Perception Score: A Learned Metric for Open-ended Text Generation Evaluation

Jing Gu[1], **Qingyang Wu**[1], **Zhou Yu**[2]
[1] University of California, Davis
[2] Columbia University
{jkgu,wilwu}@ucdavis.edu, zy2461@columbia.edu

Abstract

Automatic evaluation for open-ended natural language generation tasks remains a challenge. We propose a learned evaluation metric: Perception Score. It utilizes a pre-trained model and considers context information for conditional generation. Perception Score assigns a holistic score along with uncertainty measurement. We conduct experiments on three open-ended conditional generation tasks and two open-ended unconditional generation tasks. Perception Score achieves state-of-the-art results on all the tasks consistently in terms of correlation with human evaluation scores.

Introduction

With the recent advances in natural language generation (NLG) (Radford et al. 2019; Vaswani et al. 2017), automatic evaluation has drawn more attention from the research community. However, current automatic evaluation metrics often have limitations in measuring the actual generation quality. In contrast, human evaluation is considered more reliable than the automatic evaluation metrics, but it is expensive and time-consuming, especially for the generation tasks that require extensive domain expertise (Celikyilmaz, Clark, and Gao 2020). Therefore, it is important to have an evaluation metric that provides an acceptable proxy for quality and is relatively affordable.

N-gram overlapping such as BLEU (Papineni et al. 2002) and ROUGE (Lin 2004) are the most widely used metrics across NLG tasks. However, these metrics only consider word-form variation and often fail to capture the deeper semantic meaning. Thus, these metrics usually have a low correlation with human judgment on open-ended NLG tasks (Lowe et al. 2017; Zhou and Xu 2020; Cui et al. 2018; Chaganty, Mussmann, and Liang 2018).

Recently, various metrics have also been proposed to measure the similarity between the references and the generations beyond the lexical level. BERTScore (Zhang* et al. 2020) and MoverScore (Zhao et al. 2019) are proposed to use the contextual embedding from a large pre-trained neural model to measure the semantic similarity. BLEURT (Sellam, Das, and Parikh 2020) mixes various existing metrics to improve the robustness. While these metrics obtain a good result on machine translation tasks, they may have a less satisfactory performance in open-ended natural language generation tasks because a sentence could be of high quality but different from the given reference in semantic level.

Context of Conversation
Speaker A: I usually drink soda or milk in the morning.
Speaker B: Cool, what do you want today?
Model Response
I prefer the latter this morning.
Reference Response
I would like some milk today.

Context of Story
I tried going to the park the other day. The weather seemed nice enough for a walk. Within minutes of getting there I started sneezing. My eyes were watery and it was hard to breathe.
Model Generated Ending
My allergies were too bad and I had to go back home.
Reference Ending
I took my allergy pill and felt better afterwards.

Figure 1: Demonstrations of the importance of incorporating context into sentence quality evaluation process. First example: similar semantics but different surface forms. Second example: different semantics but both reasonable.

Another problem in current metrics is that they usually do not incorporate context information into the evaluation process (Celikyilmaz, Clark, and Gao 2020). The context may be important when evaluating a generation model's quality in open-ended natural language generation tasks, as exemplified in Figure 1. First, the semantic meaning of a sentence could be context-dependent. Taking context into consideration would lead to a more accurate similarity measurement between the generated sentence and the reference one. Secondly, there can exist multiple valid target sentences in open-ended generation tasks given the same context. Therefore previous metrics with only one reference sentence may lead to misjudgment of other reasonable endings.

We propose Perception Score, an automatic evaluation metric for open-ended text generation tasks. Perception Score is a learned metric that measures the overall quality of a generative model via assigning a single holistic score based on the distribution difference between the generations and the references. The overall evaluation process contains three steps. First, we incorporate context into the generated sentence and the reference one to get a more comprehensive textual representation. Then it utilizes a pre-trained neural language model to distinguish between each generated sentence and its corresponding reference sentence. Secondly, it calculates the data uncertainty and model uncertainty separately and presents a total uncertainty estimation for each sample. Finally, each sample is weighted with its uncertainty to compute the overall score. Thus, Perception Score provides direct observation of the generation model quality by comparing the generated sentences and the reference ones.

We validate Perception Score on five tasks, including three conditional generation tasks, i.e., DailyDialog, ROCStories and Large Movie Review Conditional, and two unconditional tasks, i.e., Large Movie Review Unconditional and COCO Image Captions. The experiments demonstrate that our method has a higher correlation than the previous automatic evaluation metrics such as BLEU, BERTScore, and BLEURT. Moreover, Perception Score shows strong robustness when evaluating the quality of human-written endings on the ROCStories dataset.

Related Work

Text evaluation is an essential topic in natural language generation (NLG). Researchers have proposed different types of automatic evaluation metrics to facilitate the evaluation and the development of NLG models.

N**-gram matching** is the most used evaluation method in the NLG task. BLEU is a commonly used metric via measuring the weighted geometric mean of n-gram precision in various conditional generation tasks, such as machine translation. In unconditional generation tasks such as in COCO Image Captions task, where a generated sentence does not have a fixed reference sentence, BLEU utilizes all samples in the dataset as references for each generated sentence. METEOR measures sentence quality based on the harmonic mean of the unigram precision and recall. Some variants of METEOR also consider surface forms, stemmed forms, and meanings. ROUGE (Lin 2004) calculates n-gram recall in text summarization task. However, these n-gram matching metrics only consider local consistency and do not consider the sentence's overall grammaticality or sentence meaning.

Perplexity is another commonly used metric in open-ended generation tasks such as chit-chat. It measures the probability distribution in which a generative model predicts the reference sentence. However, it does not directly reflect the generation sentence quality (Celikyilmaz, Clark, and Gao 2020).

Text embedding based metrics are proposed in recent researches. FED (de Masson d'Autume et al. 2019) computes the Frechet distance (Semeniuta, Severyn, and Gelly 2018) via utilizing the embeddings trained from a Universal Sentence Encoder (Cer et al. 2018). However, it can not accurately evaluate a generative model with a large temperature (Cai et al. 2020). BERTScore (Zhang* et al. 2020) utilizes contextual embeddings in a large pre-trained neural model for each token, then applies greedy matching to maximize the cosine similarity between references and candidates. MoverScore (Zhao et al. 2019) combines the contextual embeddings from a pre-trained model with Word Mover's Distance to evaluate text generation. These metric utilizes large pre-trained models to evaluate the semantic similarity between the references and candidates. However, a generated sentence could be different from the reference while still have high-quality.

Learned metrics are also proposed to evaluate sentence quality beyond word surface level. Blend (Ma et al. 2017) uses an SVM model to combine different existing evaluation metrics. RUSE (Shimanaka, Kajiwara, and Komachi 2018) evaluates machine translation by training sentence embeddings on data obtained in other tasks. Cui et al. (2018) trains a neural network conditioning on image to distinguish between human and machine-generated captions. In more recent works, BLEURT utilizes a BERT model fine-tuned on various automatic metrics to evaluate the generated sentence's quality. Zhou and Xu (2020) proposes to compare a pair of generated sentences by fine-tuning BERT.

Another popular research trend is to train an automatic metric to calibrate automatic evaluation metrics and human judgments. HUSE (Hashimoto, Zhang, and Liang 2019) connects statistical evaluation with the human evaluation to evaluate summarization and chit-chat dialogue. Building a user simulator (Shi et al. 2019) could also be a potential direction. Lowe et al. (2017) trains an evaluation model based on large human judgment to score dialog response. However, these methods require massive human annotation as supervision and risk poor generalization to new domains.

Method

The evaluation process of Perception Score contains three steps. First, a Perception Model will learn to assign a sample-level Perception Score based on a generated sentence's overall quality. Then we calculate the uncertainty for the sample-level Perception Score. Finally, we aggregate each sample with its corresponding uncertainty and get the system-level Perception Score for the generation model.

Sample-Level Perception Score

For a generation system, denote its generations on the dataset as $G = \langle \hat{x}_1, \ldots, \hat{x}_n \rangle$, where $\hat{x}_i$ is the generation for i-th sample in the dataset, and denote the corresponding reference as $R = \langle x_1, \ldots, x_n \rangle$. An automatic metric evaluates a generation system by comparing G with R.

The perception of realness usually includes various criteria, and one perspective does not guarantee the overall quality. However, previous metrics usually focus on the similarity on a specific aspect between the generated sentence and the reference ones. For example, BLEU focuses on word surface overlapping, and BERTScore focuses on token-level semantic similarity. We define Perception Score as a function δ that perceives the realness of the generation. Since standard of realness changes across NLG tasks, in contrast to the static evaluation metrics, Perception Score should be learned to fit various NLG tasks.

We define this realness by aggregating the inter-distances

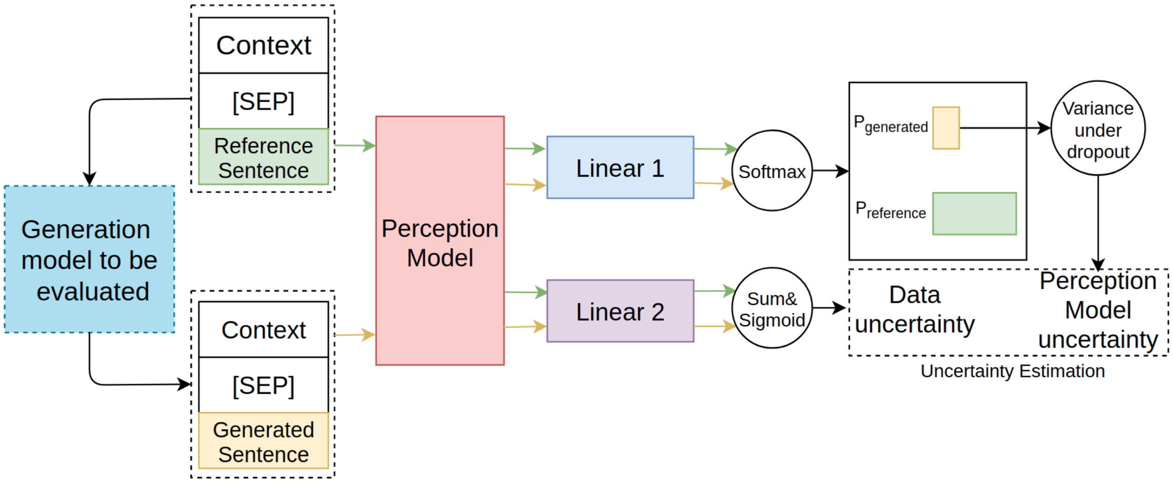

Figure 2: The main structure of Perception Score. We enhance the representation by incorporating context. A Perception Model with strong understanding ability learns the difference between a generated sentence and the corresponding reference sentence. Then we calculate data uncertainty and Perception Model uncertainty.

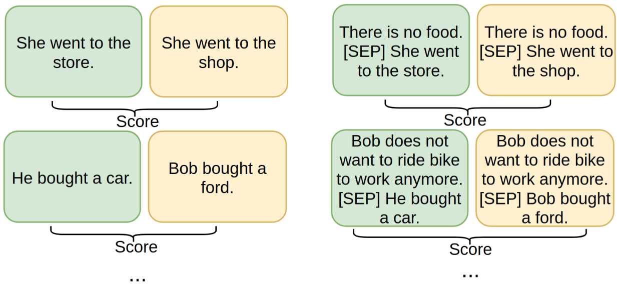
She went to the store.
She went to the shop.
Score
He bought a car.
Bob bought a ford.
Score
...
There is no food. [SEP] She went to the store.
There is no food. [SEP] She went to the shop.
Score
Bob does not want to ride bike to work anymore. [SEP] He bought a car.
Bob does not want to ride bike to work anymore. [SEP] Bob bought a ford.
Score
...

Figure 3: Common evaluation metrics (left) do not consider context information, Perception Score (right) takes context into consideration and therefore the semantic meaning is more comprehensive.

between every pair of samples. In mathematics terms, such realness measurement is defined as:

$$\delta_\theta = \arg\max_{\theta} \mathbb{E}_{\hat{x}\sim\mathbb{P}_G, x\sim\mathbb{P}_R}[\delta_\theta(\hat{x}, x)] \quad (1)$$

As exemplified in Figure 1, the semantic meaning of the generation is often context-dependent, we incorporate context information into G and R to get a more comprehensive semantic representation, as in Figure 3. We now have G^+ as $\langle\{c_1, \hat{x}_1\}, \dots, \{c_n, \hat{x}_n\}\rangle$, and have R^+ as $\langle\{c_1, x_1\}, \dots, \{c_n, x_n\}\rangle$, where c_i is the context of i-th sample in the dataset.

Note that Eq 1 can be viewed as one form of Earth Mover's Distance (EMD). The goal of Perception Score is to compute a reasonable and efficient approximation of EMD. For Earth-Movers' Distance, we have

$$W(\mathbb{P}_{G^+}, \mathbb{P}_{R^+}) = \inf_{\gamma\in\prod(\mathbb{P}_{G^+}, \mathbb{P}_{R^+})} \mathbb{E}_{(x,\hat{x})\sim\gamma}[||x-\hat{x}||] \quad (2)$$

where $\prod(\mathbb{P}_{G^+}, \mathbb{P}_{R^+})$ denotes the set of all joints distributions. γ indicates how much "mass" must be transported from $\hat{x}$ to x in order to transform the distribution $\mathbb{P}_{G^+}$ to the distribution $\mathbb{P}_{R^+}$. The EM distance is the optimal transport plan.

Equation 2 is intractable due to the infinite possibility of the joint distribution. Following Arjovsky, Chintala, and Bottou (2017), we convert the formula by using the supremum over the 1-Lipschitz functions.

$$W(\mathbb{P}_{G^+}, \mathbb{P}_{R^+}) = \sup_{|f|_L\leq 1} \mathbb{E}_{x\sim\mathbb{P}_{R^+}}[f(x)] - \mathbb{E}_{\hat{x}\sim\mathbb{P}_{G^+}}[f(\hat{x})] \quad (3)$$

Eq. 3 is equivalent to optimize the following problem:

$$W(\mathbb{P}_{G^+}, \mathbb{P}_{R^+}) = \max_{\theta} \mathbb{E}_{\hat{x}\sim\mathbb{P}_{G^+}, x\sim\mathbb{P}_{R^+}}[f_\theta(x) - f_\theta(\hat{x})] \quad (4)$$

with a gradient penalty loss to fulfill the Lipschitz constrain:

$$L_{GP} = \mathbb{E}_{\hat{x}\sim\mathbb{P}_{G^+}, x\sim\mathbb{P}_{R^+}}[(||\nabla_{\hat{x},x} D(\hat{x}, x)||_2 - 1)^2] \quad (5)$$

However, directly optimizing Eq. 4 is problematic in text space. Unlike in images where the resolution is fixed at the beginning, In natural language processing, the inputs are discrete and of varied lengths. Then there exists a much larger space for the possible generated sequences than the actual valid sequences. To avoid divergence , we bound the maximum earth-mover distance to be 1 by normalizing the output. We denote the output of the Perception model of reference sample and generation sample as $T(x)$ and $T(\hat{x})$, l_1 as linear forward layers, then we have $f_\theta(x)$ as $P_{generated}$ and $f_\theta(\hat{x})$ as $P_{reference}$,

$$P_{generated} = f_\theta(\hat{x}) = \frac{e^{l_1(T(\hat{x}))}}{e^{l_1(T(\hat{x}))} + e^{l_1(T(x))}} \quad (6)$$

and

$$P_{reference} = f_\theta(x) = \frac{e^{l_1(T(x))}}{e^{l_1(T(\hat{x}))} + e^{l_1(T(x))}} \quad (7)$$

Optimizing $P_{reference}$ is equivalent to optimizing EMD distance in Eq. 4. The Perception Model learns the difference of a generated sentence and its corresponding reference sentence in the training set, and evaluate the generation model quality on the test set. While in a general classification task where a more powerful model usually leads to higher accuracy, in our evaluation process, a powerful model leads to a more accurate approximation of the EMD between generated sentences and reference sentences. Please refer to appendix for more details.

Sample-Level Perception Score Uncertainty

To reflect different samples' influence on the final system-level Perception score, we also calculate each sample's uncertainty and use them as weights during score aggregation. We utilize both data uncertainty and model uncertainty in Perception Score evaluation process.

Following Gal and Ghahramani (2016), we use the variance of the prediction under different dropout settings to measure model uncertainty of the Perception Model. We use the variance of these results as the measure of the model uncertainty. When $m \to 1$, Perception Score is confident about its prediction.

$$m = 1 - \mathrm{Var}(P_{reference}) \quad (8)$$

As for data uncertainty, following DeVries and Taylor (2018), we use additional fully-connected layers l_2 to get the confidence degree for the Perception Score and use a sigmoid function to constrain uncertainty between 0–1.

$$c = sigmoid(l_2(T(\hat{x}) + T(x))). \quad \hat{x} \in \mathbb{P}_{G^+}, x \in \mathbb{P}_{R^+} \quad (9)$$

Then the optimization objective is adjusted by interpolating between the original Perception Score and the data uncertainty:

$$p' = c \cdot P_{reference} + (1 - c)y. \quad (10)$$

where the y is the target distribution.

Instead of optimizing $P_{reference}$ in 7, we optimize p':

$$\mathcal{L}_t = -\sum_{i=1}^{M} \log(p'_i). \quad (11)$$

As suggested by DeVries and Taylor (2018), we utilize an extra regularizer to prevent the network from minimizing the loss by always choosing data uncertainty c as 0.

$$\mathcal{L}_c = -\log(c). \quad (12)$$

We simply add data uncertainty and model uncertainty to form the final uncertainty for each sample.

The final loss is the weighted sum of the losses we mentioned:

$$\mathcal{L} = \mathcal{L}_t + \lambda \mathcal{L}_c + \beta L_{GP}. \quad (13)$$

The tuning of λ and β is identical in Arjovsky, Chintala, and Bottou (2017) and DeVries and Taylor (2018). Please check appendix for the hyper-parameter tuning details.

System-Level Perception Score

During the test time, as introduced before, we weight each generated sample with its corresponding data uncertainty and model uncertainty. Then the Perception Score for G is:

$$P_{sys} = \sum_{i=1}^{n} w_i * P^i_{generated} \quad (14)$$

where w is calculated by model uncertainty m and data uncertainty c.

$$w_i = \frac{c_i + m_i}{\sum_{k=1}^{n} c_k + m_k} \quad (15)$$

Higher P_{sys} means higher generation system quality. When P_{sys} is around 0.5, meaning the Perception Model considers the generated sentences and the references ones have the similar quality.

Experiment Setting

We evaluate Perception Score in three open-ended conditional generation and two open-ended unconditional generation tasks. We use RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$ as the Perception Model in all the experiments. We choose the various transformer-based generative models with different training hyper-parameter choices to be evaluated by Perception Score. For baseline metrics, we consider classical metrics including BLEU (Papineni et al. 2002) and perplexity, and recently proposed state-of-the-art metrics including Comparator Evaluator (Zhou and Xu 2020), BERTScore (Zhang* et al. 2020), MoverScore (Zhao et al. 2019), and BLEURT (Sellam, Das, and Parikh 2020).

Conditional Generation

We use DailyDialog(Li et al. 2017) dataset, ROCStories-dataset (Mostafazadeh et al. 2016) and IMDB review conditional dataset. Large Movie Review Conditional dataset comes from Large Movie Review Dataset v1.0 (Maas et al. 2011). For each review in the Large Movie Review Dataset, we set the last sentence as the review ending and the remaining text as context. The purpose of ROCStories is to generate an open-ended story ending for a four-sentence short context. The purpose of Large Movie Review Unconditional is to finish the review based on a movie review context. The purpose of DailyDialog is to generate an appropriate response based on dialog history. In these tasks, Perception Score is expected to assign higher scores to high quality generations.

Unconditional Generation

For unconditional generation task, we use COCO Image Captionsdataset (Chen et al. 2015) and IMDB review unconditional dataset. Large Movie Review Unconditional dataset comes from Large Movie Review Dataset v1.0 (Chen et al. 2015). The purpose of Large Movie Review Conditional is to generate an original and high-quality movie, and the purpose of COCO Image Captions is to generate high-quality image caption unconditionally. Unlike in conditional generation, there is no fixed reference for a system generation. Other metrics such as BLEU need to set all text in the dataset

Task Type	Dataset Name	Train Samples	Dev Samples	Test Samples
Conditional	DailyDialog	11118	1000	1000
	ROCStories	98,161	1,871	1,871
	Large Movie Review Conditional	21730	17366	4343
Unconditional	COCO Image Captions	10,000	-	10,000
	Large Movie Review Unconditional	22146	17696	4429

Table 1: Dataset statistics of five open-ended generation tasks.

as references. However, since Perception Score scores based on how much the generation fits the task and is not limited to word-surface or semantic level similarity, a few references would be enough for Perception Score to measure a generation. In the experiment, we use four references for each generation and present the average Perception Score. Please refer to appendix for the details about the unconditional generation datasets.

Training Process

Unless specified otherwise, all experiments contain three steps: text generating (unconditional or conditionally, depending on the task), training two separate Perception Models for the two generation models to be compared, comparing P_{sys} of two generation model. We experiment with two versions of Perception Model, RoBERTa$_{\text{base}}$ (12 layers, 768 hidden units, 12 heads) and RoBERTa$_{\text{large}}$ (24 layers, 1024 hidden units, 16 heads). We use batch size 32, learning rate between 1e-5–5e-5 with AdamW optimizer. We run the evaluation on the validation set and store the checkpoint that performs best. The report results are based on the test set.

Human Evaluation Procedure

The performance of an evaluation metric is usually measured by its correlation with the human judgment (Celikyilmaz, Clark, and Gao 2020). Following previous work (Zhang* et al. 2020; Zhao et al. 2019), we use Turing score M1 as the human judgment in our experiments. Turing score M1 is the percentage of a model's generations that are evaluated as better or equal to the references. We compare with various evaluation metrics including: BLEU, perplexity, Comparator Evaluator, BERTScore, MoverScore, BLEURT.

We calculate M1 scores for each generation model on each open-ended generation task. For each sample, 10 Amazon Mechanical Turk (also known as Turkers) with English language proficiency will choose the better ending between one generated ending and one ground truth reference, or they answer "Can not decide".

Results

We introduce the experimental results in this section.

Correlation with Human Judgment

A high-quality automatic metric should have a high correlation with human judgment. We evaluate the correlation of the tested evaluation metrics on generative models.

Conditional Generation Tasks. As shown in Table 2, Perception Score has the highest correlation with human judgment among all metrics across all three conditional generation tasks. The popular n-gram metric BLEU does not even show a moderate correlation with human judgment, which is consistent with the results in previous works (Lowe et al. 2017; Zhang* et al. 2020). We find that BLEU score suffers from a low word-level overlap in these open-ended generation tasks. Besides, the overlap between candidates and references constitutes a considerable ratio of less informative words such as pronouns and be verb. Therefore, the word-level overlap can not guarantee high quality of a generated sentence. Although perplexity has a lower correlation than Perception Score, it still achieves comparable results with other neural model based metrics, especially in DailyDialog. However, since perplexity calculates the probability distribution of the reference sentence, it can not directly reflect the sentence quality under a specific decoding strategy.

Perception Score outperforms other metrics that utilize a pre-trained model by a large margin. We test Perception Score, BERTScore, MoverScore and BLEURT for both of a base size (12 layers) and of a large size (24 layers). Note that Perception Score of a base version outperforms BERTScore and MoverScore of a large version across all three conditional generation tasks.

Unconditional Generation Tasks. Perception Score outperforms all baseline metrics by a large margin. Since there is no corresponding reference for each generation in unconditional generation tasks, we calculate BLEU by utilizing all samples in the dataset as references following previous researchers. However, BLEU only achieves a low correlation with human judgments. Besides, BERTScore, MoverScore and BLEURT are designed for conditional generation tasks and require a fixed reference sentence for each generated sentence, therefore they can not be applied to unconditional generation tasks. Meanwhile, Perception Score can be applied to unconditional generation tasks because it only requires one reference sentence for each generated sentence to calculate the realness.

Besides, we notice that in Large Movie Review Unconditional, no metric produces a moderate correlation with human judgments. Generated reviews in the Large Movie Review Unconditional are usually of longer length and contain between 50–200 words. Furthermore, we found that all metrics have a higher correlation with a review of less text length. However, our method still outperforms all other baseline metrics.

Type	Metrics	DailyDialog	RocStory	LMRC	CIC	LMRU
Metrics that do not utilizes a pre-trained model	BLEU-1	0.102	0.195	0.386	0.504	0.119
	BLEU-2	-0.081	0.491	0.316	0.093	0.091
	BLEU-3	-0.159	0.432	-0.485	0.116	0.152
	BLEU-4	0.107	0.355	-0.390	0.052	0.173
	perplexity	0.496	0.638	0.419	0.494	0.128
Metrics that utilize a pre-trained model	Comparator Evaluator	0.364	0.331	0.387	0.251	0.176
	BERTScore (base)	0.329	0.290	0.454	-	-
	BERTScore (large)	0.371	0.282	0.474	-	-
	MoverScore (base)	0.391	0.313	0.467	-	-
	MoverScore (large)	0.411	0.328	0.487	-	-
	BLEURT (base)	0.415	0.513	0.467	-	-
	BLEURT (large)	0.455	0.529	0.491	-	-
	Perception Score (base)	0.471	0.671	0.488	0.563	0.231
	Perception Score (large)	**0.559**	**0.692**	**0.494**	**0.578**	**0.249**

Table 2: Correlation with human judgment. LMRC is short for Large Movie Review Conditional dataset; CIC is short COCO Image Captions dataset; LMRU is short for Large Movie Review Unconditional dataset. Perception Score outperforms other metrics by a large margin in all four tasks.

Ablation Study

We conduct an ablation study on ROCStories and COCO Image Captions to study the influence of different components. Table 4 presents the results. It shows that both data uncertainty and model uncertainty improve the correlation. Besides, data uncertainty contributes more to the evaluation performance than the model uncertainty. Note Perception Score still outperforms other metrics without using uncertainty to re-weight each generation sample. Besides, we find that incorporating context is necessary when Perception Score is used in conditional generation tasks.

Error Analysis

We describe the errors in DailyDialog dataset and Large Movie Review Unconditional dataset. We notice the performance on Large Movie Review Conditional is not comparable with that on other tasks. We randomly sampled 30 generated movie reviews with a high Perception Score but with low human scores, and 30 generated responses with a low Perception Score but with high human scores. We find these generated reviews have longer text lengths than other generated reviews. We suspect that evaluating longer review would require better understanding ability. In addition, some reviews contain common-sense mistakes but are still scored relative high by Perception Score. We use the same sampling method on DailyDialog and obtaine 60 generated responses. We find 78.3% of them are with longer dialog context. This is consistent with the fact we found on Large Movie Review Unconditional. Besides, the context in these samples usually consists of more turns. This suggests that Perception Score has a better evaluation performance on dialogs with fewer turns.

Robustness Analysis

We test if Perception Score is robust to the variation of the text. We suspect that the model learns the implicit standards of a qualified generation during training. To verify the assumption, we first created four kinds of story endings based on the test set of the ROCStories dataset, and then evaluate these endings with various evaluation metrics. The created endings types are as following: 1) Human-written ending (HWE). Given the story context, Turkers created a reasonable story ending. This is used to test metric performance for a high-quality generation. 2) Lexical different human-written ending (LDHWE). Given the story context and the original story ending, Turkers created a reasonable story ending while avoiding using words from the original one. 3) Human-written ending with better quality (HWEBQ). Given the story context and the story ending reference, Turkers created a story ending that is of better quality than the original reference. To reduce the bias, another group of Turkers will make sure the created endings have better quality than the original ones. 4) Adversarial endings (AE). Given the context and the reference, Turkers modify the original ending as little as possible to create an unreasonable story ending. Figure 4 shows one example of created story endings. We randomly select 100 stories from the test set of ROCStories, and 100 Turkers create these four kinds of endings.

The evaluation results of various metrics are shown in Table 3. We also include the evaluation metrics' scores for the model generated endings (MGE) in the second column. Since BERTScore has a small scale between 0.8–1.0, MoverScore could give a negative score, and BLEURT scores range from a negative number to more than 1.0, we re-scale these three metrics's scores to between 0.0–1.0 for readability.

As is shown in the third column, we find that all the baseline metrics have difficulty in recognizing the quality of the good story endings. Since all other metrics are measuring similarity between sentences in word or semantic level, they fail to fairly evaluate a good story ending that has little similarity with the given reference, which is a common scenario in open-ended generation tasks. Meanwhile, Per-

Metric	MGE	HWE	LDHWE	HWEBQ	AE
BLEU-1	0.1551	0.1348	0.1406	0.1612	0.6816
BLEU-2	0.0468	0.0359	0.0309	0.0514	0.5809
BLEU-3	0.0142	0.0125	0.0104	0.0168	0.4411
BLEU-4	0.0052	0.0054	0.0038	0.0074	0.3014
BERTScore (base)	0.4223	0.3631	0.3742	0.3974	0.7586
MoverScore (base)	0.2439	0.2220	0.2359	0.2510	0.5935
BLEURT (base)	0.4785	0.4847	0.4768	0.5007	0.6458
Perception Score (base)	0.1115	0.4817	0.6001	0.7845	0.3650

Table 3: Robustness analysis results on ROCStories dataset. MGE is short for model generated ending; HWE is short for Human written ending; LDHWE is short for lexical different human written ending; HWEBQ is short for human writing ending with better quality; AE is short for Adversarial endings.

Story Context
Feliciano went olive picking with his grandmother. While they picked, she told him stories of his ancestors. Before he realized it, the sun was going down. They took the olives home and ate them together.

Original Reference Ending
Feliciano was happy about his nice day.

Human Written Ending with Better Quality
Now whenever Feliciano eats olives, he thinks of his grandmother and the stories of his ancestors.

Human Written Ending
Feliciano enjoys the time he spent with his grandmother.

Lexical Different Human Written Ending
Felicoano would never forget that great experience.

Adversarial Ending
Feliciano was upset about his day.

Figure 4: Turker created ending example used in robustness analysis.

	RocStory	CIC
Perception Score (base)	0.671	0.563
w/o model uncertainty	0.653	0.551
w/o data uncertainty	0.636	0.538
w/o context	0.251	-

Table 4: Ablation study of Perception Score. CIC is short for COCO Image Captions dataset. Since COCO Image Captions is an unconditional generation task, it is not applicable for w/o context option. p-value<0.001.

ception Score is around 0.5, correctly showing the created endings have similar quality with the original reference endings. In the LDHWE column, we find the results are similar to that in the HWE column. It shows that whether Turkers are intended to write a different ending than the original reference, the overall wording and semantic of created ending is different than the original one. It matches the assumption that there could be many ground truth references given one context in open-ended natural language generation tasks, and using similarity-based metrics with limited references will inevitably lead to quality misjudgment. In the HWEBQ column, although each baseline metric shows slightly higher scores than MGE, the value difference can not reflect true quality difference. For example, BLEU-1 only increases 0.0061. Meanwhile, our Perception Score better reflects the quality difference. We suspect that the results reflect the great generalizability of Perception Score. In the AE column, all other baselines show a significant boost and give high scores for the unreasonable endings. Meanwhile Perception Score for these endings are much less than 0.5, which shows adversarial endings are much worse than the orignal endings. These results show that Perception Score better captures the overall quality of adversarial endings than other metrics. It also shows taking context into consideration renders better sentence quality judgment.

Discussion

Perception Model is an essential part in our proposed metric. Our results shows Perception Model resolves some of the limitations of the popular automatic evaluation metrics. However, there is no one hyper-parameter configuration of Perception Model that applies to all tasks. It is important for the users to be aware that like all other learned metrics, Perception Model under different configurations and tuning processes will result in different performance. What's more, although all five generation tasks in this paper are based on English, Perception Score is applicable to other languages after replacing the Perception Model to a pre-trained model that matches the target language.

Conclusion

We propose Perception Score, a learned metric for open-ended natural language generation tasks. Perception Score incorporates context information to get a more comprehensive textual representation. Perception Score perceives the realness of the generation against the gold reference through distinguishing a generated sentence from the real sentence. It shows superiority than various metrics on both conditional generation tasks and unconditional generation tasks. Besides, Perception Score is also more robust than popular metrics. For future work, we look forward to extending the Perception Model with zero-shot or few-shot learning.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* .

Cai, P.; Chen, X.; Jin, P.; Wang, H.; and Li, T. 2020. Distributional Discrepancy: A Metric for Unconditional Text Generation.

Celikyilmaz, A.; Clark, E.; and Gao, J. 2020. Evaluation of Text Generation: A Survey. *arXiv preprint arXiv:2006.14799* .

Cer, D.; Yang, Y.; Kong, S.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.; Strope, B.; and Kurzweil, R. 2018. Universal Sentence Encoder. *CoRR* abs/1803.11175. URL http://arxiv.org/abs/1803.11175.

Chaganty, A.; Mussmann, S.; and Liang, P. 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)*, 643–653. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1060. URL https://www.aclweb.org/anthology/P18-1060.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* .

Cui, Y.; Yang, G.; Veit, A.; Huang, X.; and Belongie, S. 2018. Learning to Evaluate Image Captioning. In *CVPR*.

de Masson d'Autume, C.; Mohamed, S.; Rosca, M.; and Rae, J. 2019. Training Language GANs from Scratch. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 4300–4311. Curran Associates, Inc. URL http://papers.nips.cc/paper/8682-training-language-gans-from-scratch.pdf.

DeVries, T.; and Taylor, G. W. 2018. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865* .

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059.

Hashimoto, T.; Zhang, H.; and Liang, P. 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)*, 1689–1701. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/N19-1169. URL https://www.aclweb.org/anthology/N19-1169.

Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; and Niu, S. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lowe, R.; Noseworthy, M.; Serban, I. V.; Angelard-Gontier, N.; Bengio, Y.; and Pineau, J. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1116–1126. Vancouver, Canada: Association for Computational Linguistics. doi:10.18653/v1/P17-1103.

Ma, Q.; Graham, Y.; Wang, S.; and Liu, Q. 2017. Blend: a Novel Combined MT Metric Based on Direct Assessment — CASICT-DCU submission to WMT17 Metrics Task. In *Proceedings of the Second Conference on Machine Translation*, 598–603. Copenhagen, Denmark: Association for Computational Linguistics. doi:10.18653/v1/W17-4768. URL https://www.aclweb.org/anthology/W17-4768.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 142–150. USA: Association for Computational Linguistics. ISBN 9781932432879.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 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*, 839–849. San Diego, California: Association for Computational Linguistics. doi:10.18653/v1/N16-1098. URL https://www.aclweb.org/anthology/N16-1098.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI* .

Sellam, T.; Das, D.; and Parikh, A. P. 2020. BLEURT: Learning Robust Metrics for Text Generation.

Semeniuta, S.; Severyn, A.; and Gelly, S. 2018. On Accurate Evaluation of GANs for Language Generation. *CoRR* abs/1806.04936. URL http://arxiv.org/abs/1806.04936.

Shi, W.; Qian, K.; Wang, X.; and Yu, Z. 2019. How to Build User Simulators to Train RL-based Dialog Systems. *CoRR* abs/1909.01388. URL http://arxiv.org/abs/1909.01388.

Shimanaka, H.; Kajiwara, T.; and Komachi, M. 2018. RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task*

Papers, 751–758. Belgium, Brussels: Association for Computational Linguistics. doi:10.18653/v1/W18-6456. URL https://www.aclweb.org/anthology/W18-6456.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, u.; and Polosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6000–6010. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Zhang*, T.; Kishore*, V.; Wu*, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Zhao, W.; Peyrard, M.; Liu, F.; Gao, Y.; Meyer, C. M.; and Eger, S. 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)*, 563–578. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1053.

Zhou, W.; and Xu, K. 2020. Learning to Compare for Better Training and Evaluation of Open Domain Natural Language Generation Models. In *AAAI*, 9717–9724.