Open-Domain Text Evaluation via Contrastive Distribution Methods

Sidi Lu [1] **Hongyi Liu** [2] **Asli Celikyilmaz** [3] **Tianlu Wang** [3] **Nanyun Peng** [1]

Abstract

Recent advancements in open-domain text generation, driven by the power of large pre-trained language models (LLMs), have demonstrated remarkable performance. However, assessing these models' generation quality remains a challenge. In this paper, we introduce a novel method for evaluating open-domain text generation called Contrastive Distribution Methods (CDM). Leveraging the connection between increasing model parameters and enhanced LLM performance, CDM creates a mapping from the *contrast* of two probabilistic distributions – one known to be superior to the other – to quality measures. We investigate CDM for open-domain text generation evaluation under two paradigms: 1) *Generative* CDM, which harnesses the contrast of two language models' distributions to generate synthetic examples for training discriminator-based metrics; 2) *Discriminative* CDM, which directly uses distribution disparities between two language models for evaluation. Our experiments on coherence evaluation for multi-turn dialogue and commonsense evaluation for controllable generation demonstrate CDM's superior correlate with human judgment than existing automatic evaluation metrics, highlighting the strong performance and generalizability of our approach. [1]

1. Introduction

In recent years, open-domain text generation, fueled by large pretrained generative language models (LLMs), has made significant advancements, garnering substantial attention (Radford et al., 2018; 2019; Brown et al., 2020; OpenAI, 2022; 2023). These systems have showcased remarkable capabilities, such as producing human-like responses, contributing to natural language comprehension, and even performing complex tasks like programming and content generation. With the empirical success, the development of reliable and scalable automatic evaluation metrics for these models become imperative, yet the problem remains an unresolved challenge.

Existing automatic evaluate metrics from pre-LLM eras have their respective limitations. Specifically, reference-based statistical metrics (e.g. BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee & Lavie, 2005)) do not work well for open-ended generation problems with high content diversity like storytelling (Yao et al., 2019) and dialogue systems (Mesgar et al., 2019; Li et al., 2017; Wen et al., 2016), as for these tasks, it is challenging, if not impossible, to collect a sufficiently large number of reference examples to represent the distribution of all feasible outputs. Therefore, prior works have shown their low correlation with human judgments (Liu et al., 2016; Hu et al., 2020). With recent progress in pretrained models, model-based reference metrics like BERTScore (Zhang et al., 2019), Bluert (Sellam et al., 2020) are proposed to facilitate automatic evaluation for text generation. They alleviate the sample efficiency issue of statistical reference-based methods by using pretrained models to compute the similarities between texts based on higher-level semantics. However, the effectiveness of such methods is still reliant on the representativeness of the reference set, and thus falls short when the output semantic space is also highly diverse.

Reference-free evaluation metrics, which assess text directly and provide a quality score, offer a more flexible solution for automatically evaluating open-domain text generation. There are two major paradigms for training models to evaluate texts without references: 1) *Discriminator-based approaches* like ADEM (Lowe et al., 2017) and DEAM (Ghazarian et al., 2022), treat the problem as a prediction task. They train a classifier or regressor to generate a score as the quality assess. However, these methods typically require extensive human annotations or involve dedicated manual designs for generating negative samples to train the classifier. 2) *Distribution/Divergence-based approaches* (Pillutla et al., 2021; Pimentel et al., 2022) focus on obtaining a continuous divergence score between distributions. These approaches have shown promising results in system-level evaluations.

[1]Department of Computer Science, University of California, Los Angeles [2]Shanghai Jiao Tong University [3]Meta FAIR. Correspondence to: Sidi Lu <sidilu@cs.ucla.edu>, Nanyun Peng <violetpeng@cs.ucla.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]Code: https://github.com/PlusLabNLP/CDM

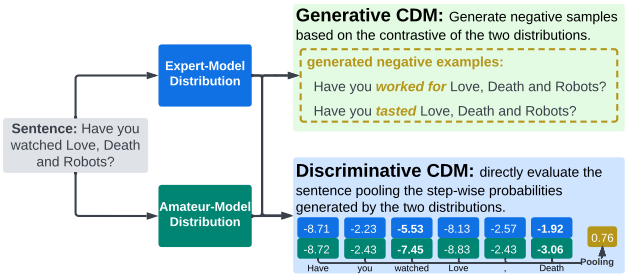


Figure 1. Conceptual illustration of the Contrastive Distribution Methods (CDM). (a) *Generative CDM* generates negative examples for training a discriminator-based metric. (b) *Discriminative CDM* directly evaluate the distribution/sequence by contrasting the step-wise likelihood scores.

However, they often face challenges to accurately assigning credit to individual data points, limiting their ability to perform instance-level evaluations.

In this paper, we propose Contrastive Distribution Methods (CDM), a general and reference-free framework for evaluating open-domain text generation. CDM operates on an intuitive yet broadly applicable premise: models with similar architectures but varying sizes generally exhibit improved performance as model size increases. Consequently, CDM is designed to capture the dynamics of model performance as it scales with the increasing number of parameters. Utilizing such dynamics, CDM *contrasts* two language models' distributions and conduct inference in both *generative* and *discriminative* manners to create automatic evaluation metrics. Specifically, Generative CDM as illustrated in the upper right corner of Figure 1 produces effective negative samples to facilitate the learning of discriminator-based evaluation metrics without the requirement of additional human annotations or sophisticated design for the data generation process, and Discriminative CDM as illustrated in the lower right corner of Figure 1 provides a distribution-level measurement of quality for each instance, and thus results in reliable distribution-based metrics without compromising instance-level evaluation performance.

Experiments on open-domain dialogue evaluation and commonsense keywords-to-text evaluation demonstrate strong performance of CDMs, consistently outperforming strong baselines such as G-Eval (Liu et al., 2023a) in terms of correlation with human judgements across datasets.

2. Background and Related Works

Open-Domain Text Evaluation There has been a synchronously growing interest in developing robust evaluation methods for open-domain text generation models. Traditional evaluation metrics, such as BLEU and ROUGE, have been shown to be inadequate for assessing the quality of complex, multi-sentence responses generated by these models. As a result, researchers have explored alternative evaluation methods, including human evaluation, adversarial evaluation, and unsupervised metrics. Human evaluation remains the gold standard, but it is time-consuming and costly. Adversarial evaluation, which involves testing models against a set of challenging examples, has shown promise in identifying weaknesses in current models. Unsupervised metrics, such as BERTScore and Perplexity, provide quick and automated evaluation, but their correlation with human judgments remains a topic of debate. The field of open-domain text evaluation continues to evolve, and developing reliable evaluation methods will be essential for advancing the state-of-the-art in this exciting area of research.

Discriminator-based Metrics ADEM (Lowe et al., 2017) is one of the first attempts at training a model to evaluate machine-generated text. It deals with single-turn dialogue evaluation problem, and uses the contextualized representation of the context in interaction with that of the responses to train the model. DEAM (Ghazarian et al., 2022) and AMRFact (Qiu et al., 2023) are novel evaluation metrics that aim to assess open-end generation models with structured manipulations to create negative samples from positive ones, allowing for a more nuanced assessment of model performance like coherence (for dialogue system) or factuality (for summarization models). Typically, they operate by first parsing the sequence into an abstract meaning representation (AMR), and then manipulating the AMR to introduce inconsistencies and irrelevancies that undermine the coherence of the dialogue. The manipulated AMR is then transformed back into text form for evaluation. This method supports multi-turn dialogue evaluation and has achieved state-of-the-art performance on various benchmark datasets. By using AMR-based semantic manipulations, these methods provide a class of promising approaches for performing automatic evaluation in a more comprehensive and accurate manner. *Generative* CDM shares a similar process , as it manipulates the positive true samples for the generation of negative samples, serving the purpose of training a classifier.

Distribution/Divergence-based Metrics MAUVE and follow-up works (Pillutla et al., 2021; Pimentel et al., 2022) analyse the quality gap between human-generated text and machine-generated text by studying the divergence frontier of human-generated samples in contrast to the learnt model. While their setup is not directly relevant to our approach, it provides an insightful perspective of using the likelihood predictions of LMs for evaluation purposes. Zhong et al. (2022a) proposes a multi-dimensional evaluation system for more robust automatic evaluation. It ensembles the score from a set of *discriminator-based* metrics, each of which trained to evaluate a specific aspect in intuition of the text quality. GPTEval (Liu et al., 2023b) tries to quantitatively exploit large language models that are trained with strong human alignment. It uses the score prediction from GPT-4 (OpenAI, 2023) to evaluate how well the given text adheres

to human opinion. *Discriminative* CDM falls under this paradigm, since it serves as a metric with more continuously distributed scores for the evaluated text.

Contrastive Decoding, Contrastive Momentum and ExPO Contrastive decoding is a decoding algorithm that leverages the strengths of two language models: a stronger expert model and a weaker amateur model. The algorithm decodes towards the objective of maximizing the difference between the log-probabilities of the expert and amateur models, resulting in high-quality generated samples. Specifically, the algorithm tries to decode sequences that maximize the *contrastive momentum*:

$$\log p_{\mathrm{e}}(x) - \log p_{\mathrm{a}}(x), \tag{1}$$

where p_{e} and p_{a} represent the expert and the amateur models, respectively, and x is the generated sample. The original paper (Li et al., 2022) demonstrates that this approach results in higher quality samples than decoding from the expert model alone. Contrastive decoding provides an insightful way to study the dynamics of how models' capabilities scale up with larger parameter numbers. The proposed CDM is highly inspired by the Contrastive decoding method, yet leveraging it for evaluation purposes.

Noticeably, a recent preference optimization method called ExPO (Zheng et al., 2024) also shares a similar idea. ExPO significantly improves the instruction following abilities of (open-sourced) large language models without the necessity of performing any costly training and with even less data and/or trial sampling from the LLMs. It creates the extrapolation of human-aligned models (in our notion, the *expert*) in contrast to its primitive version after only the supervised finetuning (SFT) stage (in our notion, the *amateur*) on instruction-following data. The biggest difference between ExPO and contrastive decoding or this paper is, since the *amateur* and *expert* models in ExPO share the same parameter space and is assumed to be very close to each other, ExPO directly performs the extrapolation in the parameter space, instead of the log-probability space as in ours or the contrastive decoding algorithm.

3. Methodology

3.1. Notations and Problem Formulation

We use $\mathbf{s}$ to denote a sequence and s_i to denote the $i-$th token in $\mathbf{s}$. $p(\mathbf{s})$ denotes the probability of sequence $\mathbf{s}$ under a model p. We assume model p is a probabilistic distribution defined on Σ^*, where Σ is the set of valid tokens and Σ^* is the universal set of all sequences consisting of such tokens.

Consider an imaginary distribution-level oracle metric $E(p)$ which projects from a model distribution $p(\mathbf{s})$ to "a measure of model performance" – a scalar. This function does not necessarily have an analytical form, however, we assume

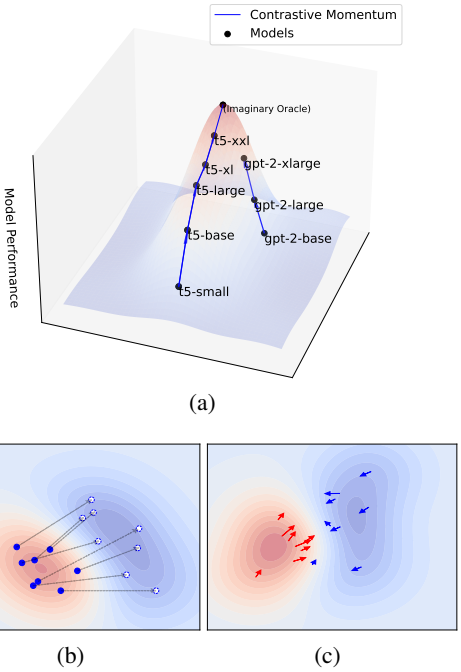


Figure 2. (a) While it is hard to assume a total order for models from different model classes under the oracle metric $E(p)$, it is plausible to assume partial orders for models from the same model class. (b) Generative CDM uses the degraded distribution p_n to synthesize fake samples for training a discriminator as the metric. The warm/cold region indicates the decision boundary of the resulting trainable metric induced by fake samples from p_n. (c) Discriminative CDM directly determines the decision boundary by pooling the values of the step-wise contrastive momentum.

that we have access to some partial order relations it defines. Intuitively, this imaginary oracle $E(p)$ should correlate perfectly with human judgements of the model performance, and any evaluation metric that correlates better with human judgments is a better approximation of $E(p)$.

With the notion of oracle $E(p)$, we can perform:

- *Discriminative* inference:
  a) **Distribution-level evaluation** to evaluate any existing models by ranking them according to $E(p)$
  b) **Sample-level evaluation** to use $\frac{\partial E(p)}{\partial p(\mathbf{s})}$ to reflect the *quality* of $\mathbf{s}$. Because given the evaluated sequence $\mathbf{s}$, $\frac{\partial E(p)}{\partial p(\mathbf{s})}$ represents whether and how much altering the model p towards higher $p(\mathbf{s})$ would improve $E(p)$.
- *Generative* inference: to improve or degenerate the generation quality by altering p towards better or worse of $E(p)$. The altered distribution produces more obfuscating fake examples, which can then be used to train discriminator-based sample-level evaluation metrics.

In the following, we will explain discriminative and gen-

erative inference of CDM for automatically evaluation of open-domain generation in more details.

3.2. The Partial Order Assumption

While it is nontrivial to come up with analytical forms for $E(p)$, we can make some assumptions to obtain partial orders from $E(p)$. Consider a *series* of models that share similar architectures and other pretraining/finetuning setups, but differ in model sizes (e.g. T5-small/base/large, etc.). It is usually safe to assume that the model with a larger number of parameters perform better than the smaller one under most aspects. More formally, we can assume a partial order (a linear order within one concerned model class) induced by the oracle metric $E(p)$ as illustrated in Equation 2 and Figure 2(a):

$$E(p_{small}) < E(p_{base}) < E(p_{large}) \tag{2}$$

Limitation Note that, while the partial order assumption is usually true for most existing model families in empirical practices, we are *open* to the possibility that it *might not hold* in some cases. As a result, the effectiveness of the proposed approach is inherently limited to cases where the partial order assumption holds.

3.3. First Order Approximation of $E(p)$

Since we do not assume the knowledge about the analytical form of $E(p)$, it is intractable to compute $\frac{\partial E(p)}{\partial p(\mathbf{s})}$. However, following similar approach as in (Li et al., 2022), we can approximate $E(p)$ using a secant hyperplane between two distributions in the range of $E(p)$, i.e., the *amateur* distribution p_a and the *expert* distribution p_e. In other words, we approximate $E(p)$ use the following analytic form:

$$E(p) = \sum_{\mathbf{s}} \Big(\log p_e(\mathbf{s}) - \log p_a(\mathbf{s})\Big) p(\mathbf{s}), \tag{3}$$

It's trivial to prove that this approximation ensures $E(p_e) > E(p_a)$. We can further define the *contrastive momentum* $m(\mathbf{s}) = \log p_e(\mathbf{s}) - \log p_a(\mathbf{s})$. Different choices of p_a and p_e result in different *contrastive momentum* and thus distinct quality of the first-order approximations for $E(p)$, hence different performance of the evaluation metric. We investigate the general principle for choosing the expert and amateur distributions in the experiment section.

3.4. Contrastive Distribution Methods

3.4.1. Generative CDM

Generative CDM focuses on synthetic data generation using contrastive distributions. We follow prior works such as ADEM (Lowe et al., 2017) and DEAM (Ghazarian et al., 2022) to formulate reference-free evaluation metrics as prediction tasks. In order to evaluate generated texts, a discriminator can be trained on positive and negative examples to serve as the evaluation metric.[2] However, while we can assume human-written texts are positive examples, negative examples are non-trivial to obtain. Randomly generating negative examples using a uniform distribution over all possible sequences of tokens is not efficient, as most negative examples generated this way would be too trivial. On the other hand, generating negative examples by masking out spams in positive examples and having pretrained large-language models to fill in the masks may not result in real low-quality texts, which would confuse the discriminator.

To this end, generative CDM provides a controllable approach to *reduce* the quality of pretrained language models to generate "deceptive negative examples". Specifically, it generates from a "novice" distribution p_n that descends along the direction of $-\frac{\partial E(p)}{\partial \log p}$ from the amateur model p_a — a weaker distribution than the amateur model. Applying the approximation in Equation 3, we follow the reversed direction of the contrastive momentum $m = \log p_e - \log p_a$ to degenerate from the *amateur* model p_a. Mathematically, we obtain a probability distribution $\log p_n \propto \log p_a - \gamma m$ that *amplifies* the likelihood of "machine artifacts" in a controllable (by setting the hyper-parameter γ) scale. Sampling from p_n allows us to obtain suitable negative examples.

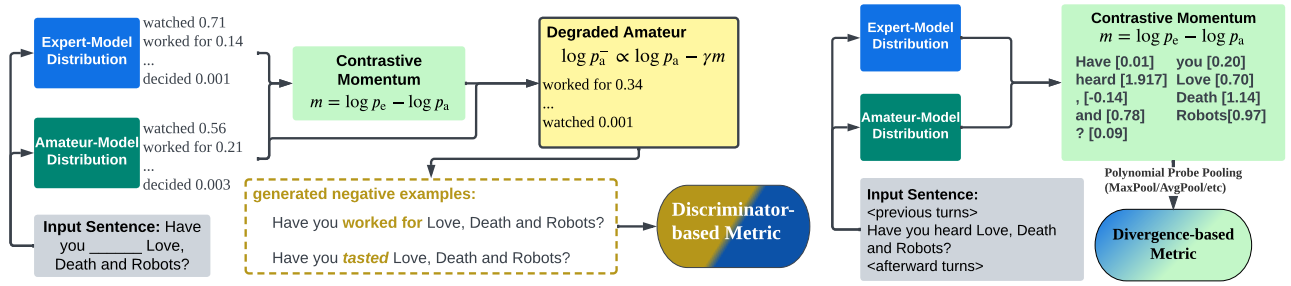
Implementation Details. We hereby discuss how to generate targeted negative examples. We start from existing positive examples $\mathbf{s}$ and construct the negative samples by masking out certain part $\mathbf{s}^{M+}$ of the positive ones, then conduct conditional generation using the remaining part $\mathbf{s} \setminus \mathbf{s}^{M+}$ as the initial context. As a result, the generated negative examples would be more disguising compared to sampling directly from p_n. To achieve this, we train a *segment infilling* model. Given a positive example and the position at which a segment is removed (randomly or strategically), we model the conditional distribution that reconstructs the original segment. We train an expert and an amateur model with segment infilling capabilities, we can then compose the distribution for sampling in the following form:

$$\begin{aligned} \log p_{edit}(\mathbf{s}^M|\mathbf{s} \setminus \mathbf{s}^{M+}) &\propto \log p_a(\mathbf{s}^M|\mathbf{s} \setminus \mathbf{s}^{M+}) \\ &\quad -\gamma m(\mathbf{s}^M|\mathbf{s} \setminus \mathbf{s}^{M+}), \\ \text{where } \ m(\mathbf{s}^M|\mathbf{s} \setminus \mathbf{s}^{M+}) &= \log p_e(\mathbf{s}^M|\mathbf{s} \setminus \mathbf{s}^{M+}) \\ &\quad - \log p_a(\mathbf{s}^M|\mathbf{s} \setminus \mathbf{s}^{M+}) \end{aligned} \tag{4}$$

This enables us to flexibly generate targeted negative examples that are deceptive. Figure 3(a) and 2(b) illustrates this process in the procedural and distributional views.

The full process of generative CDM can be summarized as

[2]The discriminator does not necessarily need to provide binary decisions, it can also produce scores. But we use binary examples for simplicity.



(a) Generative CDM (b) Discriminative CDM

Figure 3. A more detailed illustration of the two Contrastive Distribution Methods (CDM). (a) *Generative CDM* constructs fake negative samples from positive ones for training a discriminator-based metric. (b) *Discriminative CDM* directly evaluate the distribution/sequence by contrasting and aggregating the step-wise likelihood scores.

follows:

Algorithm 1 Generative CDM

1: Train the amateur model p_a to solve the segment insertion problem
2: Train the expert model p_e to solve the segment insertion problem
3: Construct the contrastive momentum $m_{a \to e} = \log p_e - \log p_a$
4: Construct the degraded distribution $\log p_a^- \propto \log p_a - \gamma m_{a \to e}$
5: negativeSamples = {}
6: **for** positiveSample $\mathbf{s}^+$ **in** positiveSamples **do**
7: Remove a segment $\mathbf{e}^+ \subset \mathbf{s}^+$ from $\mathbf{s}^+$ to construct the context $\mathbf{c} = \mathbf{s}^+ - \mathbf{e}^+$
8: Regenerate a segment $\mathbf{e}^-$ in the same position using $p_a^-(\mathbf{e}|\mathbf{c})$
9: Obtain the reconstructed negative sample $\mathbf{s}^- = \mathbf{c} \cup \mathbf{e}^-$
10: Add $\mathbf{s}^-$ to negativeSamples
11: **end for**
12: Train the metric model D as a discriminator with {negativeSamples, positiveSamples}
13: **return** metric D

3.4.2. Discriminative CDM

Although Generative CDM is a reasonably flexible and scalable framework, there are many variable factors in the generation process (e.g. how to choose which segment to remove, the degradation strength factor γ etc.) that may affect the performance of the resulting data and thus the evaluation metrics. Therefore, we propose an alternative paradigm under the CDM framework to remove the generation subroutine completely. In generative CDM, after data generation, we train a discriminator to distinguish positive and negative examples as the evaluation metrics. Effectively, we are learning the boundary between positive and negative samples, because we usually do not have a tractable model for the positive or negative distribution. However, under the CDM framework, we do have a tractable model for the negative distribution, which is composed from the amateur model p_a and the expert model p_e. In light of this, we can consider directly deploying m as a divergence-based metric for evaluation.

For each sequence, we collect the step-wise contrastive momentum $m(x_t|\mathbf{s}_{<t}) = \log p_e(x|\mathbf{s}_{<t}) - \log p_a(x|\mathbf{s}_{<t})$ composed from the *amateur* model and the *expert* model. For a good data sample, both models' likelihood prediction will be relatively high while the expert model will assign significantly higher probability to the sample, thus $\sum_t m(x|\mathbf{s}_{<t})$ should be significantly larger than 0.

We can directly sum up the step-wise contrastive momentum over the entire sequence (*i.e.* sum-pooling) to be the metrics for generation quality evaluation. See Figure 3(b) and 2(c). However, the sum-pooled score would be numerically influenced by the sequence length. Moreover, sum-pooling overemphasizes the impact of extremely low probability steps because the discrepancy between the amateur model and expert model predictions in low-probability regions could be significantly amplified on the logarithmic scale. Therefore, in the experiments, we compare different strategies to *pool* the sequence of step-wise contrastive momentum values into a sentence-level evaluation score. We call this paradigm of using CDM as *Discriminative CDM*, as we directly operate on two models' contrastive momentum on data samples as their quality evaluation metrics.

4. Experiments

To support our claim that CDM is an evaluation metric that works generally for open-domain text generation tasks, we hereby conduct experiments under two distinct scenarios: 1) using CDM to evaluate dialogue systems, representing the tasks with a diverse distribution of outputs; 2) using CDM to evaluate the commonsense of lexically-constrained generation models, representing tasks with medium to low

diversity in the output.

4.1. Dialogue Evaluation

The first part of our experiment is primarily focused on dialogue evaluation. Given a set of annotated dialogues, each with human-annotated quality scores ranging from 0.0 to 1.0, our objective is to assign scores to each evaluated sequence that maximizes the correlation with human annotations. Additionally, for dialogue evaluation, we assume we are not permitted to perform any training on data within the same domain. Our training/fine-tuning exercises are conducted on a subset of dialogues from both TopicalChat (Gopalakrishnan et al., 2019) and PersonaChat datasets (Zhang et al., 2018), following the setup in Ghazarian et al. (2022). We evaluate our methods on annotated dialogues from the FED (Mehri & Eskenazi, 2020) and DSTC9 (Gunasekara et al., 2020) datasets.

In our experiment results, we report spearman correlation of different approaches for dialogue evaluation. All reported correlation coefficients from our approaches have p-value (with Bonferroni correction) < 0.01.

Dataset and Experiment Setup We adopt most experimental settings from DEAM (Ghazarian et al., 2022) to verify the effectiveness of our method. The statistics of the involved datasets in our experiments are shown as follows:

Table 1. Data usage in the dialogue evaluation experiment.

Dataset	**size**	**Avg. len**
TopicalChat (Gopalakrishnan et al., 2019)	17567	377
+ PersonaChat (Zhang et al., 2018)	+2078	
FED (test) (Mehri & Eskenazi, 2020)	125	168
DSTC9 (test) (Gunasekara et al., 2020)	2200	318

4.1.1. Model Specification

For Generative CDM, there are multiple strategies to manipulate positive examples to generate negative examples. We study the following strategies:

- Segment/Utterance-Single: The manipulation of data is only applied once to a random *segment no longer than 20 tokens* or *a random utterance* in a real dialogue.
- Mixed-Single/Multi: The manipulation of data is applied to a random utterance or a random segment no longer than 20 tokens in a real dialogue, for once or a uniformly random value from 1 to 4 times.
- AMR-Multi: The location of data manipulation is guided by similar approach as in DEAM (Ghazarian et al., 2022).

Similarly, for discriminative CDM, there are many different aggregation strategies to pool step-wise contrastive momentums, and we study following:

Table 2. Main results of CDMs for dialogue evaluation in comparison to a few baselines. For G-Eval, to keep the comparison fair, we report results by instructing the largest models involved in CDMs. We include the original G-Eval metric result in our appendix. For UniEval, it is not capable of performing conversation-level evaluation, we report the average-pooled results of all utterance-level scores. Spearman correlation with human is reported. We highlight the best-performing results with **bolded** numbers and second-best with underlined numbers. The models are selected using the validation set of Topical-Personal chat.

Model	FED		DSTC9	
	Coherence	Overall	Coherence	Overall
Mesgar et al. (2019)	0.10	-0.01	0.02	0.05
Vakulenko et al. (2018)	0.13	0.10	0.00	0.00
DynaEval (Zhang et al., 2021)	-0.36	-0.4	-0.03	-0.01
DEAM (Ghazarian et al., 2022)	0.47	0.55	0.19	0.20
G-Eval (Liu et al., 2023a)				
- (w/ LLaMa2-7b-Vicuna)	0.57	0.54	0.15	0.14
- (w/ Flan-T5-11b)	0.49	0.48	0.19	0.18
UniEval (Zhong et al., 2022b)	0.33	0.35	0.14	0.13
Generative CDM (Ours)	0.53	0.55	0.22	0.24
Discriminative CDM (Ours)	**0.59**	**0.62**	**0.28**	**0.27**

- Classifier-Pooled: We train a small linear classifier to convert the sequence of contrastive momentum scores as a trainable pooler using annotated training data (from the original dataset or as synthesized by DEAM (Ghazarian et al., 2022)). Intuitively, this is to align the aspect-agnostic *contrastive momentum* with the concerned specific metric.
- Trivial pooling along the timestep axis (Avg-Pooled/Max-Pooled/Min-Pooled)

4.1.2. Baselines

Following the setup in Ghazarian et al. (2022), we compare against existing methods on negative sampling for trainable metrics, including Mesgar et al. (2019), Vakulenko et al. (2018), DynaEval(Zhang et al., 2021) and DEAM(Ghazarian et al., 2022). We also report our comparison with some more advanced automatic evaluation metrics using pretrained large language models, including UniEval(Zhong et al., 2022b) and GEval(Liu et al., 2023a). For a fair comparison, we report GEval using instruction-tuned models no larger than the best *expert* model involved in our CDM results.

4.1.3. Result Discussion and Detailed Ablation

Our approach aligns with methodology established by previous research and report the Spearman correlation to better evaluate CDM against these baselines. All hyperparameters for training the likelihood functions *amateur* and *expert* are determined with the best log-likelihood on the in-domain validation set from Topical-PersonaChat dataset.

Table 3. Ablation studies focusing on manipulation strategies in generative CDM and pooling strategies in discriminative CDM with fixed armature model of T5-small and expert model of T5-large. Spearman correlation with humans is reported. All correlation coefficients from our approaches have p-value (with Bonferroni correction) < 0.01.

Model	FED		DSTC9	
	Coherence	Overall	Coherence	Overall
Generative CDM (small-large)				
- Segment-Single	0.12	0.07	0.11	0.10
- Utterance-Single	0.29	0.36	0.05	0.08
- Mixed-Single	0.32	0.35	0.14	0.12
- Mixed-Multi	0.42	0.40	0.17	0.18
- AMR-Multi	0.49	0.53	0.20	0.22
Discriminative CDM (small-large)				
- Avg-Pooled	0.31	0.32	0.12	0.13
- Min-Pooled	0.27	0.28	0.07	0.04
- Max-Pooled	0.46	0.43	0.16	0.15
- Classifier-Pooled	0.53	0.56	0.24	0.22

Main Results We report the main results with the two versions of CDM built from T5 models (Raffel et al., 2019; Wei et al., 2021) checkpoints respectively in Table 2. Discriminative CDM methods present less bias across datasets and offer more efficiency during training, as they eliminate the necessity for collecting negative samples and training an additional deep-NN-based classifier model. In general, Generative CDM performs the best among all trainable metrics using an explicit negative sampling process, while Discriminative CDM performs generally even better and achieve the state-of-the-art in all evaluated metrics.

Ablation study: Negative sampling and pooling strategies With the both versions of CDM being a composition of different components, we study how different components in our negative sampling strategy (for Generative CDM) and pooling (for Discriminative CDM) contribute to the performance of CDM metrics. See Table 3. The performance of the metric obtained from Generative CDM is greatly impacted towards the negative sampling strategies. Manipulating the real samples in both utterance-level and segment-level for multiple times overall produces the best-quality negative samples. In addition, combining the idea in DEAM(Ghazarian et al., 2022), we observe that further using AMR to perform the negative sample in a guided fashion is helpful, sufficiently making Generative CDM superior among all negative sampling-based metrics.

Ablation study: Impact of contrastive model sizes We study how different model sizes in CDMs impact the performance of resulting metrics. See Table 4. Our findings indicate that larger performance gap between the amateur/-expert models in general induces better performance.

Ablation study: Generative CDM versus simply re-sampling from infilling models We also conduct experiments of directly synthesizing the negative samples by directly sampling from different sizes of amateur model only. We observe inverse scaling (i.e. sampling from better amateur model induces worse Generative CDM metric) in such attempts. This shows that contrasting the amateur model against the expert model is necessary for generating high-quality negative samples. See Table 5.

Table 4. Ablation studies focusing on varying amateur and expert model sizes in CDMs with the best manipulation strategy for generative CDM and the best pooling strategy for discriminative CDM. In addition to the T5 family, we report results of CDM using different sizes of LLaMa 2 as the amateur and expert models. Spearman correlation with humans is reported. All correlation coefficients have p-value (with Bonferroni correction) < 0.01.

Model	FED		DSTC9	
	Coherence	Overall	Coherence	Overall
Generative CDM				
- T5 small-base	0.48	0.51	0.19	0.20
- T5 small-large	0.49	0.53	0.20	0.22
- T5 small-xl	0.51	0.52	0.19	0.23
- T5 small-11b	0.53	0.55	0.22	0.24
- T5 base-large	0.29	0.31	0.08	0.09
- T5 base-xl	0.30	0.32	0.09	0.09
- T5 base-11b	0.31	0.32	0.09	0.10
Discriminative CDM				
- T5 small-base	0.42	0.44	0.13	0.10
- T5 small-large	0.53	0.56	0.24	0.22
- T5 small-xl	0.59	0.61	0.27	0.25
- T5 small-11b	0.59	0.62	0.28	0.27
- T5 base-large	0.39	0.40	0.09	0.11
- T5 base-xl	0.47	0.46	0.12	0.13
- T5 base-11b	0.52	0.51	0.15	0.16
Finetuned LLaMa 2 7B-13B, No Infilling				
- Generative CDM	0.18	0.21	0.06	0.08
- Discriminative CDM	0.20	0.22	0.13	0.15

Ablation Study: CDM with decoder-only models In Table 6 we show results of CDM composed from decoder-only models. We report results with Pythia models in two different setups: 1) we finetine Pythia models in a regular paradigm 2) we finetune Pythia models with infilling capabilities using the fill-in-the-middle objective (Bavarian et al., 2022). In addition, we also report results with stronger decoder-only models like LLaMa 2 (Touvron et al., 2023). In general, infilling capabilities provide better flexibility in negative sampling as well as likelihood estimation with *bi-directional* contexts, leading to better CDM metrics.

4.2. Commmonsense Evaluation

In addition to the previous dialogue evaluation task, we consider a different setup where we use CDM to evaluate the commonsense of generated outputs in controllable generation problems. Generative commonsense reasoning has

Table 5. Comparisons between generative CDM and sampling directly from the amateur models. Spearman correlation with humans is reported. All correlation coefficients have p-value (with Bonferroni correction) < 0.01.

Model	FED		DSTC9	
	Coherence	Overall	Coherence	Overall
Generative CDM				
- T5 small-base	0.48	0.51	0.19	0.20
- T5 small-large	0.49	0.53	0.20	0.22
- T5 small-xl	0.51	0.52	0.19	0.23
- T5 small-11b	0.53	0.55	0.22	0.24
Resampling from amateur models				
- T5-small	0.31	0.28	0.09	0.08
- T5-base	0.19	0.16	0.05	0.04
- T5-large	0.09	0.05	-0.01	0.02

long been an interesting and challenging task, especially for models with smaller parameter numbers. We conduct this experiment not only to show that CDM can serve as a strong evaluation metric for commonsense evaluation, but also to demonstrate the generalizability of CDM.

4.2.1. Dataset Preparation

CommonGen (Lin et al., 2020) is a generative commonsense reasoning dataset that examines language models' capability of capturing commonsense and human logic. In CommonGen, the model is given a set of *concept* keywords, and is expected to produce a descriptive sentence that: 1) contains all concept keywords (with necessary inflections for grammaticality); 2) compliant to commonsense.

In the training split, CommonGen contains 32,650 unique concept sets, each with 1-3 annotated description. It also provides a validation set consisting of 992 unique concept sets, each with 4-5 annotated reference description. In test phase, there are 1496 compositionally varied concept sets that are intentionally made to be *NOT i.i.d.* to the training set. The golden annotations for these test input are not provided publicly.

Evaluating Commonsense Metrics with CommonGen-trinity We reorganize and further annotate the test split of the dataset into a new dataset called *CommonGen-trinity* to evaluate commonsense metrics. For each concept set in the test split, we use GPT-4 (OpenAI, 2023) to annotate 6 distinct descriptions containing every one of the keywords or its inflection. Furthermore, we prompt and control the large language model to produce samples with diverse degrees of commonsense: we generate 2 of the 6 annotations as fully commonsensical sentences; we control other 2 of the annotations to be of medium violation of commonsense, and 2 of the annotations to be completely non-commonsensical. See Table 7. We report the prompt (adapted from the ones in CommonGen-Lite(Lin et al., 2020) for such fine-grained auto-annotation in our appendix.

Table 6. Comparisons between different manipulation capabilities in generative CDM. To achieve infilling with Pythia (an autoregressive model), we follow the fill-in-the-middle objective (Bavarian et al., 2022). Spearman correlation with humans is reported. All correlation coefficients have p-value (with Bonferroni correction) < 0.01.

Model	FED		DSTC9	
	Coherence	Overall	Coherence	Overall
Manipulation using text infilling				
- T5 small-base	0.48	0.51	0.19	0.20
- T5 small-large	0.49	0.53	0.20	0.22
- T5 small-xl	0.51	0.52	0.19	0.23
- T5 small-11b	0.53	0.55	0.22	0.24
- Pythia 70M-160M	0.28	0.31	0.04	0.06
- Pythia 70M-410M	0.32	0.34	0.09	0.10
- Pythia 70M-1.0B	0.35	0.37	0.10	0.11
- Pythia 70M-1.4B	0.35	0.38	0.10	0.12
- Pythia 70M-6.9B	0.44	0.41	0.18	0.22
Manipulation using autoregressive generation				
- Pythia 70M-160M	0.26	0.27	0.02	0.04
- Pythia 70M-410M	0.31	0.29	0.04	0.04
- Pythia 70M-1.0B	0.36	0.37	0.06	0.05
- Pythia 70M-1.4B	0.36	0.38	0.07	0.07
- Pythia 70M-6.9B	0.41	0.43	0.09	0.08

During evaluation, we always annotate the highly commonsensical descriptions with a score of 1.0. We report the results under two setups for the annotation of other test descriptions:

- **CommonGen-trinity-*binarized***: In this case, both *mediocre* and *non-commonsensical* descriptions are treated as non-commonsensical samples, and annotated with a score of 0.0.
- **CommonGen-trinity-*raw***: In this case we annotate *mediocre* samples with a score of 0.5 and *non-commmonsensical* samples with a score of 0.0.

Similar to our setup in Section 4.1, for each baseline or model variant, we assign a score to each test description, and report the Spearman correlation to the golden scores of each sample.

4.3. Results and Analysis

We compare against G-Eval, the commonsense oracle built in BOOST (Tian et al., 2023) from COMET (Bosselut et al., 2019) and ACCENT (Ghazarian et al., 2023). Following the setup in previous experiments, for a fair comparison, we report the G-Eval scores using instruction-tuned models no larger than the largest *expert* model involved in CDMs.

Discussion In general, CDM achieve the state-of-the-art

Table 7. Example of the proposed CommonGen-trinity dataset.

Concept Set	sidewalk dog walk leash
Commonsensical	A woman walks her dog on the sidewalk, holding tightly to the leash
Mediocre	A dog walks its owner on a leash along the sidewalk.
Non-Commonsensical	A sidewalk walks a dog with a leash on a dog.

Table 8. Main results for evaluating the commonsense metrics using CommonGen-trinity dataset. Spearman correlation with the golden labels is reported. We highlight the best-performing results with **bolded** numbers and second-best with underlined numbers.

Model	raw	binarized
G-Eval (Liu et al., 2023a)		
- (w/ LLaMa2-7b-Vicuna)	0.38	0.33
- (w/ Flan-T5-11b)	0.61	0.53
BOOST Commonsense Oracle (Tian et al., 2023)		
- (w/ *mean* aggregation)	0.10	0.07
- (w/ *total* aggregation)	0.01	-0.00
ACCENT (Ghazarian et al., 2023)	-0.08	-0.07
Generative CDM (Ours, small-11b)	0.63	0.55
Discriminative CDM (Ours, small-11b)	**0.73**	**0.61**

among the evaluated metrics using a pretrained model with a maximal size of 11b. G-Eval using the LLaMa2-7b model falls short for this task compared to results obtained from Flan-T5-11b, yielding that LLaMa2 may not be a better world model than Flan-T5. For BOOST Oracle, its detection of entity relation is limited to types in **UsedFor/AtLocation/CapableOf/PartOf**. Thus, while it is successful in guiding a generator, our results argue that it might not be the most competitive commonsense evaluation metric. For ACCENT, it is designed and trained for dialogue commonsense evaluation instead of single-sentence evaluation. To mitigate this, we report the results by composing a prompt containing problem descriptions as the virtual conversation history and evaluating the scene description as the next utterance. With these compromises, it is possible that the results for ACCENT may not reflect its true performance for the original use case.

5. Conclusion

This paper presents the Contrastive Distribution Methods (CDM) as a general framework for evaluating open-domain text generation models. CDM is constructed around analyzing the correlation between model scales and the respective distribution prediction, and how it can be exploited to alter the performance of a certain model on-the-fly in inference. We demonstrate how CDM can be used for evaluation purposes in two general paradigms: Generative CDM, which manipulates existing positive samples to generate in-domain negative samples and subsequently trains a classifier, and Discriminative CDM, which employs the contrastive momentum as a direct metric for evaluation. Our experiments results in multi-turn dialogue evaluation and commonsense evaluation for controllable generation illustrate that CDM correlates better with human intuition than traditional metrics. In summary, the CDM method emerges as a promising and scalable approach for evaluating open-domain text generation systems, among others.

For future work, it is interesting to consider the contrastive momentum concerning more than two distributions as a reflection of an extended series of models across different scales.

Limitation

We hereby list a few potential limitations of the proposed method:

- While the method is proposed as a very general framework, training a metric with CDM is still a highly task-dependent practice. With our paper providing evidences for several principled practices (enlarging the discrepancies between models, etc), it could still need some efforts to apply CDM to a new task domain.
- It is theoretically infeasible for CDM to produce metrics for evaluating the inverse-scaling tasks (*i.e.* the **bigger** the base model, the **worse** the performance), as it goes against the very basic assumption the CDM approaches rely on.
- Currently presented results of CDM are based on the linear, first order approximation of the oracle $E(p)$ using a secant hyperplane. This approach can be highly limited compared to a more accurate approximation. We leave this for future work.

Acknowledgement

This research is supported by Meta Sponsor Research Award and Okawa Foundation Research Grant. We would also like to thank Sarik Ghazarian, Honghua Zhang, Meihua Dang and many other of UCLA/USC-PlusLab members for their kind writing suggestions and more. We also would like to express our deep appreciation to many of our anonymous reviewers for their insightful comments.

Impact Statement

The goal of this paper is to quantitatively study how the scaling-law of large language models correlate with human preferences without an explicit human-preference alignment process of any form. It might shed a light on a deeper understanding of the *super-alignment* of language models, which means it could be possible that large language models achieve high agreement rate with human by simply using the feedback of language models with lesser capabilities.

References

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

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*, 2022.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Sarik Ghazarian, Nuan Wen, Aram Galstyan, and Nanyun Peng. Deam: Dialogue coherence evaluation using amr-based semantic manipulations. *arXiv preprint arXiv:2203.09711*, 2022.

Sarik Ghazarian, Yijia Shao, Rujun Han, Aram Galstyan, and Nanyun Peng. Accent: An automatic event commonsense evaluation metric for open-domain dialogue systems. *arXiv preprint arXiv:2305.07797*, 2023.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pp. 1891–1895, 2019. doi: 10.21437/Interspeech.2019-3079. URL http://dx.doi.org/10.21437/Interspeech.2019-3079.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, et al. Overview of the ninth dialog system technology challenge: Dstc9. *arXiv preprint arXiv:2011.06486*, 2020.

Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. What makes a good story? designing composite rewards for visual storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7969–7976, 2020.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*, 2022.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense reasoning. *Findings of EMNLP*, 2020.

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

Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*, 2016.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment, may 2023. *arXiv preprint arXiv:2303.16634*, 2023a.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023b.

Ryan Lowe, Michael Noseworthy, Iulian V Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. Towards an automatic turing test: Learning to evaluate dialogue responses. *arXiv preprint arXiv:1708.07149*, 2017.

Shikib Mehri and Maxine Eskenazi. Unsupervised evaluation of interactive dialog with dialogpt. *arXiv preprint arXiv:2006.12719*, 2020.

Mohsen Mesgar, Sebastian Bücker, and Iryna Gurevych. Dialogue coherence assessment without explicit dialogue act labels. *arXiv preprint arXiv:1908.08486*, 2019.

OpenAI. Gpt-3.5. https://openai.com/blog/chatgpt, 2022.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.

Tiago Pimentel, Clara Meister, and Ryan Cotterell. On the usefulness of embeddings, clusters and strings for text generator evaluation. *arXiv*, 2022.

Haoyi Qiu, Kung-Hsiang Huang, Jingnong Qu, and Nanyun Peng. Amrfact: Enhancing summarization factuality evaluation with amr-driven training data generation. *arXiv preprint arXiv:2311.09521*, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

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

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

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020.

Yufei Tian, Felix Zhang, and Nanyun Peng. Harnessing black-box control to boost commonsense in LM's generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5417–5432, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.329. URL https://aclanthology.org/2023.emnlp-main.329.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Svitlana Vakulenko, Maarten de Rijke, Michael Cochez, Vadim Savenkov, and Axel Polleres. Measuring semantic coherence of a conversation. In *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part I 17*, pp. 634–651. Springer, 2018.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7378–7385, 2019.

Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee, and Haizhou Li. Dynaeval: Unifying turn and dialogue level evaluation. *arXiv preprint arXiv:2106.01112*, 2021.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*, 2024.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*, 2022a.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*, 2022b.

A. Appendix

A.1. Prompt for G-Eval and Full Results with GPT-4 Model

The prompts (for overall/coherence evaluation) we used for G-Eval are adapted from the original G-Eval repository. We hereby show case the one for dialogue overall score. Other prompts are mostly following the similar fashion.

You will be given a conversation between two agents.

Your task is to rate the dialogue on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Overall (1-10) - the overall quality of the whole dialogue.

Evaluation Steps:

1. Read the dialogue carefully to get a general understanding of the overall quality of it.

2. Assign an overall score on a scale of 1 to 10, where 1 is the lowest and 10 is the highest based on the Evaluation Criteria.

Dialogue:

{**The Dialogue**}

Evaluation Form (scores ONLY):

- Overall:

.....

Table 9. All G-Eval results on the dialogue evaluation dataset. Spearman correlation with the golden labels is reported. We highlight the best-performing results with **bolded** numbers and second-best with underlined numbers.

Model	FED		DSTC9	
	Coherence	Overall	Coherence	Overall
G-Eval (Liu et al., 2023a)				
- (w/ GPT-4, original)	**0.72**	**0.73**	**0.28**	**0.27**
- (w/ LLaMa2-7b-Vicuna)	0.57	0.54	0.15	0.14
- (w/ Flan-T5-11b)	0.49	0.48	0.19	0.18

Table 10. All G-Eval results for evaluating the commonsense metrics using CommonGen-trinity dataset. Spearman correlation with the golden labels is reported. We highlight the best-performing results with **bolded** numbers and second-best with underlined numbers.

Model	raw	binarized
G-Eval (Liu et al., 2023a)		
- (w/ GPT-4, Pseudo Upper Bound)	**0.81**	**0.67**
- (w/ LLaMa2-7b-Vicuna)	0.38	0.33
- (w/ Flan-T5-11b)	0.61	0.53

A.2. Prompt for CommonGen-trinity Synthesis

Instruction
Given several concepts (i.e., nouns or verbs), write a short and simple sentence that contains *all* the required words.
With higher commonsense strength, the sentence should describe a more natural scene.
With lower commonsense strength, introduce more abnormal usages of the concepts or incorrect relations between them.
Make sure to generate as compact sentences as possible.
Examples
Example 1
- Concepts: "dog, frisbee, catch, throw"
- Commonsense Strength: 5 out of 5
- Sentence: The dog catches the frisbee when the boy throws it into the air.

Example 2
- Concepts: "dog, frisbee, catch, throw"
- Commonsense Strength: 3 out of 5
- Sentence: A dog throws a frisbee at a dog as it tries to catch it.

Example 3
- Concepts: "dog, frisbee, catch, throw"
- Commonsense Strength: 1 out of 5
- Sentence: A dog throws a dog, while a frisbee trying to catch it.

Your Task
- Concepts: "{The concept set}"
- Commonsense Strength: {Commonsense Strength} out of 5
- Sentence: