where all machine-
generated excerpts are drawn from the same generative model but with different decoding strategies. We find that, in general, discriminators transfer poorly between decoding strategies, but that training on a mix of data from methods can help. We also show the rate at which discriminator accuracy increases as excerpts are lengthened.

We further study the ability of expert human raters to perform the same task. We find that rater accuracy varies wildly, but has a median of 74%, which is less than the accuracy of our best-performing discriminator. Most interestingly, we find that human raters and discriminators make decisions based on different qualities, with humans more easily noticing semantic errors and discriminators picking up on statistical artifacts. In our experiments, these artifacts are most prominent with top-k sampling. However, any strategy that oversamples high-likelihood words is susceptible. As the \( p \) in nucleus sampling is set increasingly lower than 1, we observe a phenomenon similar to the top-\( k \) effect: Middle and high likelihood words are more likely to be ignored, while lower likelihood words are more likely to be included, as if generated randomly. This suggests that generating high-quality text requires a careful balance of diversity and coverage.

Table 4: Some 192-token examples where at least two expert raters agreed with each other, but were not in agreement with the automatic discriminators. The first row shows examples where the ground-truth was human-written, the second shows machine-generated examples where the corresponding discriminator guessed incorrectly, and the third shows machine-generated examples where the discriminator was correct, but raters got it wrong.

<table>
<thead>
<tr>
<th>Truth</th>
<th>Raters</th>
<th>p1.0</th>
<th>k40</th>
<th>p0.96</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>EDIT-O凯: I guess that’ll work for now. &gt;_&gt; <a href="http://www.teamfortress.com/">http://www.teamfortress.com/</a>! and then</td>
<td>go play the game and experience some of the best online gaming I have ever played.</td>
<td>&lt;Both girls had a really fun time and I had a GREAT time making both of these</td>
<td>costumes. Everything was altered even a little bittingly the pants a darker grey and</td>
<td>painting the boots and skirt. But my piece de resistance would have to be my eldest’s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truth</th>
<th>Raters</th>
<th>p1.0</th>
<th>k40</th>
<th>p0.96</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>H</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>First-off, this thread has done a pretty good job of describing in detail yet another broken touchscreen. That’s the difference between a smartphone and a PC with no prying eyes having to snap shots for the police to find.</td>
<td>What I would like to address is the mindset that</td>
<td>generally surrounds Chrome OS users. To me this is analogous to saying that Apple</td>
<td>does ‘hate their Windows’, or that HP does ‘hate their Macs’ as if it <a href="http://twitter.com/">http://twitter.com/</a>)</td>
<td>(and that quote is from two years ago), that anyone who covers smartphones and tablets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truth</th>
<th>Raters</th>
<th>p1.0</th>
<th>k40</th>
<th>p0.96</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>H</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Exidentia at Earnari, an upcoming Cryptopia event which is currently still in development. Be a part of the first live stream of this year’s event on 15-16 January 2018! &lt;Since the release of v1.22, Exidentia has received a fair amount of user feedback. This event takes place in the underwater Cryptopia they have built. During this event, you will learn about the ocean and areas around it, and be reached by a treasure hunter that helps you explore the different areas. &lt;There will be six different levels in this event that you will become acquainted with. Designing Polar Lava, Ocean Seared Cones and Celestine Floors, Sea Damaged Aerie Bricks, coast Puddle (congpit stopping at red water), Ice Camp and Bugnite. At rotating points, you will learn how to access various types of creatures.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Truth</th>
<th>Raters</th>
<th>p1.0</th>
<th>k40</th>
<th>p0.96</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>Ever since the opening of the North American College of Art Education in 1990, the demand for art education in America has grown steadily, and in recent years we have seen the rise of students that pursue art education not in the classroom but at art academies. This year saw another 50 percent increase in the number of art academies in the United States offering courses – with an additional 10 percent of students in 2017 taking art.</td>
<td>Some major changes have occurred in recent years with regard to the art curriculum and the way students learn, and we will explore each of the prongs for future research:</td>
<td>Prioritizing students without acknowledging the trade-off that ex-</td>
<td>amining poor word choices that are easy for humans to</td>
<td>mimic the human cadence without introducing</td>
</tr>
</tbody>
</table>

Holtzman et al. (2019) explain how a unique at-
References


Jeremy Singer-Vine and Kevin Collier. 2019. The impersonators: Political operatives are faking voter outrage with millions of made-up comments to benefit the rich and powerful. *BuzzFeed News*.


Chris J Vargo, Lei Guo, and Michelle A Amazeen. 2018. The agenda-setting power of fake news: A big data analysis of the online media landscape from


