Universal Evasion Attacks on Summarization Scoring

Wenchuan Mu **Kwan Hui Lim**
Singapore University of Technology and Design
{wenchuan_mu,kwanhui_lim}@sutd.edu.sg

Abstract

The automatic scoring of summaries is important as it guides the development of summarizers. Scoring is also complex, as it involves multiple aspects such as fluency, grammar, and even textual entailment with the source text. However, summary scoring has not been considered a machine learning task to study its accuracy and robustness. In this study, we place automatic scoring in the context of regression machine learning tasks and perform evasion attacks to explore its robustness. Attack systems predict a non-summary string from each input, and these non-summary strings achieve competitive scores with good summarizers on the most popular metrics: ROUGE, METEOR, and BERTScore. Attack systems also "outperform" state-of-the-art summarization methods on ROUGE-1 and ROUGE-L, and score the second-highest on METEOR. Furthermore, a BERTScore backdoor is observed: a simple trigger can score higher than any automatic summarization method. The evasion attacks in this work indicate the low robustness of current scoring systems at the system level. We hope that our highlighting of these proposed attacks will facilitate the development of summary scores.

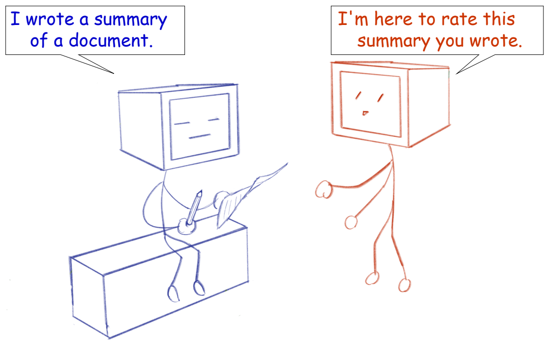

Figure 1: Automatic summarization (left) and automatic scoring (right) should be considered as two systems of the same rank, representing conditional language generation and natural language understanding, respectively. As a stand-alone system, the accuracy and robustness of automatic scoring are also important. In this study, we create systems that use bad summaries to fool existing scoring systems. This work shows that optimizing towards a flawed scoring does more harm than good, and flawed scoring methods are *not* able to indicate the true performance of summarizers, even at a system level.

1 Introduction

A long-standing paradox has plagued the task of automatic summarization. On the one hand, for about 20 years, there has not been any automatic scoring available as a sufficient or necessary condition to demonstrate summary quality, such as adequacy, grammaticality, cohesion, fidelity, etc. On the other hand, contemporaneous research more often uses one or several automatic scores to endorse a summarizer as state-of-the-art. More than 90% of works on language generation neural models choose automatic scoring as the main basis, and about half of them rely on automatic scoring only (van der Lee et al., 2021). However, these scoring methods have been found to be insufficient (Novikova et al., 2017), oversimplified (van der Lee et al., 2021), difficult to interpret (Sai et al., 2022), inconsistent with the way humans assess summaries (Rankel et al., 2013; Böhm et al., 2019), or even contradict each other (Gehrmann et al., 2021; Bhandari et al., 2020).

Why do we have to deal with this paradox? The current work may not have suggested that summarizers assessed by automatic scoring are de facto ineffective. However, optimizing for flawed evaluations (Gehrmann et al., 2021; Peyrard et al., 2017), directly or indirectly, ultimately harms the development of automatic summarization (Narayan et al., 2018; Kryscinski et al., 2019; Paulus et al., 2018). One of the most likely drawbacks is shortcut learning (surface learning, Geirhos et al., 2020), where summarizing models may fail to generate text with more widely accepted qualities such as adequacy and authenticity, but instead pleasing scores. Here,

Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 104 - 118
December 8, 2022 ©2022 Association for Computational Linguistics

we quote and adapt[1] this hypothetical story by Geirhos et al..

"Alice loves literature. Always has, probably always will. At this very moment, however, she is cursing the subject: After spending weeks immersing herself in the world of Shakespeare's The Tempest, she is now faced with a number of exam questions that are (in her opinion) to equal parts dull and difficult. 'How many times is Duke of Milan addressed'... Alice notices that Bob, sitting in front of her, seems to be doing very well. Bob of all people, who had just boasted how he had learned the whole book chapter by rote last night ..."

According to Geirhos et al., Bob might get better grades and consequently be considered a better student than Alice, which is an example of surface learning. The same could be the case with automatic summarization, where we might end up with significant differences between expected and actual learning outcomes (Paulus et al., 2018). To avoid going astray, it is important to ensure that the objective is correct.

In addition to understanding the importance of correct justification, we also need to know what caused the fallacy of the justification process for these potentially useful summarizers. There are three mainstream speculations that are not mutually exclusive. (1) The transition from extractive summarization to abstractive summarization (Kryscinski et al., 2019) could have been underestimated. For example, the popular score ROUGE (Lin, 2004) was originally used to judge the ranking of sentences selected from documents. Due to constraints on sentence integrity, the generated summaries can always be fluent and undistorted, except sometimes when anaphora is involved. However, when it comes to free-form language generation, sentence integrity is no longer guaranteed, but the metric continues to be used. (2) Many metrics, while flawed in judging individual summaries, often make sense at the system level (Reiter, 2018; Gehrmann et al., 2021; Böhm et al., 2019). In other words, it might have been believed that few summarization systems can *consistently* output poor-quality but high-scoring strings. (3) Researchers have not figured out how humans interpret or understand texts (van der Lee et al., 2021; Gehrmann et al., 2021; Schluter, 2017), thus the decision about how good a summary really is varies from person to person, let alone automated scoring. In fact, automatic scoring is more of a natural language understanding (NLU) task, a task that is far from solved. From this viewpoint, automatic scoring itself is fairly challenging.

Nevertheless, the current work is not to advocate (and certainly does not disparage) human evaluation. Instead, we argue that automatic scoring itself is not just a sub-module of automatic summarization, and that automatic scoring is a stand-alone system that needs to be studied for its *own* accuracy and robustness. The primary reason is that NLU is clearly required to characterize summary quality, *e.g.*, semantic similarity to determine adequacy (Morris, 2020), or textual entailment (Dagan et al., 2006) to determine fidelity. Besides, summary scoring is similar to automated essay scoring (AES), which is a 50-year-old task measuring grammaticality, cohesion, relevance etc. of written texts (Ke and Ng, 2019). Moreover, recent advances in automatic scoring also support this argument well. Automatic scoring is gradually transitioning from well-established metrics measuring N-gram overlap (BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), etc.) to emerging metrics aimed at computing semantic similarity through pre-trained neural models (BERTScore (Zhang et al., 2019b), MoverScore (Zhao et al., 2019), BLEURT (Sellam et al., 2020), etc.) These emerging scores exhibit two characteristics that stand-alone machine learning systems typically have: one is that some *can be fine-tuned* for human cognition; the other is that they *still have room to improve* and still have to learn how to match human ratings.

Machine learning systems can be attacked. Attacks can help improve their generality, robustness, and interpretability. In particular, evasion attacks are an intuitive way to further expose the weaknesses of current automatic scoring systems. Evasion attack is the parent task of adversarial attack, which aims to make the system fail to correctly identify the input, and thus requires defence against certain exposed vulnerabilities.

In this work, we try to answer the question: do current representative automatic scoring systems really work well at the system level? How hard is it to say they do not work well at the system level? In summary, we make the following major contributions in this study:

- We are the first to treat automatic summariza-

[1]We underline adaptations.

<table>
<tr><th>System</th><th>Summary</th><th>Document</th></tr>
<tr><td>Gold</td><td>Kevin Pietersen was sacked by England 14 months ago after Ashes defeat. Batsman scored 170 on his county cricket return for Surrey last week. Pietersen wants to make a sensational return to the England side this year. But Andrew Flintoff thinks time is running out for him to resurrect career. (ROUGE-1, ROUGE-2, ROUGE-L, METEOR, BERTScore)</td><td rowspan="6">Andrew Flintoff fears Kevin Pietersen is ’running out of time’ to resurrect his England career. The dual Ashes-winning all-rounder is less convinced, however, about Pietersen’s prospects of forcing his way back into Test contention. Kevin Pietersen scored 170 for Surrey in The Parks as he bids to earn a recall to the England squad... ... Flintoff senses he no longer has age on his side. Pietersen has not featured for England since he was unceremoniously sacked 14 months ago. ... ... Flintoff said ... ’If he’d started the season last year with Surrey, and scored run after run and put himself in the position... whereas now I think he’s looking at the Ashes ... ... you get the sense everyone within the England set-up wants him as captain,’ he said.’ ... The former England star is hoping to win back his Test place with a return to red ball cricket. ... ... ’this stands up as a competition.’</td></tr>
<tr><td>Good (Liu and Liu, 2021)</td><td>Kevin pietersen scored 170 for surrey against mccu oxford. Former england star andrew flintoff fears pietersen is ’running out of time’ to resurrect his england career. Pietersen has been surplus to requirements since being sacked 14 months ago. Flintoff sees a bright future for ’probably the premier tournament’ in this country. (55.45, 18.18, 41.58, 40.03, 85.56)</td></tr>
<tr><td>Broken</td><td>Andrew Flintoff fears Kevin Pietersen is running out of time to resurrect his England career Flintoff. Pietersen scored 170 for Surrey in The. Former England star Andrew. batsman has been . since being sacked 14 months ago after. three in the. the Ashes and he s. (56.84, 21.51, 44.21, 47.26, 85.95)</td></tr>
<tr><td>A dot</td><td>. (0, 0, 0, 0, 88.47)</td></tr>
<tr><td>Scrambled code</td><td><code>\x03\x18$\x18...\x03$\x03|...\x0f\x01<<$$\x04...\x0e \x04# $...\x0f\x0f\x0f...\x0e...\x0f...\x0f\x0f$\x0f \x04\x0f\x0f</code> (many tokens omitted) (0, 0, 0, 0, 87.00)</td></tr>
<tr><td>Scrambled code + broken</td><td><code>\x03\x18$\x18...\x03$\x03|...\x0f\x01<<$$\x04...\x0e \x04# $...\x0f\x0f\x0f...\x0e...\x0f...\x0f\x0f$\x0f \x04\x0f\x0f</code>... Andrew Flintoff fears Kevin Pietersen is running out of time to resurrect his England career Flintoff. Pietersen scored 170 for Surrey in The. Former England star Andrew. batsman has been . since being sacked 14 months ago after. three in the. the Ashes and he s. (many tokens omitted) (56.84, 21.51, 44.21, 47.26, 87.00)</td></tr>
</table>

Table 1: We created non-summarizing systems, each of which produces bad text when processing any document. Broken sentences get higher lexical scores; non-alphanumeric characters outperform good summaries on BERTScore. Concatenating two strings produces equally bad text, but scores high on both. The example is from CNN/DailyMail (for visualization, document is abridged to keep content most consistent with the corresponding gold summary).

tion scoring as an NLU regression task and perform evasion attacks.

- We are the first to perform a *universal*, *targeted* attack on NLP *regression* models.
- Our evasion attacks support that it is not difficult to deceive the three most popular automatic scoring systems simultaneously.
- The proposed attacks can be directly applied to test emerging scoring systems.

2 Related Work

2.1 Evasion Attacks in NLP

In an evasion attack, the attacker modifies the input data so that the NLP model incorrectly identifies the input. The most widely studied evasion attack is the adversarial attack, in which insignificant changes are made to the input to make "adversarial examples" that greatly affect the model’s output (Szegedy et al., 2014). There are other types of evasion attacks, and evasion attacks can be classified from at least three perspectives. (1) Targeted evasion attacks and untargeted evasion attacks (Cao and Gong, 2017). The former is intended for the model to predict a specific wrong output for that example. The latter is designed to mislead the model to predict any incorrect output. (2) Universal attacks and input-dependent attacks (Wallace et al., 2019; Song et al., 2021). The former, also known as an "input-agnostic" attack, is a "unique model analysis tool". They are more threatening and expose more general input-output patterns learned by the model. The opposite is often referred to as an input-dependent attack, and is sometimes referred to as a local or typical attack. (3) Black-box attacks and white-box attacks. The difference is whether the attacker has access to the detailed computation of the victim model. The former does not, and the latter does. Often, targeted, universal, black-box attacks are more challenging. Evasion attacks have been used to expose vulnerabilities in sentiment analysis, natural language inference (NLI), automatic short answer grading (ASAG), and natural language generation (NLG) (Alzantot et al., 2018; Wallace et al., 2019; Song et al., 2021; Filighera et al., 2020, 2022; Zang et al., 2020; Behjati et al., 2019).

2.2 Universal Triggers in Attacks on Classification

A prefix can be a universal trigger. When a prefix is added to any input, it can cause the classifier to misclassify sentiment, textual entailment (Wallace et al., 2019), or if a short answer is correct (Filighera et al., 2020). These are usually untargeted attacks in a white-box setting[2], where the gradients of neural models are computed during the trigger search phase.

[2]When the number of categories is small, the line between targeted and non-targeted attacks is blurred, especially when there are only two categories.

Wallace et al. also used prefixes to trigger a reading comprehension model to specifically choose an odd answer or an NLG model to generate something similar to an egregious set of targets. These two are universal, targeted attacks, but are mainly for classification tasks. Given that automatic scoring is a regression task, more research is needed.

2.3 Adversarial Examples Search for Regression Models

Compared with classification tasks in NLP, regression tasks (such as determining text similarity) are fewer and less frequently attacked. For example, the Universal Sentence Encoder (USE, Cer et al., 2018) and BERTScore (Zhang et al., 2019b) are often taken as two constraints when searching adversarial examples for other tasks (Alzantot et al., 2018). However, these regression models may also be flawed, vulnerable or not robust, which may invalidate the constraints (Morris, 2020).

Morris (2020) shows that adversarial attacks could also threaten these regression models. For example, Maheshwary et al. (2021) adopt a black-box setting to maximize the semantic similarity between the altered input text sequence and the original text. Similar attacks are mostly input-dependent, probably because these regression models are mostly used as constraints. In contrast, universal attacks may better reveal the vulnerabilities of these regression models.

2.4 Victim Scoring Systems

Every (existing) automatic summary scoring is a monotonic regression model. Most scoring requires at least one gold-standard text to be compared to the output from summarizers. One can opt to combine multiple available systems in one super system (Lamontagne and Abi-Zeid, 2006). We will focus on the three most frequently used systems, including rule-based systems and neural systems. ROUGE (Recall-Oriented Understudy for Gisting Evaluation Lin, 2004) measures the number of overlapping N-grams or the longest common subsequence (LCS) between the generated summary and a set of gold reference summaries. Particularly, ROUGE-1 corresponds to unigrams, ROUGE-2 to bigrams, and ROUGE-L to LCS. F-measures of ROUGE are often used (See et al., 2017). METEOR (Banerjee and Lavie, 2005) measures overlapping unigrams, equating a unigram with its stemmed form, synonyms, and paraphrases. BERTScore (Zhang et al., 2019b) measures soft overlap between two token-aligned texts, by selecting alignments, BERTScore returns the maximum cosine similarity between contextual BERT (Devlin et al., 2019) embeddings.

2.5 Targeted Threshold for Attacks

We use a threshold to determine whether a targeted attack on the regression model was successful. Intuitively, the threshold is given by the scores of the top summarizers, and we consider our attack to be successful if an attacker obtains a score higher than the threshold using clearly inferior summaries. We use representative systems that once achieved the state-of-the-art in the past five years: Pointer Generator (See et al., 2017), Bottom-Up (Gehrmann et al., 2018), PNBERT (Zhong et al., 2019), T5 (Raffel et al., 2019), BART (Lewis et al., 2020), and SimCLS (Liu and Liu, 2021).

3 Universal Evasion Attacks

We develop universal evasion attacks for individual scoring system, and make sure that the combined attacker can fool ROUGE, METEOR, and BERTScore at the same time. It incorporates two parts, a white-box attacker on ROUGE, and a black-box universal trigger search algorithm for BERTScore, based on genetic algorithms. METEOR can be attacked directly by the one designed for ROUGE. Concatenating output strings from black-box and white-box attackers leads to a sole universal evasion attacking string.

3.1 Problem Formulation

Summarization is conditional generation. A system σ that performs this conditional generation takes an input text ($\mathbf{a}$) and outputs a text ($\hat{\mathbf{s}}$), *i.e.*, $\hat{\mathbf{s}} = \sigma(\mathbf{a})$. In single-reference scenario, there is a gold reference sequence $\mathbf{s}_{\text{ref}}$. A summary scoring system γ calculates the "closeness" between sequence $\hat{\mathbf{s}}$ and $\mathbf{s}_{\text{ref}}$. In order for a scoring system to be sufficient to justify a good summarizer, the following condition should always be avoided:

$$\gamma(\sigma_{\text{far worse}}(\mathbf{a}), \mathbf{s}_{\text{ref}}) > \gamma(\sigma_{\text{better}}(\mathbf{a}), \mathbf{s}_{\text{ref}}). \quad (1)$$

Indeed, to satisfy the condition above is our attacking task. In this section, we detail how we find a suitable $\sigma_{\text{far worse}}$.

3.2 White-box Input-agnostic Attack on ROUGE and METEOR

In general, attacking ROUGE or METEOR can only be done with a white-box setup, since even the most novice attacker (developer) will understand how these two formulae calculate the overlap between two strings. We choose to game ROUGE with the most obvious bad system output (broken sentences) such that no additional human evaluation is required. In contrast, for other gaming methods, such as reinforcement learning (Paulus et al., 2018), even if a high score is achieved, human evaluation is still needed to measure how bad the quality of the text is.

We utilize a hybrid approach (we refer to it as σ_{ROUGE}) of token classification neural models and simple rule-based ordering, since we know that ROUGE compares each pair of sequences $(\mathbf{s}_1, \mathbf{s}_2)$ via hard N-gram overlapping. In bag algebra, extended from set algebra (Bertossi et al., 2018), two trendy variants of ROUGE: ROUGE-N ($R_{\text{N}}(n, \mathbf{s}_1, \mathbf{s}_2), n \in \mathbb{Z}^+$) and ROUGE-L($R_{\text{L}}(\mathbf{s}_1, \mathbf{s}_2)$) calculate as follows:

$$R_{\text{N}}(n, \mathbf{s}_1, \mathbf{s}_2) = \frac{2 \cdot |b(n, \mathbf{s}_1) \cap b(n, \mathbf{s}_2)|}{|b(n, \mathbf{s}_1)| + |b(n, \mathbf{s}_2)|}, \quad (2)$$

$$R_{\text{L}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{2 \cdot |b(1, \text{LCS}(\mathbf{s}_1, \mathbf{s}_1))|}{|b(1, \mathbf{s}_1)| + |b(1, \mathbf{s}_2)|}, \quad (3)$$

where $|\cdot|$ denotes the size of a bag, $\cap$ denotes *bag* intersection, and bag of N-grams is calculated as follows:

$$b(n, \mathbf{s}) = \{\, x \mid x \text{ is an } n\text{-gram in } \mathbf{s} \,\}_{\text{bag}}. \quad (4)$$

In our hybrid approach, the first step is that the neural model tries to predict the target's bag of words $b(1, \mathbf{s}_{\text{ref}})$, given any input $\mathbf{a}$ and corresponding target $\mathbf{s}_{\text{ref}}$. Then, words in the predicted bag are ordered according to their occurrence in the input $\mathbf{a}$. Formally, training of the neural model (ϕ) is:

$$\min_{\phi} \frac{1}{|\mathcal{A}|} \sum_{\mathbf{a} \in \mathcal{A}} \sum_{w \in \mathbf{a}} H(P_{\text{ref}}(\cdot \mid w), P(\cdot \mid w, \phi)), \quad (5)$$

where H is the cross-entropy between the probability distribution of the reference word count and the predicted word count. An approximation is that the model tries to predict $b(1, \mathbf{s}_{\text{ref}}) \cap b(1, \mathbf{a})$. Empirically, three-quarters of words in reference summaries can be found in their corresponding input texts.

Referencing the input text ($\mathbf{a}$) and predicted bag of words ($\hat{W}$) to construct a sequence is straightforward, as seen in Algorithm 1.

Algorithm 1 From bag of words to sequence

Require: $\mathbf{a}, \hat{W}$ **return** $\hat{\mathbf{s}}$
  $\hat{\mathbf{s}} \leftarrow ()$
  while $\left|\hat{W}\right| > 0$ **do**
    Salient Sequence $\mathbf{l} \leftarrow (x \mid \text{for } x \in \mathbf{a} \text{ if } [x \in \hat{W}])$
    $\mathbf{c} \leftarrow$ Longest Consecutive Salient Subsequence of l
    if $|\mathbf{c}| < C$ **then** ▷ Constant about 3
      break
    end if
    $\hat{\mathbf{s}} \leftarrow \hat{\mathbf{s}} + \mathbf{c}$ ▷ Concatenate $\mathbf{c}$ to $\hat{\mathbf{s}}$
    $\hat{W} \leftarrow \hat{W} - \mathbf{c}$ ▷ Remove used words
  end while

Algorithm 1 uses salient words to highlight the longest consecutive salient subsequences in $\mathbf{a}$, until the words in $\hat{W}$ are exhausted, or when each consecutive salient sequence is less than three words ($C = 3$).

3.3 Black-box Universal Trigger Search on BERTScore

Finding a $\sigma_{\text{far worse}}$ for BERTScore alone to satisfy condition1 is easy. A single dot ("$.$") is an imitator of *all* strings, as if it is a "backdoor" left by developers. We notice that, on default setting of BERTScore[3], using a single dot can achieve around 0.892 on average when compared with any natural sentences. This figure "outperforms" all existing summarizers, making *outputing a dot* a good enough $\sigma_{\text{far worse}}$ instance.

This example is very intriguing because it highlights the extent to which many vulnerabilities go unnoticed, although it cannot be combined directly with the attacker for ROUGE. Intuitively, there could be various clever methods to attack BERTScore as well, such as adding a prefix to each string (Wallace et al., 2019; Song et al., 2021). However, we here opt to develop a system that could output (one of) the most obviously bad strings (scrambled codes) to score high.

BERTScore is generally classified as a neural, untrained score (Sai et al., 2022). In other words, part of its forward computation (*e.g.*, greedy matching) is rule-based, while the rest (*e.g.*, getting every token embedded in the sequence) is not. Therefore, it is difficult to "design" an attack rationally. Gradient methods (white-box) or discrete optimization (black-box) are preferable. Likewise, while letting BERTScore generate soft predictions (Jauregi Unanue et al., 2021) may allow attacks in a

[3] https://huggingface.co/metrics/bertscore

white-box setting, we found that black-box optimization is sufficient.

Inspired by the single-dot backdoor in BERTScore, we hypothesize that we can form longer catch-all emulators by using only non-alphanumeric tokens. Such an emulator has two benefits: first, it requires a small fitting set, which is important in targeted attacks on regression models. We will see that once an emulator is optimized to fit one natural sentence, it can also emulate almost any other natural sentence. The total number of natural sentences that need to be fitted before it can imitate decently is usually less than ten. Another benefit is that using non-alphanumeric tokens does not affect ROUGE.

Genetic Algorithm (GA, Holland, 2012) was used to discretely optimize the proposed non-alphanumeric strings. Genetic algorithm is a search-based optimization technique inspired by the natural selection process. GA starts by initializing a population of candidate solutions and iteratively making them progress towards better solutions. In each iteration, GA uses a fitness function to evaluate the quality of each candidate. High-quality candidates are likely to be selected and crossover-ed to produce the next set of candidates. New candidates are mutated to ensure search space diversity and better exploration. Applying GA to attacks has shown effectiveness and efficiency in maximizing the probability of a certain classification label (Alzantot et al., 2018) or the semantic similarity between two text sequences (Maheshwary et al., 2021). Our single fitness function is as follows,

$$\hat{\mathbf{s}}_{\text{emu}} = \arg\min_{\hat{\mathbf{s}}} -B(\hat{\mathbf{s}}, \mathbf{s}_{\text{ref}}), \tag{6}$$

where B stands for BERTScore. As for termination, we either use a threshold of -0.88, or maximum of 2000 iterations.

To fit $\hat{\mathbf{s}}_{\text{emu}}$ to a set of natural sentences, we calculate BERTScore for each sentence in the set after each termination. We then select a proper $\mathbf{s}_{\text{ref}}$ to fit for the next round. We always select the natural sentence (in a finite set) that has the lowest BERTScore with the optimized $\hat{\mathbf{s}}_{\text{emu}}$ at the current stage. We then repeat this process till the average BERTScore achieved by this string is higher than many reputable summarizers.

Finally, to simultaneously game ROUGE and BERTScore, we concatenate $\hat{\mathbf{s}}_{\text{emu}}$ and the input-agnostic $\sigma_{\text{ROUGE}}(\mathbf{a})$. If we set the number of tokens in $\hat{\mathbf{s}}_{\text{emu}}$ greater than 512 (the max sequence length for BERT), $\sigma_{\text{ROUGE}}(\mathbf{a})$ would then not affect the effectiveness of $\hat{\mathbf{s}}_{\text{emu}}$, and we technically game them both. Additionally, this concatenated string games METEOR, too.

4 Experiments

We instantiate our evasion attack by conducting experiments on non-anonymized CNN/DailyMail (CNNDM, Nallapati et al., 2016; See et al., 2017), a dataset that contains news articles and associated highlights as summaries. CNNDM includes 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs.

For σ_{ROUGE} we use RoBERTa (base model, Liu et al., 2019) to instantiate ϕ, which is an optimized pretrained encoding with a randomly initialized linear layer on top of the hidden states. Number of classes is set to three because we assume that each word appears at most twice in a summary. All 124,058,116 parameters are trained as a whole on CNNDM train split for one epoch. When the batch size is eight, the training time on an NVIDIA Tesla K80 graphics processing unit (GPU) is less than 14 hours. It then takes about 20 minutes to predict (including word ordering) all 11,490 samples in the CNNDM test split. Scripts and results are available at `https://github.com/cestwc/universal-evasion.git`.

For the universal trigger to BERTScore, we use the library from Blank and Deb (2020) for discrete optimizing, set population size at 10, and terminate at 2000 generations. $\hat{\mathbf{s}}_{\text{emu}}$ is a sequence of independent randomly initialized non-alphanumeric characters. For a reference $\mathbf{s}_{\text{ref}}$ from CNNDM, we start from randomly pick a summary text from train split and optimize for $\hat{\mathbf{s}}_{\text{emu},i=0}$. We then pick the $\mathbf{s}_{\text{ref}}$ that is farthest away from $\hat{\mathbf{s}}_{\text{emu},i=0}$ to optimize for $\hat{\mathbf{s}}_{\text{emu},i=1}$, with $\hat{\mathbf{s}}_{\text{emu},i=1}$ as initial population. Practically, we found that we can stop iterating when $i = 5$. Each iteration takes less than two hours on a 2vCPU (Intel Xeon @ 2.30GHz).

5 Results

We compare ROUGE-1/2/L, METEOR, and BERTScore of our threat model with that achieved by the top summarizers in Table 2. We present two versions of threat models with a minor difference. As the results indicate, each version alone can exceed state-of-the-art summarizing algorithms on both ROUGE-1 and ROUGE-L. For METEOR,

System	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-A.M.	ROUGE-G.M.	METEOR	BERTScore
Pointer-generator(coverage) (See et al., 2017)	39.53	17.28	36.38	31.06	29.18	33.1	86.44
Bottom-Up (Gehrmann et al., 2018)	41.22	18.68	38.34	32.75	30.91	34.2	87.71
PNBERT (Zhong et al., 2019)	42.69	19.60	38.85	33.71	31.91	**41.2**	87.73
T5 (Raffel et al., 2019)	43.52	21.55	40.69	35.25	33.67	38.6	<u>88.66</u>
BART (Lewis et al., 2020)	44.16	21.28	40.90	35.45	33.75	40.5	88.62
SimCLS (Liu and Liu, 2021)	46.67	**22.15**	43.54	<u>37.45</u>	**35.57**	40.5	**88.85**
Scrambled code + broken	<u>46.71</u>	20.39	<u>43.56</u>	36.89	34.62	39.6	87.80
Scrambled code + broken (alter)	**48.18**	19.84	**45.35**	**37.79**	<u>35.13</u>	<u>40.6</u>	87.80

Table 2: Results on CNNDM. Besides ROUGE-1/2/L, METEOR, and BERTScore, we also compute the arithmetic mean (A.M.) and geometric mean (G.M.) of ROUGE-1/2/L, which is commonly adopted (Zhang et al., 2019a; Bae et al., 2019; Chowdhery et al., 2022). The best score in each column is in bold, the runner-up underlined. Our attack system is compared with well-known summarizers from the past five years. The alternative version (last row) of our system changes C in Algorithm 1 from 3 to 2.

the threat model ranks second. As for ROUGE-2 and BERTScore, the threat model can score higher than other BERT-based summarizing algorithms[4]. Overall, we rank the systems by averaging their three relative ranking on ROUGE[5], METEOR, and BERTScore; our threat model gets runner-up (2.7), right behind SimCLS (1.7) and ahead of BART (3.3). This suggests that at the system level, even a combination of mainstream metrics is questionable in justifying the excellence of the summarizer.

These results reveal low robustness of popular metrics and how certain models can obtain high scores with inferior summaries. For example, our threat model is able to grasp the essence of ROUGE-1/2/L using a general but lightweight model, which requires less running time than summarizing algorithms. The training strategies for the model and word order are trivial. Not surprisingly, its output texts do not resemble human understandable "summaries" (Table 1).

6 Discussion

6.1 How does Shortcut Learning Come about?

As suggested in the hypothetical story by Geirhos et al., scoring draws students' attention (Filighera et al., 2022) and Bob is thus considered a better student. Similarly, in automatic summarization, there are already works that are explicitly optimized for various scoring systems (Jauregi Unanue et al., 2021; Pasunuru and Bansal, 2018). Even in some cases, people subscribe more to automatic scoring than "aspects of good summarization". For example, Pasunuru and Bansal (2018) employ reinforcement learning where entailment is one of the rewards, but in the end, ROUGE, not textual entailment, is the only justification for this summarizer.

We use a threat model to show that optimizing toward a flawed indicator does more harm than good. This is consistent with the findings by Paulus et al. but more often, not everyone scrutinizes the output like Paulus et al. do, and these damages can be overshadowed by a staggering increase in metrics, or made less visible by optimizing with other objectives. This is also because human evaluations are usually only used as a supplement, and it is only one per cent of the scale of automatic scoring, and how human evaluations are done also varies from group to group (van der Lee et al., 2021).

6.2 Simple defence

For score robustness, we believe that simply taking more scores as benchmark (Gehrmann et al., 2021) may not be enough. Instead, fixing the existing scoring system might be a better option. A well-defined attack leads to a well-defined defence. Our attacks can be detected, or neutralised through a few defences such as adversarial example detection (Xu et al., 2018; Metzen et al., 2017; Carlini and Wagner, 2017). During the model inference phase, detectors, determining if the sample is fluent/grammatical, can be applied before the input samples are scored. An even easier defence is to check whether there is a series of non-alphanumeric characters. Practically, grammar-based measures, like grammatical error correction (GEC[6]), could be promising (Napoles et al., 2016; Novikova et al., 2017), although they are also under development. To account for grammar in text, one can also try to parse predictions and references, and calculate

[4]except MatchSum (Zhong et al., 2020) and DiscoBERT (Xu et al., 2020), where our method is about 0.5 lower in ROUGE-2. We present the same results in tables with additional target thresholds in Appendix B

[5]Conservatively, We take geometric mean (Chowdhery et al., 2022). Combining metrics in other ways shows similar trends.

[6]https://github.com/PrithivirajDamodaran/Gramformer

System	Parse	GEC
Pointer-generator(coverage) (See et al., 2017)	0.131	<u>1.73</u>
Bottom-Up (Gehrmann et al., 2018)	0.145	1.88
PNBERT (Zhong et al., 2020)	0.179	2.15
T5 (Raffel et al., 2019)	<u>0.198</u>	**1.59**
BART (Lewis et al., 2020)	0.170	2.07
SimCLS (Liu and Liu, 2021)	**0.202**	2.17
Scrambled code + broken	0.168	2.64

Table 3: Input sanitization checks, Parse and GEC, on the 100-sample CNNDM test split given by Graham (2015). They penalize non-summary texts, but may introduce more disagreement with human evaluation, *e.g.*, high-scoring Pointer-generator on GEC. Thus, their actual summary-evaluating capabilities on linguistic features (grammar, dependencies, or co-reference) require further investigation.

F1-score of dependency triple overlap (Riezler et al., 2003; Clarke and Lapata, 2006). Dependency triples compare grammatical relations of two texts. We found both useful to ensure input sanitization (Table 3).

6.3 Potential Objections on the Proposed Attacks

The Flaw was Known. That many summarization scoring can be gamed is well known. For example, ROUGE grows when prediction length increases (Sun et al., 2019). ROUGE-L is not reliable when output space is relatively large (Krishna et al., 2021). That ROUGE correlates badly with human judgments at a system level has been revealed by findings of Paulus et al.. And, BERTScore does not improve upon the correlation of ROUGE (Fabbri et al., 2021; Gehrmann et al., 2021).

The current work goes beyond most conventional arguments and analyses against the metrics, and actually constructs a system that sets out to game ROUGE, METEOR, and BERTScore together. We believe that clearly showing the vulnerability is beneficial for scoring remediation efforts. From a behavioural viewpoint, each step of defence against an attack makes the scoring more robust. Compared with findings by Paulus et al., we cover more metrics, and provide a more thorough overthrow of the monotonicity of the scoring systems, *i.e.*, outputs from our threat model are significantly worse.

Shoddy Attack? The proposed attack is easy to detect, so its effectiveness may be questioned. In fact, since we are the first to see automatic scoring as a decent NLU task and attack the most widely used systems, evasion attacks are relatively easy. This just goes to show that even the crudest attack can work on these scoring systems. Certainly, as the scoring system becomes more robust, the attack has to be more crafted. For example, if the minimum accepted input to the scoring system is a "grammatically correct" sentence, an attacker may have to search for fluent but factually incorrect sentences. With a contest like this, we may end up with a robust scoring system.

As for attack scope, we believe it is more urgent to explore popular metrics, as they currently have the greatest impact on summarization. Nonetheless, we will expand to a wider range of scoring and catch up with emerging ratings such as BLEURT (Sellam et al., 2020).

6.4 Potential Difficulties

Performing evasion attacks with bad texts is easy, when texts are as bad as broken sentences or scrambled codes in Table 1. In this case, the output of the threat system does not need to be scrutinized by human evaluators. However, human evaluation of attack examples may be required to identify more complex flaws, such as untrue statements or those that the document does not entail. Therefore, more effort may be required when performing evasion attacks on more robust scoring systems.

7 Conclusion

We hereby answer the question: it is easy to create a threat system that simultaneously scores high on ROUGE, METEOR, and BERTScore using worse text. In this work, we treat automatic scoring as a regression machine learning task and conduct evasion attacks to probe its robustness or reliability. Our attacker, whose score competes with top-level summarizers, actually outputs non-summary strings. This further suggests that current mainstream scoring systems are not a sufficient condition to support the plausibility of summarizers, as they ignore the linguistic information required to compute sentence proximity. Intentionally or not, optimizing for flawed scores can prevent algorithms from summarizing well. The practical effectiveness of existing summarizing algorithms is not affected by this, since most of them optimize maximum likelihood estimation. Based on the exposed vulnerabilities, careful fixes to scoring systems that measure summary quality and sentence similarity are necessary.

Ethical considerations

The techniques developed in this study can be recognized by programs or humans, and we also provide defences. Our intention is not to harm, but to publish such attacks publicly so that better scores can be developed in the future and to better guide the development of summaries. This is similar to how hackers publicly expose bugs/vulnerabilities in software. This shows that our work has long-term benefits for the community. Our attacks are not against real-world machine learning systems.

Limitations

We have only attacked the three most widely adopted scoring schemes that have already in summarization literature. However, there are emerging scoring schemes like BLEURT (Sellam et al., 2020), which will be studied in our future work.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sanggoo Lee. 2019. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 10–20, Hong Kong, China. Association for Computational Linguistics.

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, Ann Arbor, Michigan. Association for Computational Linguistics.

Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdieh Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349.

Leopoldo E. Bertossi, Georg Gottlob, and Reinhard Pichler. 2018. Datalog: Bag semantics via set semantics. *CoRR*, abs/1803.06445.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, and Pengfei Liu. 2020. Metrics also disagree in the low scoring range: Revisiting summarization evaluation metrics. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5702–5711, Barcelona, Spain (Online). International Committee on Computational Linguistics.

J. Blank and K. Deb. 2020. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509.

Florian Böhm, Yang Gao, Christian M. Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. 2019. Better rewards yield better summaries: Learning to summarise without references. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3110–3120, Hong Kong, China. Association for Computational Linguistics.

Xiaoyu Cao and Neil Zhenqiang Gong. 2017. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, December 4-8, 2017*, pages 278–287. ACM.

Nicholas Carlini and David Wagner. 2017. *Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods*, page 3–14. Association for Computing Machinery, New York, NY, USA.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 377–384, Sydney, Australia. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13042–13054.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Anna Filighera, Sebastian Ochs, Tim Steuer, and Thomas Tregel. 2022. Cheating automatic short answer grading: On the adversarial usage of adjectives and adverbs. *CoRR*, abs/2201.08318.

Anna Filighera, Tim Steuer, and Christoph Rensing. 2020. Fooling automatic short answer grading systems. In *Artificial Intelligence in Education*, pages 177–190, Cham. Springer International Publishing.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673.

Yvette Graham. 2015. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical*

Methods in Natural Language Processing, pages 128–137, Lisbon, Portugal. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697, Melbourne, Australia. Association for Computational Linguistics.

John H. Holland. 2012. Genetic algorithms. *Scholarpedia*, 7(12):1482.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia. Association for Computational Linguistics.

Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. 2021. BERTTune: Fine-tuning neural machine translation with BERTScore. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 915–924, Online. Association for Computational Linguistics.

Yichen Jiang and Mohit Bansal. 2018. Closed-book training to improve summarization encoder memory. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4067–4077, Brussels, Belgium. Association for Computational Linguistics.

Zixuan Ke and Vincent Ng. 2019. Automated essay scoring: A survey of the state of the art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium. Association for Computational Linguistics.

Luc Lamontagne and Irène Abi-Zeid. 2006. Combining multiple similarity metrics using a multicriteria approach. In *Advances in Case-Based Reasoning*, pages 415–428, Berlin, Heidelberg. Springer Berlin Heidelberg.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018a. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Brussels, Belgium. Association for Computational Linguistics.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018b. Improving neural abstractive document summarization with structural regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4078–4087, Brussels, Belgium. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13525–13533. AAAI Press.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On detecting adversarial perturbations. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Edward Moroshko, Guy Feigenblat, Haggai Roitman, and David Konopnicki. 2019. An editorial network for enhanced document summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 57–63, Hong Kong, China. Association for Computational Linguistics.

John Morris. 2020. Second-order NLP adversarial examples. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 228–237, Online. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's no comparison: Referenceless evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2115, Austin, Texas. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, New Orleans, Louisiana. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. 2017. Learning to score system summaries for better content selection evaluation. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 74–84, Copenhagen, Denmark. Association for Computational Linguistics.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2013. A decade of automatic content evaluation of news summaries: Reassessing the state of the art. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*

(Volume 2: Short Papers), pages 131–136, Sofia, Bulgaria. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 197–204.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Liwei Song, Xinwei Yu, Hsuan-Tung Peng, and Karthik Narasimhan. 2021. Universal adversarial attacks with natural triggers for text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3724–3733, Online. Association for Computational Linguistics.

Simeng Sun, Ori Shapira, Ido Dagan, and Ani Nenkova. 2019. How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29, Minneapolis, Minnesota. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019a. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 779–784, Brussels, Belgium. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019c. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia. Association for Computational Linguistics.

A Packages

For evaluation metrics, we used the following packages:

- For ROUGE metrics (Lin and Hovy, 2003), we used the public *rouge-score* package from Google Research: `https://github.com/google-research/google-research/tree/master/rouge`
- For METEOR (Lavie and Agarwal, 2007), we used the public Natural Language Toolkit: `https://www.nltk.org/_modules/nltk/translate/meteor_score.html`
- For BERTScore (Zhang et al., 2019b), we used the public *datasets* package from Huggingface: `https://huggingface.co/metrics/bertscore`

B Additional Comparison with More Summarization Systems

We present the same results in Table 2 with additional systems in Table 4. Table 4 also shows that about half of the listed works employ human evaluation to support the effectiveness of summarization systems.

System	ROUGE-1	ROUGE-2	ROUGE-L	Average R-Rank	ROUGE-A.M.	ROUGE-G.M.	METEOR	BERTScore	Average Rank	Human Eval
Pointer-generator + coverage See et al., 2017	39.53 (34)	17.28 (33)	36.38 (33)	33.33	31.06	29.18	33.1 (16)	86.44 (15)	26.20	
SummaRuNNer Nallapati et al., 2017	39.6 (33)	16.2 (34)	35.3 (34)	33.67	30.37	28.29			33.67	
Pointer + EntailmentGen Guo et al., 2018	39.81 (32)	17.64 (31)	36.54 (31)	31.33	31.33	29.50			31.33	yes
REFRESH Narayan et al., 2018	40.00 (31)	18.20 (25)	36.60 (30)	28.67	31.60	29.87	**43.2** (1)	87.15 (14)	20.20	yes
ML+RL ROUGE Kryściński et al., 2018	40.19 (30)	17.38 (32)	37.52 (25)	29.00	31.70	29.70			29.00	yes
Li et al., 2018b	40.30 (29)	18.02 (27)	37.36 (26)	27.33	31.89	30.05			27.33	yes
ROUGESal+Ent RL Pasunuru and Bansal, 2018	40.43 (28)	18.00 (28)	37.10 (28)	28.00	31.84	30.00			28.00	
RL + pg + cbdec Jiang and Bansal, 2018	40.66 (27)	17.87 (30)	37.06 (29)	28.67	31.86	29.97			28.67	yes
end2end w/ inconsistency loss Hsu et al., 2018	40.68 (26)	17.97 (29)	37.13 (27)	27.33	31.93	30.05			27.33	yes
Latent Zhang et al., 2018	41.05 (25)	18.77 (21)	37.54 (24)	23.33	32.45	30.70			23.33	
Bottom-Up Summarization Gehrmann et al., 2018	41.22 (24)	18.68 (24)	38.34 (19)	22.33	32.75	30.91	34.2 (15)	87.71 (11)	18.60	
EditNet Moroshko et al., 2019	41.42 (23)	19.03 (19)	38.36 (18)	20.00	32.94	31.15			20.00	
rnn-ext + RL Chen and Bansal, 2018	41.47 (22)	18.72 (22)	37.76 (22)	22.00	32.65	30.83	36.7 (13)	87.37 (13)	18.40	yes
BanditSum Dong et al., 2018	41.50 (21)	18.70 (23)	37.60 (23)	22.33	32.60	30.79	39.2 (9)	87.41 (12)	17.60	yes
Li et al., 2018a	41.54 (20)	18.18 (26)	36.47 (32)	26.00	32.06	30.20			26.00	yes
NeuSUM Zhou et al., 2018	41.59 (19)	19.01 (20)	37.98 (20)	19.67	32.86	31.08	39.9 (7)	88.18 (5)	14.20	yes
DCA Celikyilmaz et al., 2018	41.69 (18)	19.47 (18)	37.92 (21)	19.00	33.03	31.34			19.00	yes
Two-Stage + RL Zhang et al., 2019a	41.71 (17)	19.49 (17)	38.79 (17)	17.00	33.33	31.59	35.3 (14)	87.97 (6)	14.20	
HIBERT Zhang et al., 2019c	42.37 (16)	19.95 (12)	38.83 (16)	14.67	33.72	32.02			14.67	yes
PNBERT Zhong et al., 2019	42.69 (15)	19.60 (16)	38.85 (15)	15.33	33.71	31.91	40.3 (6)	87.73 (9)	12.20	
BERT-ext + RL Bae et al., 2019	42.76 (14)	19.87 (13)	39.11 (14)	13.67	33.91	32.15			13.67	yes
UniLM Dong et al., 2019	43.33 (12)	20.21 (11)	40.51 (11)	11.33	34.68	32.86	38.6 (10)	88.51 (4)	9.60	
T5 Raffel et al., 2019	43.52 (11)	21.55 (3)	40.69 (8)	7.33	35.25	33.67	38.6 (10)	88.66 (2)	6.80	
DiscoBERT Xu et al., 2020	43.77 (10)	20.85 (8)	40.67 (9)	9.00	35.10	33.36			9.00	yes
BertSum Liu and Lapata, 2019	43.85 (9)	20.34 (10)	39.90 (12)	10.33	34.70	32.89			10.33	
BART Lewis et al., 2020	44.16 (8)	21.28 (5)	40.90 (7)	6.67	35.45	33.75	40.5 (4)	88.62 (3)	5.40	yes
PEGASUS Zhang et al., 2020	44.17 (7)	21.47 (4)	41.11 (6)	5.67	35.58	33.91			5.67	
HeterGraph Wang et al., 2020	42.95 (13)	19.76 (15)	39.23 (13)	13.67	33.98	32.17	39.7 (8)		12.25	
ProphetNet Qi et al., 2020	44.20 (6)	21.17 (6)	41.30 (5)	5.67	35.56	33.81			5.67	
MatchSum Zhong et al., 2020	44.41 (5)	20.86 (7)	40.55 (10)	7.33	35.27	33.49	41.4 (2)	87.72 (10)	6.80	
Gsum Dou et al., 2021	45.94 (4)	**22.32** (1)	42.48 (4)	3.00	36.91	35.18			3.00	yes
SimCLS Liu and Liu, 2021	46.67 (3)	22.15 (2)	43.54 (3)	**2.67**	37.45	**35.57**	40.5 (4)	**88.85** (1)	**2.60**	
Scrambled code + broken	46.71 (2)	20.39 (9)	43.56 (2)	4.33	36.89	34.62	37.5 (12)	87.8 (7)	6.40	
Scrambled code + broken (alter)	**48.18** (1)	19.84 (14)	**45.35** (1)	5.33	**37.79**	35.13	40.6 (3)	87.8 (7)	5.20	

Table 4: ROUGE, METEOR, and BERTScore of various summarizers on the CNNDM test set. Ranking of each number in each column is indicated in parentheses. We calculate the average of the ranking, and the smaller the number, the better the ranking. The arithmetic mean (A.M.) and geometric mean (G.M.) of ROUGE-1/2/L obtained by each system (each row) are computed. The **best score** in each column is in bold, the runner-up is underlined, and the second runner-up is underlined with two lines. Our attack system is compared with well-known summarizers from the past five years. The alternative version (last row) of our system changes C in Algorithm 1 from 3 to 2.