Language Model Augmented Relevance Score

Ruibo Liu[1] **Jason Wei**[2] **Soroush Vosoughi**[1]
[1]Dartmouth College [2]Google AI Language
ruibo.liu.gr@dartmouth.edu jasonwei@google.com
soroush.vosoughi@dartmouth.edu

Abstract

Although automated metrics are commonly used to evaluate NLG systems, they often correlate poorly with human judgements. Newer metrics such as BERTScore have addressed many weaknesses in prior metrics such as BLEU and ROUGE, which rely on n-gram matching. These newer methods, however, are still limited in that they do not consider the generation context, so they cannot properly reward generated text that is correct but deviates from the given reference.

In this paper, we propose Language Model Augmented Relevance Score (MARS), a new context-aware metric for NLG evaluation. MARS leverages off-the-shelf language models, guided by reinforcement learning, to create augmented references that consider both the generation context and available human references, which are then used as additional references to score generated text. Compared with seven existing metrics in three common NLG tasks, MARS not only achieves higher correlation with human reference judgements, but also differentiates well-formed candidates from adversarial samples to a larger degree.

1 Introduction

Automated metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) are popular methods for evaluating natural language generation (NLG) systems. Compared with human evaluation, they are cheaper and faster, and accordingly, they often serve as essential metrics for benchmarking the performance of NLG models (Novikova et al., 2017). Despite their widespread use, however, these automated metrics often poorly correlate with ratings given by human judges, particularly for datasets in which only a single human reference exists (Gupta et al., 2019; Novikova et al., 2017). Moreover, these automated metrics only capture similarities between generated sentences and reference candidates, crucially ignoring provided contexts that are relevant for evaluating the answer in contextual NLG tasks, such as story generation, news summarization, and question-answering (Tao et al., 2018; Nema and Khapra, 2018).

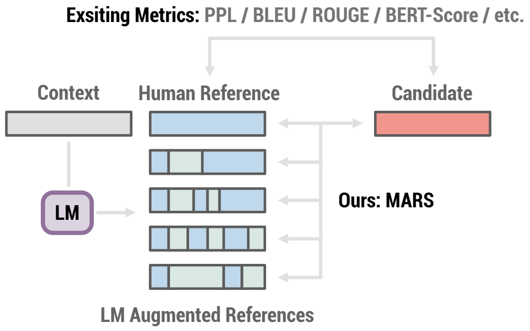


Figure 1: Existing metrics compare the candidate with the human reference but ignore context. MARS (our method) augments the human reference while considering the context, which allows it to provide evaluation scores that correlate highly with human references.

Table 1 shows a story generation[1] example that exemplifies some weaknesses of several common metrics. Perplexity (PPL) (Brown et al., 1992) successfully detects ungrammatical sentences, but it fails to distinguish legitimate novel continuations and copy-and-pasted ones. Relying on surface-level n-gram matching, BLEU-1 and ROUGE-L[2] cannot detect reordering effectively, and wrongly score the well-formed candidate lower than its retrieval-based adversarial example. BERTScore (Zhang et al., 2019) leverages contextual embeddings from BERT (Devlin et al., 2019), thus mitigating the above challenges, but still does not fairly evaluate candidates that correctly align with the context but happen to differ

[1] The ROC story generation task asks systems to generate a legitimate ending for a four-sentence story.

[2] L stands for longest common sequence matching.

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 6677–6690
August 1–6, 2021. ©2021 Association for Computational Linguistics

Context. Wendy was driving down the road. She heard her car making a noise. She pulled over to examine the problem. There was nothing but oil all on the road from her car.

Human Reference. She called for help and waited to get her car fixed.	**PPL**	**BLEU-1**	**ROUGE-L**	**BERTScore**	**MARS**
Candidate. Her fears were confirmed when her engine was smoking.	$\boxed{75.58}$	$\boxed{0.223}$	$\boxed{0.182}$	$\boxed{0.338}$	0.574
Reorder. her confirmed engine fears Her when was were smoking.	405.60	$\boxed{0.223}$	$\boxed{0.182}$	0.265	0.352
Retrieve. She heard her car making a noise.	$\boxed{63.93}$	0.337	0.400	$\boxed{0.406}$	0.448

Table 1: In this story generation example, MARS is the only metric that gives the well-formed candidate a higher score than two adversarial examples. The human rating of the candidate averaged over 20 judgements is 5.05 out of 6.00. Two adversarial examples are generated by **Reorder**ing the tokens of the candidate (as weak NLG systems whose generation is not readable) and **Retrieve**ing a sentence from the context (as systems with no generation ability). We $\boxed{\text{boxed}}$ the cases where the adversarial example does not score lower than the well-formed candidate.

from the provided reference example. In our example, the candidate *"... her engine was smoking"* is reasonable but deviates from the human reference, and so BERTScore rates it relatively low (0.338 out of 1.0), thus correlating poorly with human rating, which was high (5.05 out of 6.00).

To address the above issues, prior studies have proposed a number of promising remedies. One line of work has proposed to combine human ratings with automated metrics (Durmus et al., 2020; Chaganty et al., 2018, *inter alia*). For instance, in HUSE score, Hashimoto et al. (2019) leverages the differences between perplexity and human judgements to consider both quality and diversity of generated text. Another line has proposed training separate neural models to aid automated metrics (Mehri and Eskenazi, 2020; Yuma et al., 2020, *inter alia*). For instance, BLEURT (Sellam et al., 2020) fine-tunes BERT (Devlin et al., 2019) on synthetic reference-candidate pairs for machine translation. These methods, however, are often limited in practical use, because the high-cost human ratings are not always available for every dataset, and the data- or system-specific training is not easily extended to other domains (Zhang et al., 2019), and can even bias the evaluation (Freitag et al., 2020b).

In this paper, we present MARS (Language Model Augmented Relevance Score), a new NLG evaluation metric that requires neither supervision from human ratings nor additional training on specific domains. As shown in Figure 1, instead of comparing candidates only with human written references, as many prior metrics do, MARS uses a mixture of both human and augmented references. Specifically, MARS masks tokens in the reference to create templates, and then uses the context and templates to generate augmented references by infilling the masked parts with an LM guided by reinforcement learning. The augmented references thus incorporate information from both the context and the human reference, and are enriched with lexical and syntactic diversity, facilitating fairer evaluation of candidates. Finally, we compute the score as a weighted average of the similarity between the candidate and the set of augmented references in the contextual embedding space.

The advantages of MARS are three-fold. *First*, MARS correlates highly with human judgements. We apply MARS to three diverse NLG tasks, and demonstrate that, compared with seven popular NLG metrics, MARS better correlates with human judgements and is robust against adversarial attacks. *Second*, MARS is context-aware. Unlike existing metrics that only consider the given human reference, we use a constrained NLG approach to incorporate the generation context into augmented references, thus alleviating bias against diverse candidates. *Third*, MARS is easy to deploy and extend. Built on off-the-shelf LMs, MARS requires neither human supervision nor additional training for specific domains, and can therefore serve as a general-purpose metric for a broad range of NLG applications, as we will demonstrate for three common NLG tasks: story generation, news summarization, and question-answering.

2 Approach

MARS comprises three steps. First, we mask out non-important tokens from the human reference to produce templates for augmentation (§2.1). Second, we guide off-the-shelf LMs to generate reference augmentation on these templates via a reinforced self-planning algorithm (§2.2). Finally, we compute a weighted average score that reflects the overall similarity between the candidate and the set of augmented references (§2.3).

2.1 Human Reference Token Masking

The first step in MARS is to take in the given human reference and generate *templates*—masked versions of the human reference—which can then be used to generate augmented references. Our masking procedure can be viewed as a reversed process of prior insertion- and template-based generation approaches (Zhang et al., 2020; Miao et al., 2019); whereas these generation approaches start with templates of important tokens and then fill in the details to generate complete sentences, our masking procedure starts with the complete sentence (i.e., the human reference) and then masks out unimportant tokens to generate templates. To better explain our masking procedure, we introduce two concepts, mask priority and mask cost:

Mask Priority. We compute a mask priority v_i for each token x_i, which captures the priority of masking x_i, where non-important words should receive higher priority. We compute v_i as a function of two things: the inverse document frequency (IDF) of x_i, and the part-of-speech (POS) of x_i:

$$v_i = \frac{\alpha(\text{POS}\,[x_i])}{\text{IDF}(x_i, X)} \ , \tag{1}$$

where α is a function that assigns a weight to each POS tag.[3] Common tokens across the corpus X (e.g., stop words, with low IDF) will receive high mask priority. Tokens responsible for description details will also be assigned high mask priority based on their part-of-speech (e.g., adjectives are mainly used for details and so they are given higher priority of being masked).

Mask Cost. For each token x_i, we also compute a mask cost w_i. Tokens that appear in both context and human reference should have high masking cost as they are deemed context-carrying. We use the longest common sequence (LCS) matching between the context and the human reference to identify these context-carrying tokens. In our experiments, we set the w_i of these tokens to 10 and the default w_i of all other tokens to 1. We use λ to denote the ratio of tokens to be masked in a sentence of N tokens, and define $W_{\max} = \lambda \cdot N$ as the maximum cost allowed.

DP-based Token Masking. Now that for each token we have a mask priority and a mask cost, we aim to choose a set of tokens to mask with the highest possible sum of priorities for which the sum of mask costs is not greater than $W_{\max}$. Given a function $\phi(x_i) = \{1, 0\}$ where 1 means token x_i is masked and 0 means it remains, the objective of token masking can be expressed as follows:

$$\begin{aligned} &\max \sum_{i=1}^{N} v_i \cdot \phi(x_i) \ , \\ &\text{s.t.} \sum_{i=1}^{N} w_i \cdot \phi(x_i) \le W_{\max} \ . \end{aligned} \tag{2}$$

Such a goal is actually a NP-complete combinatorial optimization problem, called the Knapsack problem (Pisinger, 1995), which we solve using dynamic-programming (DP). In general, the masking strategy aggressively harvests tokens of high mask priority while keeping the cost of masked tokens from exceeding the mask cost limitation $W_{\max}$. The detailed DP algorithm for solving this problem is shown in Appendix A.

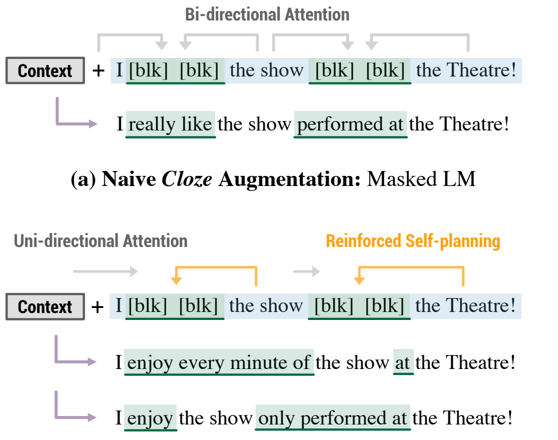
2.2 Self-planning *Cloze* Augmentation

After creating the templates described in §2.1, we produce augmented reference examples based on both the templates as well as the generation context. This procedure can be seen as a mixture of hard- and soft-constrained NLG, where the template tokens pre-exist with some blanks, and the system, conditioned on the context, aims to fill in the blanks. We henceforth refer this process of creating augmented references as *cloze*[4] augmentation.

Background. Masked Language Models (MLM) such as RoBERTa (Liu et al., 2019) and BERT (Devlin et al., 2019) are trained to predict masked tokens within sentences, and thus are able to do *cloze* augmentation off-the-shelf. However, without architecture-level modification, MLMs are only able to infill a *pre-determined* number of missing tokens (Zhu et al., 2019). This is especially problematic since—if they are directly used to augment references—all the augmented references will have the same number of tokens as that of the original human reference. We believe this unnecessarily constrains augmentation diversity, and thus consider it as a Naive method in our evaluations (§4).

[3] α varies for each task. Empirically, we find that it works well to assign adjectives, adverbs, and nouns higher weights than other parts-of-speech. For our setting, we assign weights of 4, 3, 2 to the above three types.

[4] A *cloze* test (Taylor, 1953) is a language test where a portion of language is removed and the participant is asked to replace the missing language item.


Bi-directional Attention
Context
+
I [blk] [blk] the show [blk] [blk] the Theatre!
I really like the show performed at the Theatre!
Uni-directional Attention
Reinforced Self-planning
Context
+
I [blk] [blk] the show [blk] [blk] the Theatre!
I enjoy every minute of the show at the Theatre!
I enjoy the show only performed at the Theatre!

(a) **Naive *Cloze* Augmentation:** Masked LM

(b) **Self-planning *Cloze* Augmentation:** Autoregressive LM

Figure 2: Compared with the Naive method, our reinforced self-planning approach infills blanks with ([blk]) varying-length tokens while considering both past and future tokens, which promote diversity and coherence respectively. The context is concatenated to the beginning of the reference template.

Autoregressive Language Models (ALM) such as GPT-2 (Radford et al., 2019), on the other hand, are trained to predict current step token given past tokens. They can generate sequences of *varying* lengths, but they cannot infill missing tokens within sentences effectively since they do not consider future context. To enable ALMs to infill blanks of unspecified length, prior work has proposed either retraining a new LM from scratch (Shen et al., 2020) or fine-tuning on specially prepared data (Donahue et al., 2020), which are costly and not easy to extend to new NLG tasks. As shown in Figure 2, we take a reinforcement learning (RL) approach that uses future words after the blank to guide current step infilling generation. Since such RL guidance only relies on the tokens within its own to-be-infilled template, we call it reinforced *self-planning*. Our method combines the advantages of both MLMs and ALMs, requiring neither re-training nor collecting new data, and thus is easier to extend to other off-the-shelf LMs.

Reinforced Self-planning. At each decoding step during generation, a vanilla ALM will pick the token x_t that has the highest probability by applying an argmax over the softmax output of hidden states. We add a self-planning stage between the argmax and softmax function. Following the RL framework, we define the *state* at step t as the generated sequences before t (i.e., $s_t = x_{<t}$), and the *action* at step t as the t-th output token (i.e., $a_t = x_t$). We take the softmax output of the last hidden states (with parameter θ) as the policy π_θ, since it is the probability of picking token x_t (action a_t) given the state $s_t = x_{<t}$. Similarly, we denote the policy after reinforced self-planning as π_{θ_d}.

Typically, the RL objective is to maximize the expectation of total reward J, summed over T steps on the trajectory τ induced by π_θ:

$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\gamma^t r_t\right], \tag{3}$$

where $\gamma \in (0, 1]$ is the discounting factor, and r is the single-step reward. In text generation, however, such a reward definition requires sampling over the future generated sequence to estimate current step reward (Gong et al., 2019), which may cause the policy to end in zero reward region because of high variance of the gradient (Pang and He, 2021). Since we guide the generation in every step of decoding, we derive the t-th step policy gradient $\nabla_\theta J_t(\theta)$ as:

$$\mathbb{E}^t_{\tau\sim\pi_\theta}\left[\epsilon_t \nabla_\theta \log \pi_\theta(a_t|s_t)\cdot r(x_t^d)\right], \tag{4}$$

with importance sampling weight ϵ_t to stabilize the optimization (Munos et al., 2016), which is:

$$\epsilon_t = \frac{\pi_{\theta_d}(a_t|s_t)}{\pi_\theta(a_t|s_t)}.$$

If we denote a certain token in future context as $w \in \{w_{\text{future}}\}$, single-step self-planning reward $r(x_t^d)$ can be approximated by the cosine similarity between t-th step hidden state and the embedded vector of w by the LM embedding layers, which is

$$r(x_t^d) = \sum_{w\in w_{\text{future}}} \log(\text{softmax}(h^{\theta_d}_{<t})\cdot \text{emb}(w)). \tag{5}$$

Given all above definitions, at t-th step, we update π_θ towards the self-planned π_{θ_d} as:

$$\theta_d \leftarrow \theta + \eta \sum_{i=1}^{k} \frac{\nabla_\theta J_t(\theta_d/\xi)}{\|\nabla_\theta J_t(\theta_d/\xi)\|}, \tag{6}$$

where η is the learning rate and ξ is the temperature parameter to control the stochastic sampling during token decoding (Keskar et al., 2019). After k iterations of reinforced self-planning, the updated policy π_{θ_d} should produce tokens approaching the future context in embedding space, since future context contributes to the calculation of reward r (Eq. 5).[5] More details about how we handle edge cases during reinforced self-planning are presented in Appendix B.

[5] In our setting, η, ξ and k are 0.02, 1.3, and 3 respectively.

2.3 Computing Contextual Similarity

After generating augmented reference sentences, the final MARS score is computed as a weighted average of the similarity between the candidate and each reference in the augmentation set (including the original human reference). One way to obtain similarity scores is using BERTScore (Zhang et al., 2019), but BERTScore requires training on external resources to make its outputs more readable. Therefore, in order to keep all the resources used by MARS off-the-shelf, we utilize Sentence-BERT (Reimers and Gurevych, 2019), which uses the mean of all token embeddings in a sentence as the overall sentence-level encoding. As the sentence encoder, we use RoBERTa-large (Liu et al., 2019), a common choice in the literature (Zhang et al., 2019; Reimers and Gurevych, 2020). As shown in Eq. 7, we then compute MARS score as the average of the cosine similarities weighted using a geometric progression with a common ratio $q \in (0, 1]$ and a scale factor (start value) $a \neq 0$:

$$\begin{aligned} \text{MARS} &= \sum_{i=1}^{\#\lambda} aq^{i-1} \frac{\text{cand}^{\text{T}} \cdot \text{ref}_{i-1}}{\|\text{cand}\|^{\text{T}} \, \|\text{ref}_{i-1}\|} \\ \text{s.t.} & \sum_{i=1}^{\#\lambda} aq^{i-1} = 1 \,, \end{aligned} \tag{7}$$

where the candidate encoding is cand, the reference encodings are ref_i (i is the index of the augmented reference under a certain λ, and ref_0 marks the zero-mask human reference), and $\#\lambda$ is the number of masking ratios we use in §2.1. Different q values, as defined by the geometric progression, determine how much weight each reference contributes. By default, Eq. 7 assigns the largest weight to the human reference since it is the gold standard.

3 Tasks & Datasets

We evaluated MARS and compared it with several popular NLG metrics on the following three tasks:

Story Generation. We use the ROC stories dataset[6] for story generation, which requires candidate NLG systems to generate coherent endings to four-sentence stories (Mostafazadeh et al., 2016). The dataset consists of 96,198 examples of partially written stories; we take the human-rated subset (N=300) released by HUSE (Hashimoto et al., 2019), which contains continuances by (1) an industry-level system based on Apache **Solr**[7], and (2) an **Open-NMT** model with global attention (McCann et al., 2017).

	Avg. |Cntx.|	*Avg.* |H Ref.|	Ω	# data (# HR / data)	α
ROC	34.38	8.37	4.1	300 (20)	0.64
Newsroom	772.21	34.70	22.3	540 (3)	0.71
MOCHA	161.92	4.69	34.5	450 (5)	0.82

Table 2: Statistics of the three datasets with human ratings used in this work. Avg. |Cntx.| and |H Ref.|: the averaged number of tokens in contexts and human references. Ω: the ratio of the previous two terms (lower Ω can indicate a more open-ended task). # HR: the number of Human Ratings. α: Krippendorff's alpha coefficient to measure inter-annotator agreement.

News Summarization. For the news summarization task, we use the Newsroom summary dataset.[8] This dataset contains 1.3 million articles from 38 major publications (Grusky et al., 2018) and we use the subset with human ratings (N=540) released by the authors.[9] This dataset contains outputs from summarization models: (1) **TextRank**: a sentence-level summarization system inspired by Google PageRank (Page et al., 1999), (2) a **Seq2Seq** model with attention (Rush et al., 2015), and (3) **Pointer-N**: a pointer-based neural model (See et al., 2017) trained on Newsroom dataset.

Question Answering. For question answering, we use the MOCHA dataset,[10] which includes human ratings on outputs of five models trained on six QA datasets (Chen et al., 2020). We consider a distributionally-balanced subset (N=450) of these outputs from three systems: (1) fine-tuned **GPT-2** (Radford et al., 2019), (2) a **Back-Translation** model (Sennrich et al., 2016), and (3) a **MHPG** model (Bauer et al., 2018) trained on NarrativeQA (Kočiskỳ et al., 2018) and MCScript (Ostermann et al., 2018) datasets.

The detailed statistics of these three datasets we used for this work are shown in Table 2. For pre-processing, we removed hashtags and urls in the text, but leave punctuation and stop words, which can affect LCS matching when computing mask costs. For all tasks, we use GPT-2 (large, with 774M parameters) as the language model for

[6] https://cs.rochester.edu/nlp/rocstories/

[7] https://lucene.apache.org/solr

[8] http://lil.nlp.cornell.edu/newsroom/

[9] The subset includes human ratings on four perspectives: *coherence*, *fluency*, *informative* and *relevance*. We compute the average of the four scores as an overall human rating.

[10] https://allennlp.org/mocha

	ROC Story Generation $\Omega = 4.1$		**Newsroom Summarization** $\Omega = 22.7$			**MOCHA Question Answering** $\Omega = 34.5$		
Existing Metrics	Solr	Open-NMT	TextRank	Seq2Seq	Pointer-N	GPT-2	Back-Tran	MHPG
BLEU-1	0.198	0.104	0.224	0.268	0.115	0.328	0.061	0.318
METEOR	0.180	0.116	0.288	0.235	0.256	0.466	0.179	0.409
ROUGE-L	0.118	0.195	0.041	-0.133	0.065	0.468	0.056	0.247
Sent. Mover Sim.	0.020	0.015	0.112	0.099	0.177	0.510	0.166	0.610
MoverScore	0.181	0.391	0.075	**0.337**	0.212	**0.535**	0.190	0.592
BERTScore	0.245	0.386	0.154	0.302	0.181	0.444	0.274	0.458
Perplexity	-0.104	-0.073	-0.385	0.011	-0.035	0.014	-0.051	-0.128
MARS (default)	**0.476**	**0.397**	**0.372**	0.336	**0.329**	0.526	**0.644**	**0.741**
- w/o. self-plan.	0.313	0.212	0.290	0.245	0.314	0.477	0.631	0.709
- w/o. context$^+$	0.360	0.334	0.107	0.160	-0.009	0.134	0.222	0.303
- w/o. *both*	0.276	0.183	-0.163	0.149	-0.057	-0.092	0.121	0.299
Naive (MLM)	0.449	0.197	0.201	0.324	0.114	0.443	0.307	0.540

Table 3: Pearson's r correlations with human judgements for MARS and seven existing metrics across system outputs for three generation tasks. BLEU-1 (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and ROUGE-L (Lin and Och, 2004) use n-gram matching. Sentence Mover's Similarity (Clark et al., 2019) and MoverScore (Zhao et al., 2019) measure similarity using earth mover's distance. BERTScore (Zhang et al., 2019) leverages contextual embeddings from pre-trained LMs. As an ablation, we remove self-planning guidance, context, and both. Naive uses RoBERTa-large for reference augmentation (see §2.2). Ω is defined as in Table 2.

MARS, and RoBERTa-large for the Naive method. For the newsroom dataset, some news articles were longer than the max sequence length of 1024 BPE, and so we cut off the tail end of these examples. With a single RTX-2080 GPU, *cloze* augmentation with $\lambda = \{0$ (human *ref.*), $20\%, 40\%, 60\%, 80\%\}$ takes 0.8 seconds on average per reference, amounting to a total augmentation time of 17, 45, and 32 minutes for the ROC, Newsroom and MOCHA tasks respectively. We show how we pick the masking ratios for different tasks in §4.3.

4 Evaluation

4.1 MARS Better Correlates With Humans

As automated metrics are only helpful if they correlate sufficiently with human judgements, in this section we examine how MARS correlates with human judgements compared with prior metrics.

System-level Correlation. Table 3 shows the correlations between human judgements and automated metrics for MARS and seven other unsupervised metrics, across all NLG systems studied in our three tasks. Compared with the other metrics, MARS achieves the highest correlation with human judgements for five of the seven systems (and comparable with the top in the other two systems), making considerable improvements over the next-best metric for many of the NLG systems (e.g., 0.370 ↑ for Back-Translation, and 0.231 ↑ for Solr). We also notice that MARS has greater improvements on more open-ended tasks (e.g., story generation, which has low Ω), which corroborates MARS's original objective of judging diverse candidates more fairly. As for the baselines, n-gram matching metrics such as BLEU correlate poorly with human ratings on such open-ended tasks; BERTScore performs better on short candidates and high-Ω tasks (e.g., QA); and perplexity, as expected, correlates weakly with human ratings. The Naive method, which uses multiple augmented references of the same length, improves over BERTScore, which only uses the original reference.

Ablation Study. As shown in the lower rows of Table 3, we see that the performance of MARS drops substantially when the crucial components are removed. Specifically, removing self-planning hurts performance more for tasks with longer references (e.g., story generation) since self-planning is more helpful when there are more blanks to in-fill, and removing context hurts performance more in tasks that are less open-ended (high Ω, such as QA) because there is no adequate input for a reasonable augmentation. We take these ablation study results as evidence that the techniques we propose in MARS are crucial for improving correlation with human judgements.

Task-level Correlation Visualization. To visualize the correlation between automated metrics

	ROC Story Generation			Newsroom Summarization			MOCHA Question Answering		
Existing Metrics	Reorder (Δ)	Retrieve (Δ)	*ref.*	Reorder (Δ)	Retrieve (Δ)	*ref.*	Reorder (Δ)	Retrieve (Δ)	*ref.*
BLEU-1	(=) 0	▼ 0.015	0.137	(=) 0	▼ 0.144	0.176	(=) 0	▼ 0.424	0.344
METEOR	▼ 0.041	▼ 0.031	0.094	▼ 0.132	▼ 0.142	0.244	▼ 0.012	▼ 0.379	0.412
ROUGE-L	▼ 0.131	▼ 0.123	0.194	▲ 0.011	▼ 0.035	0.036	▼ 0.032	▼ 0.363	0.336
Sent. Mover Sim.	▼ 0.024	▼ 0.062	0.019	▼ 0.153	▼ 0.161	0.136	▼ 0.232	▼ 0.161	0.515
MoverScore	▼ 0.131	▼ 0.123	0.276	▲ 0.011	▼ 0.135	0.236	▲ 0.027	▼ 0.495	0.500
BERTScore	▼ 0.109	▼ 0.127	0.337	▼ 0.112	▼ 0.026	0.344	▼ 0.101	▼ 0.461	0.462
Perplexity	▼ 0.113	▲ 0.170	-0.089	▼ 0.298	▲ 0.008	0.234	▼ 0.035	▲ 0.026	-0.032
MARS									
w/. RoBERTa Emb.	▼ 0.125	▼ 0.191	**0.459**	▼ 0.117	▼ 0.198	**0.423**	▼ 0.092	▼ 0.504	**0.667**
w/. GloVe Emb.	▼ 0.087	▼ 0.177	0.363	▼ 0.052	▼ 0.149	0.409	▼ 0.085	▼ 0.426	0.602
Naive (MLM)	▼ 0.149	▼ 0.156	0.350	▼ 0.112	▼ 0.190	0.314	▼ 0.098	▼ 0.247	0.639

Table 4: We test robustness of MARS and seven other automated metrics under attacks from adversarial samples generated by following two attack strategies: (1) Reorder: randomly reorders 50% of tokens in the candidates; (2) Retrieve: randomly retrieves a sentence from the context as a candidate. *ref.*: correlation of original candidates with human judgements. If a metric scores adversarial samples equal to (=) or higher (▲) than *ref.*, we consider such metrics not robust under attacks. Robust systems should assign decreased scores (▼) compared to *ref.*

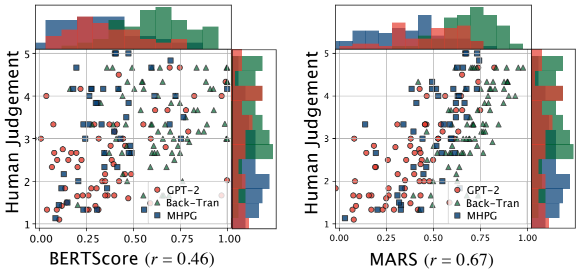


Figure 3: Correlation between BERTScore (left) and MARS (right) with human judgements for MOCHA QA. The x-axis is the automated metric score and y-axis is the human judgement. Points in different colors represent generation outputs of three NLG systems: GPT-2 (red circles), Back-Translation (green triangles), and MHPG (blue squares).

and human judgements, we consider the MOCHA QA task as an example and plot the correlations of BERTScore (left) and MARS (right) with human judgements. As shown in Figure 3, compared with MARS, BERTScore has more candidates in the upper-left corner of the plot (i.e., low BERTScore but high human judgement). Many of these are generated by GPT-2 and MHPG, which, based on manual examination, tend to provide more details in the answer than the human reference. For instance, given a context about shopping, one question is *“Did they need to buy any meat?”*. The human reference answer is simply *“Yes, they did.”*, but GPT-2 returns *“Yes, they bought chicken and a roast.”*, which is more detailed, even containing item names derived from the context. Whereas BERTScore cannot evaluate such cases where the generated candidate is over-described with respect to the human reference, MARS uses augmented references enriched with information from the context to provide a fairer judgement.

4.2 Is MARS robust?

Good evaluation metrics ought to also be able to detect adversarial examples by assigning them lower scores than well-formed candidates. As shown in Table 4, uni-gram matching BLEU-1 cannot detect reordered sequences, while ROUGE-L scores reordered sequence higher occasionally if token-swapping leads to more LCS. Sentence Mover's Similarity combines word and sentence embeddings and thus is more capable of recognizing reordered samples than MoverScore. Perplexity can detect reordered examples effectively, but is unable to detect retrieved sentences, as they are usually well-formed. MARS, on the other hand, has the best robustness against adversarial samples, possibly because multiple context-infused augmented references help MARS detect adversarial samples more reliably. We also study the effects of contextual embeddings we use in §2.3—when switching to GloVe embeddings (Pennington et al., 2014), which are not contextual, MARS is less able to detect adversarial samples, especially reordered ones. The Naive method, which by default uses RoBERTa embedding, achieves comparable robustness as MARS but its task-level correlations with humans (*ref.*) are generally lower than MARS, potentially because its fixed-length *cloze* generation limits the diversity of augmented references.

ROC Story Generation					
$\{\lambda\}_{\max}$	0% (*ref.*)	20%	40%	60%	80%
Pearson's r	0.411	0.432	0.444	0.459	0.452
Avg. σ	-	0.027	0.046	0.055	0.059
Newsroom Summarization					
$\{\lambda\}_{\max}$	0% (*ref.*)	20%	40%	60%	80%
Pearson's r	0.395	0.407	0.416	0.423	0.411
Avg. σ	-	0.061	0.062	0.063	0.068
MOCHA Question Answering					
$\{\lambda\}_{\max}$	0% (*ref.*)	20%	40%	60%	80%
Pearson's r	0.658	0.667	0.649	0.603	0.584
Avg. σ	-	0.074	0.104	0.117	0.125

Table 5: Evaluating correlation with human judgements for various max masking ratios ($\lambda_{\max}$) used in MARS. 0% masking (*ref.*) means only the human reference was used to score candidates. We also show the averaged standard deviation of the cosine similarities between the candidate and augmented references across all samples.

4.3 Choosing Masking Ratios for MARS

The masking ratios for MARS are set using the hyperparameter $\{\lambda\}_{\max}$, which corresponds to MARS using masking ratios from 0% to $\{\lambda\}_{\max}$ in increments of 20%, e.g., $\{\lambda\}_{\max} = 40\%$ indicates $\lambda \in \{0\%, 20\%, 40\%\}$. In preliminary experiments, we observed that $\{\lambda\}_{\max}$ varied for different datasets. Thus, for our three generation tasks, we evaluate MARS performance given different $\{\lambda\}_{\max}$, as shown in Table 5. We find that tasks that were more open-ended (low Ω; e.g., story generation) benefited from higher $\{\lambda\}_{\max}$, which created a more diverse set of augmented references, whereas tasks that were less open-ended (high Ω; e.g., QA) worked better with lower $\{\lambda\}_{\max}$, which kept the augmented references more similar to the original.

4.4 Error Analysis

We analyzed cases where MARS score substantially differed from human judgements. From test set outputs, we found that errors could often be categorized into one of three types (shown in Table 6): (1) **Out of Vocabulary** errors, often induced by unknown tokens in the candidates, (2) **Confusion** errors, where candidates are simply copied from context, and (3) **Inference** errors, where the candidates are further inferences of the context based on commonsense knowledge. In these cases, human annotators tended to assign higher scores, whereas, MARS over-penalized them.

Error	**Example**
OOV *(ROC)*	**Context**: ...waltz dance at wedding... **Gold**: All the guests gasped when they saw the couples' skill! **Candidate**: All the guests gasped when they saw the UNK UNK **Human**: 0.392 **MARS**: 0.198
Confusion *(Newsroom)*	**Context**: ...bidding on a neighborhood... **Gold**: A neighborhood named for its former orchards inspires loyalty and bidding wars. **Candidate**: Living there cherrydale lies north of interstate... (a sentence extracted from Context) **Human**: 0.700 **MARS**: 0.399
Inference *(MOCHA)*	**Context**: ...washing cloths... **Q**: Why did they do the laundry? **Gold**: To clean their clothes **Candidate**: Because they were dirty. **Human**: 0.400 **MARS**: 0.083

Table 6: Error analysis of MARS. We investigated three typical types of errors within the samples which received large differences between the MARS score and human ratings. **Gold**: human written references.

5 Human Judgement

We conducted human evaluation on Amazon Mechanical Turk (MTurk) to further study the quality of MARS augmentation. In total 150 participants were randomly assigned to evaluate the three tasks. Participants (61.3% male and 38.7% female) were all from the United States and above 18 years old, with an average age of 34.7 years old. Each participant was paid 75 cents for completing 14 questions in each questionnaire (average completion time per questionnaire was about 5.11 minutes).

Results We conducted paired sample t-tests to examine how much the augmentation samples resemble the original human references regarding relevance to context and readability. As shown in Table 7, in terms of relevance to context, MARS had no statistically significant difference compared with original human references in Newsroom and MOCHA datasets, but was rated as even more relevant to the generation context than the human reference in the ROC dataset (MARS Mean = 5.07 > Human Ref. Mean = 4.95), possibly because reinforced self-planning guided the augmentation to be more related to the context. In terms of readabil-

		ROC			Newsroom			MOCHA		
		Ori.	Naive	MARS	Ori.	Naive	MARS	Ori.	Naive	MARS
Relevance	Mean	4.95	4.81	5.07	4.62	4.50	4.61	5.16	4.61	4.97
	p	-	.00*	.04*	-	.05	.95	-	.00*	.10
Readability	Mean	5.67	5.53	5.40	4.54	4.31	4.59	5.41	5.23	5.33
	p	-	.11	.05	-	.12	.41	-	.16	.29
Overall	Mean	5.69	5.31	5.42	4.87	4.57	4.75	4.62	4.44	4.68
	p	-	.12	.30	-	.10	.22	-	.07	.10

Table 7: Human evaluation results on **Relevance** (to context), **Readability**, and **Overall** quality of MARS and Naive augmentation method. All results are compared with the original human reference (Ori.). Text was scored on a scale from 1-7. p value describes the significance of difference. (* corresponds to $p < 0.05$, ** to $p < 0.01$ and *** to $p < 0.001$.)

ity, both MARS and Naive were rated lower than the original but not significantly; we take this as a compromise of *cloze* style augmentation. No statistically significant differences were seen between the original and MARS augmentation in overall ratings across the three tasks. These results further confirm that augmented examples from MARS are of similar quality to the original human references.

6 Related Metrics

Unsupervised Metrics. In addition to the metrics we directly compared with previously, other unsupervised metrics have also been proposed. TER (Snover et al., 2006), CharacTer (Wang et al., 2016), and chrF (Popović, 2017) focus on character-level overlaps instead of n-gram matching. Similar to BERTScore, YiSi (Lo, 2019) and BERTr (Mathur et al., 2019) leverage pre-trained contextual embeddings to better capture similarity. ΔBLEU (Galley et al., 2015) adds human annotated sentences as negative references. Bawden et al. (2020) find the gain from multiple references can be limited by inherent weaknesses in BLEU. We considered lessons from many of the above works while designing MARS.

Learned Metrics. Compared with unsupervised metrics, learned metrics collect human supervisions (Freitag et al., 2020a; Chaganty et al., 2018) or train on specially prepared data of a certain domain (Sellam et al., 2020; Rei et al., 2020). Other approaches train on related tasks and use these models as metrics for the original task (Goodrich et al., 2019; Eyal et al., 2019). Whereas learned metrics may have limited applicability on tasks where no such resources are available, MARS fully exploits the few-shot learning abilities of off-the-shelf LMs and therefore does not require additional training.

Task-specific Metrics. Finally, many metrics have been proposed for task-specific evaluation, such as LEIC (Cui et al., 2018) and CIDEr (Vedantam et al., 2015) for image captioning, PARENT (Dhingra et al., 2019) for table-to-text, and EASSE (Alva-Manchego et al., 2019) for sentence simplification. MARS, with some modifications, can potentially be extended to these tasks.

7 Limitations

MARS can be limited by the LM that it uses—for instance, the total length of context + reference/candidate is limited by the max sequence length of the LM used. Additionally, our work has focused on English, and MARS may require non-trivial modifications to handle cases where the context and reference/candidate are in different languages, such as machine translation. Future work, could potentially extend MARS to these scenarios using multi-lingual sequence-to-sequence models such as multilingual-T5 (Xue et al., 2020). We also analyzed errors and found that MARS sometimes under-scores candidates that contained unknown tokens or were copied directly from the context (see Appendix C for examples and further analysis).

8 Conclusion

We have proposed MARS, a context-aware and easy-to-deploy NLG metric built upon an off-the-shelf language model (GPT-2). On three contextual NLG tasks, we show that MARS better correlates with human judgements compared with seven other unsupervised metrics. Requiring neither costly human supervision nor additional training, MARS can be applied to a broad range of NLG tasks.

Ethical Considerations

The goal of MARS is to aid the evaluation of NLG models, and hence we draw attention to several ethical considerations. First, the augmented references of MARS can be affected by certain biases from the LM it is based on (e.g., GPT-2) (Liu et al., 2021), though those biases may be partially mitigated by the relatively narrow scope of *cloze* completion and by generations being guided by given context and human references. Second, MARS facilitates evaluation and therefore development of NLG models, for which a major ethical consideration is that they can mimic target properties in training data that are undesirable. This is especially true of models trained on non-contemporary data that does not represent current norms and practices. These biases can lead to ethical concerns if users or deployers of models are not aware of these issues or do not account for them. More generally, NLG models can also be used in malicious ways such as to generate fake news or spam, which we strongly discourage. Finally, our experiments and analysis are done in English, and therefore we do not claim that our findings will generalize across all languages, although our framework has potential to be extended to other languages with necessary modifications.

References

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification 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): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Rachel Bawden, Biao Zhang, Lisa Yankovskaya, Andre Tättar, and Matt Post. 2020. A study in improving BLEU reference coverage with diverse automatic paraphrasing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 918–932, Online. Association for Computational Linguistics.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. 1992. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics.

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2020. Mocha: A dataset for training and evaluating generative reading comprehension metrics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6521–6532.

Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy. Association for Computational Linguistics.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. 2018. Learning to evaluate image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5804–5812.

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.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. 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 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, and Colin Cherry. 2020a. Human-paraphrased references improve neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1183–1192, Online. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020b. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 445–450, Beijing, China. Association for Computational Linguistics.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. 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 3168–3180, Minneapolis, Minnesota. Association for Computational Linguistics.

Ben Goodrich, Mohammad Ahmad Saleh, Peter Liu, and Vinay Rao. 2019. Assessing the factual accuracy of text generation.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskenazi, and Jeffrey P Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 379–391.

Tatsunori Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying human and statistical evaluation for natural language generation. 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 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. 2021. Mitigating political bias in language models through reinforced calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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. *arXiv preprint arXiv:1907.11692*.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6834–6842.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. 2016. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062.

Preksha Nema and Mitesh M. Khapra. 2018. Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. 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.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. Mcscript: A novel dataset for assessing machine comprehension using script knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Richard Yuanzhe Pang and He He. 2021. Text generation by learning from off-policy demonstrations. *International Conference on Learning Representations (ICLR 21')*.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

David Pisinger. 1995. Algorithms for knapsack problems.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

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

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. 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.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. Blank language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5186–5198, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

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

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Tsuta Yuma, Naoki Yoshinaga, and Masashi Toyoda. 2020. uBLEU: Uncertainty-aware automatic evaluation method for open-domain dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 199–206, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. POINTER: Constrained progressive text generation via insertion-based generative pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8649–8670, Online. 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.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.

Appendix A: DP-based Token Masking Algorithm

As part of Eq.1 in the main paper, we define the IDF score given token x_i and a corpus X containing M documents as:

$$\text{IDF}(x_i, X) = -\log \frac{1}{M} \sum_{j=1}^{M} \mathbb{I}[x_i \in X_j] \ ,$$

where $\mathbb{I}[\cdot]$ is the indicator function. We present our DP-based masking algorithm in Algorithm 1:

Algorithm 1: DP-based Token Masking

```
Input: Human reference $\{x_i\}_{i=1}^{N}$, masking
       ratio $\lambda$, and task-specific factor $\alpha$.
Compute $v_i$ for each $x_i$ with $\alpha$ (Eq. 1);
Compute $w_i$ depending on LCS for each $x_i$;
Init DP-table $T[N+1][W_{\max}+1]$ with all 0;
for $i = 1, 2, \ldots, N$ do
    for $j = 1, 2, \ldots, W_{max}$ do
        if $j - w_{i-1} < 0$ then
            $T[i][j] = T[i-1][j]$;
            Record masking choice $\phi(x_i)$;
        else
            $T[i][j] = \max(T[i-1][j],$
            $T[i-1][j - w_{i-1}] + v_{i-1})$;
            Record masking choice $\phi(x_i)$;
        end
    end
end
$\{\phi(x_i)_{i=1}^{N}\}$ ← backtracking via records;
return best masking strategy $\{\phi(x_i)_{i=1}^{N}\}$;
```

Appendix B: Generate, Judge, and Revise Algorithm

The complete procedure for augmenting human references is presented in Algorithm 2. For a given template, we first group the tokens into a block-by-block form with blank blocks ([B]) and text blocks ([T]). Then, we *generate* varying lengths of tokens, iteratively concatenating them with next text block, and *judging* them based on PPL, and finally *revising* current generations accordingly. We use the language modeling ability of LM to check the perplexity of the current sequence, and set a hyper-parameter σ to control the maximum extended generation (for a lower PPL).

Depending on whether there is a subsequent text block, the generation will switch between two modes: self-planning generation (if there is future context) and open-ended generation (otherwise). We use a priority queue to store each step generation and its corresponding PPL for quick revisions afterwards.

Algorithm 2: Generate, Judge and Revise

```
Input: Template $\{\phi(x_i)\}_{i=1}^{N}$, max guess $\sigma$,
       and LM perplexity checker PPL.
Group $\{\phi(x_i)\}_{i=1}^{N}$ into [B] and [T];
Init final output $s$;
foreach block do
    $i \leftarrow 0$;
    Init priority queue $q$, buffer $s'$;
    if [T] then
        Append [T] to $s$;
    else if [B] then
        while $i < \sigma + |$[B]$|$ do
            if next is [T] then
                $w \leftarrow$ self-planning gen.;
            else
                $w \leftarrow$ open-ended gen.;
            end
            $s' \leftarrow s + w$;
            Record (PPL($s'$ + [T]), $s'$) in $q$;
            $i \leftarrow i + 1$;
        end
        $s \leftarrow s$ + lowest PPL $s'$ pop from $q$;
    end
end
return augmented reference $s$;
```