Controllable Generation of Dialogue Acts for Dialogue Systems via Few-Shot Response Generation and Ranking

Angela Ramirez and **Kartik Aggarwal** and **Juraj Juraska**
and **Utkarsh Garg** and **Marilyn A. Walker**
University of California Santa Cruz
aramir62, kartik, mawalker@ucsc.edu

Abstract

Dialogue systems need to produce responses that realize multiple types of dialogue acts (DAs) with high semantic fidelity. In the past, natural language generators (NLGs) for dialogue were trained on large parallel corpora that map from a domain-specific DA and its semantic attributes to an output utterance. Recent work shows that pretrained language models (LLMs) offer new possibilities for controllable NLG using prompt-based learning. Here we develop a novel few-shot overgenerate-and-rank approach that achieves the controlled generation of DAs. We compare eight few-shot prompt styles that include a novel method of generating from textual pseudo-references using a textual style transfer approach. We develop six automatic ranking functions that identify outputs with both the correct DA and high semantic accuracy at generation time. We test our approach on three domains and four LLMs. To our knowledge, this is the first work on NLG for dialogue that automatically ranks outputs using both DA and attribute accuracy. For completeness, we compare our results to fine-tuned few-shot models trained with 5 to 100 instances per DA. Our results show that several prompt settings achieve perfect DA accuracy, and near perfect semantic accuracy (99.81%) and perform better than few-shot fine-tuning.

1 Introduction

Dialogue systems need to faithfully produce utterances that realize multiple types of dialogue acts (DAs), such as providing opinions, making recommendations, or requesting information. In the past, natural language generators (NLGs) for dialogue have been trained on large parallel corpora that map from a domain-specific meaning representation (MR) that specifies the desired DA and semantic attributes to an output utterance. The NLG must faithfully generate utterances that realize the style and form of the DA, and all of the specified attributes, as shown by the reference utterances in Table 1. Recent work shows that pretrained language models (LLMs) offer new possibilities for controllable NLG using prompt-based learning (PBL) (Brown et al., 2020; Radford et al., 2019; Liu et al., 2021). Here we present a novel few-shot overgenerate-and-rank approach that achieves the controlled generation of DAs.

Attributes and Values
(NAME [**Call of Duty: Advanced Warfare**], RATING [**excellent**], DEVELOPER [**Sledgehammer Games**], ESRB [**M (for Mature)**])
give_opinion
Call of Duty: Advanced Warfare must be **one of the best games** I've ever played. **Sledgehammer Games** always nail their **M-rated** games.
recommend
Since you seem to **love M-rated** games developed by **Sledgehammer Games**, I wonder if you have tried **Call of Duty: Advanced Warfare**.
inform
Developed by **Sledgehammer Games**, **Call of Duty: Advanced Warfare** is targeted at **mature audiences** and has overall **very positive ratings**.

Table 1: Sample ViGGO dialogue acts (DAs) (Juraska et al., 2019). The same attributes and values can be realized as different DAs.

Previous work on semantically-controlled NLG has focused on improving semantic accuracy (Rastogi et al.; Xu et al., 2021; Du et al., 2022; Wen et al., 2015; Kedzie and McKeown, 2020; Juraska and Walker, 2021). However, Table 1 shows how the the same set of semantic attributes can be realized by different DAs, such as *give_opinion*, *recommend* and *inform*, each of which affect the dialogue state differently (Traum and Allen, 1994).

Obviously an NLG for dialogue needs to faithfully realize the DA as well as the semantic attributes. However, previous work has neither *controlled for* nor *evaluated* DA accuracy. We speculate that this is because many NLG training sets, such as E2E, Weather, WebNLG, WikiBio, DART and ToTTo, only include *inform* DAs (Novikova et al., 2017b; Belz, 2008; Gardent et al., 2017; Lebret et al., 2016; Nan et al., 2021; Parikh et al., 2020). Yet NLG training sets for spoken dialogue include many types of DAs, e.g. the ViGGO corpus has 9 DAs (Juraska et al., 2019), the RNNLG corpus provides 13 DAs (Wen et al., 2015), MultiWOZ has 34 DAs (Eric et al., 2021), and Topical Chat was automatically labelled with 11 DAs (Hedayatnia et al., 2020; Mezza et al., 2018).

We present a few-shot PBL framework that overgenerates and ranks NLG outputs and achieves high accuracy for both semantic attributes and DAs. We develop high accuracy DA classifiers for three domains and use them to define 6 ranking functions that combine estimates of DA probability with measures of semantic accuracy. We also compare a combination of prompt formats, prompt sampling methods, and DA representations. Several prompt templates take the novel approach of treating DA control as a textual style transfer (TST) problem (Reif et al., 2022). For completeness, we report results for few-shot fine-tuned models trained with 5 to 100 instances per DA. Our contributions include:

- The first results showing that dialogue acts can be controlled with PBL;
- A new overgenerate-and-rank framework that automatically ranks generation outputs for DA accuracy at generation time;
- A systematic exploration of both domain-specific and general measures in ranking functions, and a comparison of their performance;
- Results showing that a ranking function that prioritizes DA correctness results in higher semantic accuracy.
- The definition of novel textual DA representations that support automatic ranking for semantic accuracy using off-the-shelf metrics such as BLEU and Beyond-BLEU;
- The systematic testing of 8 prompt formats that re-cast data-to-text generation as a text-to-text task, and an examination of their performance across 4 LLMs.

The results demonstrate large performance differences across prompt styles, but show that many prompts achieve perfect DA accuracy, and semantic accuracy as high as 99.81% with only 10 examples, while 100-shot per DA fine-tuning only achieves 97.7% semantic accuracy, and 80.6% DA accuracy.

2 Related Work

This paper applies few-shot PBL to the task of controllable generation of DAs using an overgenerate-and-rank NLG framework. The overgenerate-and-rank paradigm for NLG has primarily used two methods for ranking: (1) language model probability (Langkilde and Knight, 1998); and (2) ranking functions trained from human feedback (Rambow et al., 2001; Bangalore et al., 2000; Liu et al., 2016). We extend this framework by applying it in the context of PBL, by using DA probability in ranking, and by comparing many ranking functions, including Beyond-BLEU and BLEU baselines (Wieting et al., 2019; Papineni et al., 2002).

We know of only a few previous studies on controllable generation of DAs in the context of dialogue systems, each of which has only focused on one or two types of DAs. Obviously, tasks like question generation (QG) aim at controllable generation of questions (Harrison and Walker, 2018; Zhang et al., 2021) but research on QG is not focused on trying to control the generation of questions as opposed to other types of DAs. However, some work has focused on controlling questions in dialogue, e.g. Hazarika et al. (2021) learned a latent representation of questions from a labelled corpus and then used this as a prompt prefix to control question generation. See et al. (2019) fine-tuned a Persona Chat model and tested decoding methods that controlled question frequency, but did not guarantee a question on a particular turn. Other work has focused on dialogue acts like opinions and recommendations. For example, Oraby et al. (2019) curated opinionated utterances from user reviews that had been marked with exclamation points, and then used the exclamation points as a way to control the production of exaggerated opinions. Reed et al. (2020) used token supervision to control the production of `recommendation` as opposed to `inform` dialogue acts where `recommendation` DAs stated that a particular restaurant was the best and then justified the recommendation with attributes from the MR. Ramirez et al. (2023) used PBL with similar prompts to control the expression of Big 5 personality types (Harrison et al., 2019), rather than dialogue acts.

It is well known that data-to-text NLGs based on fine-tuned LLMs are prone to semantic errors (Ji et al., 2022; Rashkin et al., 2021), thus previous work has focused on methods for ensuring semantic correctness. This includes automatically augmenting the training data (Xu et al., 2021; Du et al., 2022), modifying the input representation (Kedzie and McKeown, 2020; Heidari et al., 2021), using rankers or classifiers or decoding methods that identify semantically accurate or acceptable candidates (Harkous et al., 2020; Juraska and Walker, 2021; Wen et al., 2015; Shen et al., 2019; Batra et al., 2021). Previous work on few-shot PBL for semantically-controlled NLG has not attempted to control DA accuracy (Reed et al., 2022; Soltan et al., 2022), and has not used an overgenerate and rank approach, resulting in lower semantic accuracies than we report here.

Much previous work on few-shot NLG has investigated few-shot finetuning rather than few-shot PBL. Previous work on the ViGGo, TV and Laptop corpora (Xu et al., 2021; Du et al., 2022; Kedzie and McKeown, 2020; Juraska and Walker, 2021) supports direct comparison to our work, but is not few-shot, does not rank outputs or use PBL. FewShotWoz trains a model called SC-GPT on a 400K data-to-text corpus, and then tests transfer learning with only 40 or 50 fine-tuning examples (Peng et al., 2020). Other recent work develops methods for augmenting FewShotWoz using synthetic data or by self-training and shows improvements in semantic accuracy and BLEU score. The FewShotWoz corpus includes many types of DAs but none of this previous work includes an evaluation of NLG DA accuracy. Previous work on few-shot finetuning in the weather domain used 300 examples in fine tuning, and also explored different ways of textualizing the MR (Heidari et al., 2021), but did not attempt to control DAs, develop ranking functions, evaluate DA accuracy, or use instructions such as our novel definitional prompts and the templates for TST tasks. Heidari et al. (2021) achieve an 85% reconstruction accuracy, while our best prompt/LLM combinations achieve 99.44% PERF score for ViGGO, 99.57% PERF for TV and 99.47% PERF for Laptop, a similar metric to reconstruction accuracy, with only 10 examples.

3 Automatically Ranking NLG Outputs

We start by providing a mathematical formulation of our problem. When generating from a DA representation, a high-quality response should: (1) manifest the specified DA; (2) have no missing or incorrect mentions of the attributes; (3) hallucinate no additional attributes; and (4) be fluent. Thus the generated utterance y, conditioned on an input x composed of DA d and attribute values a, can be formulated as $y = f(d, a)$. The conditional likelihood of y given the MR can then be decomposed using Bayes Rule into the product of three probabilities:

$$p(y|d, a) = p(d|y, a) * p(a|y) * p(y) \qquad (1)$$

The term $p(d|y, a)$ is the DA probability given the generated utterance y and the semantic attributes a. The term $p(a|y)$ represents the semantic accuracy. The term $p(y)$ is the unconditional probability of the generated utterance, which is commonly used as a measure of fluency. Below, we show how we compute estimates of these terms at generation time, and then explain their use in the ranking functions.

Dialogue Act Classifier. The term $p(d|y, a)$ requires highly accurate DA classifiers to use in automatic ranking. We fine-tuned two classifiers using pre-trained bert-base-uncased on HuggingFace. We discovered that even though the ViGGO, Laptop and TV training corpora are good size (Juraska et al., 2019; Wen et al., 2015), producing high accuracy classifiers required us to modify the training data.[1] We originally trained the ViGGO classifer with the original ViGGO training set, when we applied this classifier to the generated outputs, we noticed many cases of low confidence classification. A qualitative analysis of the data showed that many generated outputs did not actually fit into the original ViGGO ontology, which is not surprising, given that the training data for an LLM would have included many different types of DAs.

To increase the ViGGO classifier performance, we introduced an "Other" class of dialogue acts, doubly annotated another 1000 ViGGO NLG outputs by hand, and added them to the original training set. Final results are shown in Table 2.

The second classifier was trained using the complete RNNLG corpus with all 4 domains to maximize classifier domain transfer. When we tested it on the RNNLG test set, we discovered that several classes had low F1. Examination of the confusion matrix showed that the *recommend* and *inform* DAs were highly confusable, so we created a new type of DA we call “describe” by combining their

[1] We also experimented with training classifiers for MultiWoz but were unable to get high accuracies due to noise in DA labelling, which is known to be an issue with MultiWoz (Zou, 2022).

Dialogue Act	ViGGO
confirm	0.99
inform	0.98
suggest	0.91
give_opinion	0.90
recommend	0.92
request	0.94
request_attribute	0.93
request_explanation	0.99
verify_attribute	0.94
other	0.78
Weighted Average	0.97

Table 2: ViGGO DA classification F1 scores.

training sets. The final results for for the RNNLG classifiers is shown in Table 3.

Dialogue Act	Laptop	TV
compare	1.00	1.00
confirm	0.96	0.95
describe	1.00	1.00
inform all	0.86	0.92
inform count	1.00	1.00
inform no info	1.00	1.00
inform no match	0.98	0.94
inform only match	0.83	0.87
suggest	1.00	1.00
Weighted Average	0.99	0.99

Table 3: Laptop and TV DA classification F1 scores. The *describe* DA = combination of the *inform* and *recommend* DAs in the original dataset.

We provide these DA classifiers along with additional human-labelled model outputs so that other researchers can duplicate our setup.[2] The resulting classifiers achieve average F1s over .97 for all three domains.

Semantic Accuracy. Work on data-to-text NLG often computes semantic accuracy as the Slot Error Rate (SER), i.e., the percentage of slots across all outputs y that the NLG realized incorrectly, with models either carefully tuned by hand, or trained by artificially creating incorrect realizations (Wen et al., 2015; Dusek et al., 2019; Juraska et al., 2018; Reed et al., 2020; Wiseman et al., 2017; Harkous et al., 2020; Kedzie and McKeown, 2019, 2020). There is a toolkit for SER for all three domains,[3] which we use to calculate SACC:

$$\text{SACC} = 1 - \text{SER} \tag{2}$$

Because the SACC scripts are domain specific, we also create new metrics that are based on BLEU, BLEURT, Beyond-BLEU and BertScore, widely used measures of semantic accuracy and semantic preservation (Papineni et al., 2002; Wieting et al., 2019; Sellam et al., 2020; Zhang et al.; Gehrmann et al., 2021). Because these metrics require comparisons with reference utterances, which are not available at generation time, we define referenceless versions based on pseudo-references, S_{pseudo}, created from the input DAs Juraska (2022). For any MR, we create its S_{pseudo} by omitting the slot names and the DA name and then concatenating the categorical attribute values with spaces between them, and converting boolean attributes, such as HAS_MULTIPLAYER = no, into phrases using the attribute name, with a negation when needed, e.g. “no multiplayer”. For example, S_{pseudo} for the MR at the top of Table 1 would be “Call of Duty: Advanced Warfare excellent Sledgehammer Games M for Mature". Pseudo-references are available at generation time, so we use them to calculate pseudo-metrics for semantic accuracy and use them in ranking. Juraska et al. (2019) shows that the *relative* differences of these pseudo-metrics distinguish errorful NLG utterances from correct ones.

Fluency. Recent work suggests that the probability P(S) of a generated output S according to an LLM is a good automatic and referenceless measure of fluency (Kann et al., 2018; Suzgun et al., 2022). We thus adopt P(S) to measure fluency, and use GPT-2 to calculate P(S).

Ranking. The ranking functions in Table 4 aim to select NLG outputs that maximize DA accuracy, semantic accuracy, and fluency. Ranking function RF1 scores each candidate according to Equation 1.

RF1: DAC * SACC * P(S)
RF2: DAC * SACC * pBLEU * P(S)
RF2$_{\text{DA}}$: DAC | SACC | pBLEU | P(S)
RF3: DAC * pBBLEU * P(S)
RF4: pBBLEU
RF5: pBLEU

Table 4: Ranking functions. DAC = probability of the correct DA using a classifier. SACC = semantic accuracy using domain-specific SACC scripts. P(S) = LM probability as a measure of fluency. pBBLEU = pseudo-Beyond-BLEU to measure semantic accuracy. pBLEU = pseudo-BLEU as a baseline.

After a qualitative analysis of the ranking outputs from RF1 on pilot data, we developed ranker RF2 and RF2$_{DA}$ in Table 4. Our analysis revealed

[2] https://github.com/aramir62/da-nlg

[3] https://github.com/jjuraska/data2text-nlg

that the SER scripts often do not detect hallucinations, but pBLEU appeared to detect some hallucinations, so we add pBLEU to RF2. Ranking function RF2$_{DA}$ prioritizes one metric at each step, as represented by | in RF2$_{DA}$, enforcing DA correctness as more important for dialogue than perfect SACC. Matching DA candidates are preferred, but if no candidates match the required DA, the DA class *other* is preferred, or otherwise, all k candidates are selected. The second step selects candidates with the highest SACC. The third step aims to remove candidates with hallucinations by choosing the highest pBLEU outputs. The final step selects outputs with the highest fluency (P(S)).

So far RF1, RF2 and RF2$_{DA}$ all use the domain-specific SACC score for measuring semantic accuracy. To define a domain-independent ranking function, we calculate the correlation of SACC with pBLEU, pBBLEU, pBERT, and pBLEURT, defined in Section 3, on sample model outputs. See Table 12 in Appendix A.2. The results show that pBBLEU (Wieting et al., 2019) has the highest correlation across all three domains with 0.52 for Viggo, 0.32 for Laptop and 0.45 for TV. We thus define RF3 by replacing SACC in RF1 with pBBLEU. We then define RF4 as pBBLEU alone, so we can compare our novel ranking functions to pBBLEU. Finally, as a baseline reflecting the fact that previous work uses BLEU as a single measure of goodness for NLG, we define R5 as pBLEU.

4 Experimental Overview

Figure 1 provides an overview of the experimental architecture. Given a set of DA representations for a domain, we sample prompt examples from the original training sets while varying the number of samples. We then textualize the DA representations in the sample to look more similar to the LLMs free-text training data. The samples are then fed through the 8 prompt formats in Table 5. We apply this method to the ViGGO, Laptop and TV domains and utilize the 6 ranking functions in Table 4.

Prompt Formats. LLMs are typically trained on far more monologic data than dialogue, and will have rarely, if at all, seen examples of data-to-text NLG (Brown et al., 2020; Raffel et al., 2020; Devlin et al., 2018). While there are LLMs trained on dialogue such as DialoGPT (Zhang et al., 2020), and semantically-controlled dialogue data such as KGPT (Chen et al., 2020), and SC-GPT (Peng et al., 2020), there are clear benefits to using a general LLM. Previous work also shows that without specific dialogic data, many LLMs do well on NLG for dialogue (Soltan et al., 2022). Here, we test the hypothesis that performance can be improved by using prompt formats that make the data-to-text task look more like the LLM's textual training data.

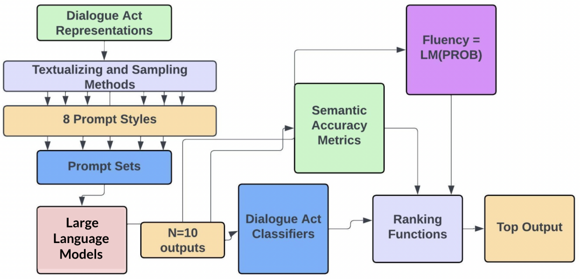

Figure 1: Experimental Architecture

Prompt ID	Prompt Template
TST VANILLA	Here is a text: "s_{pseudo}". Here is a rewrite of the text which is a(n) d dialogue act: "r_{text}"
TST DIALOGUE	Here is a text: "s_{pseudo}". Rewrite it to be a(n) d dialogue act: "r_{text}"
TST PARAPHRASE	Here is a text: "d_r s_{pseudo}". Here is a paraphrase of the text: "r_{text}"
DEFINITIONAL	description of $< d >$: D^d. Data: $d = yes \mid sa^1 = v^1..sa^n = v^n$ Data to Text for $< d >$: r_{text}
PARAPHRASE	d_r s_{pseudo} r_{text}
DIALOGIC	d_r s_{pseudo} r_{text}
PSEUDO	d s_{pseudo} r_{text}
S2S	$d = yes \mid a^1 = v^1..a^n = v^n$ r_{text}

Table 5: Prompt IDs and templates. Instantiations of each template are given in Table 11 in the Appendix.

Table 5 shows the 8 prompt templates, with full instantiations in the Appendix in Table 11. The templates vary the representation of the DAs and their attributes. We represent the DA directly by its name d, or convert the DA to a sentence starter d$_r$ such as "I recommend". The attributes of the DA constitute a set $a = a^1, a^2, ..., a^n$, each with a value in v where $v = v^1, v^2, ..., v^n$. The attributes can be represented directly or using a textual pseudo-reference s_{pseudo}, as described in Section 3. The reference text r_{text} then varies the representation of the DA and the attributes.

Prompts TST Vanilla, TST Dialogue, and TST Paraphrase of Table 5 treat data-to-text generation as a textual style transfer (TST) task, where each DA is a style, and the prompt provides instructions, e.g., “Rewrite it to be a suggest dialogue act” (Reif et al., 2022; Suzgun et al., 2022). TST Vanilla and TST Dialogue represent the MR as its pseudo-reference s_{pseudo}, while TST Paraphrase prefixes the sentence starter d_r for the DA to s_{pseudo}.

We also define a Definitional prompt with definitions of the DAs, represented as D^d, based on the instructions given to crowdworkers when ViGGO was collected, inspired by previous work providing slot descriptions (Gupta et al., 2022).

The Paraphrase prompt is based on the fact that producing paraphrases is a common task. This prompt rewrites the DA as a first-person sentence starter, e.g., “I suggest” for the *suggest* DA. The Dialogue Response prompt is similar, but mimics a request and its response, with sentence starters written as requests, e.g., “can you recommend a game Worms: Reloaded Steam?” for the *recommend* DA.

To directly evaluate the benefit of instructions, we also input the pseudo-reference without instructions as a baseline (Pseudo), as well as input the commonly used S2S format which linearizes the MR as a sequence of attributes and values (Soltan et al., 2022; Wen et al., 2015; Harkous et al., 2020).

5 Results

Experimental Roadmap. We first experiment with ViGGO over all the experimental settings from Section 4 using Jurassic-1 Jumbo, a 175B autoregressive transformer-based LLM with a different depth-width tradeoff than GPT3 (Levine et al., 2020; Lieber et al., 2021). All experiments set top P = 1, and T = 0.7 based on pilot experiments. We compare prompting to few-shot fine-tuning using 5, 25, 50 and 100 examples per DA sampled from the training data. We test the 8 prompt formats in Table 5 with 1, 5 or 10 prompt examples. Our focus is DA control, so we create a ViGGO test set with 40 instances per DA (360 total). We look-ahead to see which ranking function performs best for ViGGO and use that for the results in Table 6.

We then test the best settings from ViGGO on the Laptop and TV corpora (Wen et al., 2015) with results in Table 7. We compare ranking function performance across all domains in Table 8, and demonstrate the improved performance of our ranking functions compared to simply using BLEU. We then test for generalization with additional LLMs: we select the top three prompt settings, and test of GPT-Neo as a smaller LLM, and GPT-3 and ChatGPT as instruction-tuned LLMs, and compare them to Jurassic-1, for all three domains. These results are shown in Table 9. Table 10 then compares our best performance to recent SOTA results for both fine-tuning and few-shot fine-tuning on ViGGO, Laptop and TV. Finally we report the results of our human evaluations. We make the DA classification models, the prompts and their instantiations, and the model outputs for all experiments available.[4]

ID	N	PERF	SACC	DAC
Few-Shot Fine-Tuning Experiments				
FTune 5-per	45	38.88	85.71	54.44
FTune 25-per	225	62.22	92.19	79.72
FTune 50-per	450	71.94	96.43	79.44
FTune 100-per	900	**78.61**	**97.74**	**80.56**
Prompt Styles and Samples Experiments				
TST Vanilla	10	**85.56**	**94.73**	**100.00**
TST Dialogue	10	83.89	94.17	100.00
TST Paraphrase	10	83.90	94.20	100.00
Definition (each)	10	76.94	91.16	100.00
Definition (top)	10	82.22	93.51	100.00
Paraphrase	10	77.78	92.10	100.00
Dialogic	10	77.22	91.53	100.00
Pseudo	10	75.83	94.17	100.00
S2S	10	70.56	86.45	100.00
TST Vanilla	5	80.56	92.57	99.72
TST Dialogue	5	**83.61**	**93.88**	**100.00**
TST Paraphrase	5	80.20	92.60	99.70
Definition (each)	5	80.00	92.66	99.40
Definition (top)	5	77.22	91.25	100.00
Paraphrase	5	70.83	89.71	100.00
Dialogic	5	66.94	88.34	99.10
Pseudo	5	52.22	82.60	85.56
S2S	5	66.67	83.54	99.72
TST Vanilla	1	68.06	86.64	91.94
TST Dialogue	1	69.17	88.15	93.30
TST Paraphrase	1	**72.20**	**89.80**	93.60
Definition	1	63.89	85.32	**98.30**
Paraphrase	1	41.94	75.14	83.88
Dialogic	1	38.89	71.83	82.30

Table 6: Results after ranking via $\mathrm{RF2}_{\mathrm{DA}}$ for ViGGO. N = number of prompt examples. PERF = % outputs that are perfect. SACC = semantic accuracy using SACC scripts. DAC = DA accuracy using a classifier.

Few-Shot Fine-Tuning. To compare prompting to fine-tuning, we use the traditional linearized MR in the S2S format and vary the number of training examples per DA in few-shot fine-tuning from 5, to 25, to 50, to 100. The results in Rows 1-4 of Table 6 show that, as expected, increasing the number of training examples improves performance, with 100 examples per DA (900 overall) achieving a SACC of 97.74 after ranking. However, interestingly, the highest DAC performance is only 80.56, and the PERF score (both perfect DA and perfect SACC) is only 78.61. Table 13 in the Appendix shows more

[4] https://github.com/aramir62/da-nlg

detail, providing before and after ranking performance for fine-tuning. Overall, the results affirm previous findings that few-shot prompting beats few-shot fine-tuning (Le Scao and Rush, 2021).

Prompt Styles. All experiments provide examples for a single DA and then generate that DA, while varying the prompt style and the number of examples. The TST format provides N examples using one of the TST prompts in Table 5. The Definitional (each) format, for 10 prompts, provides 10 triplets of (definition, MR, text). For Definitional (top), the definition is mentioned once before all the MRs and examples, so for 1 prompt, there is no difference between *top* and *each*.

We first notice in Table 6 that the PERF score improves with the number of prompt examples, from 1 to 5 to 10 for all the prompt styles, with TST Vanilla, TST Dialogue, and TST Paraphrase, which provide the MR as text and include instructions (see Table 5) consistently performing the best overall. TST Vanilla-10 performs signicantly better than the other TST styles with 10 examples ($p < .01$), but TST Dialogue is the best for 5 examples and TST Paraphrase is the best for 1 example. The Definitional, Paraphrase and Dialogic formats all perform significantly worse than the TST formats, but interestingly the Definitional format gets the highest DAC with only 1 example perhaps showing the advantage of explicit definitions in PBL.

The Pseudo and S2S prompt styles are baselines, and only reported for the 5 and 10 example settings. Both baselines indicate the benefits of instructions. The S2S 10 performance is the worst for 10 examples, and the Pseudo performance is the worst for 5 examples. It is worth noting that the poorly performing S2S representation is commonly used in both fine-tuning and PBL (Soltan et al., 2022; Wen et al., 2015; Harkous et al., 2020).

Domain	**ID**	**N**	**PERF**	**SACC**	**DAC**
Laptop	TST Van.	10	80.95	95.90	100.00
TV	TST Van.	10	98.85	99.76	100.00

Table 7: Results for Laptop and TV for TST 10 using $RF2_{DA}$. N = number of examples. PERF = % outputs that are perfect. SACC = semantic accuracy using SACC scripts. DAC = DA accuracy using a classifier.

We then take the best performing prompt (TST Vanilla) and experiment with TV and Laptop. The results are shown in Table 7. $RF2_{DA}$ performs the best for both Laptop and TV so these results are ranked with $RF2_{DA}$. Interestingly, TV has the highest PERF and SACC seen so far, while Laptop also has a higher SACC than any ViGGO setting, suggesting that it is easier to achieve high performance with Laptop and TV than ViGGO.

RF	**Terms**	PERF	SACC	DAC	BLEU
	ViGGO				
RF1	**DAC, SACC, P(S)**	79.17	91.82	99.72	38.41
RF2	**DAC, SACC, pBLEU, P(S)**	78.33	91.72	99.00	38.67
$RF2_{DA}$	**DAC, SACC, pBLEU, P(S)**	**85.56**	**94.73**	**100.00**	40.08
RF3	**DAC, pBBLEU, P(S)**	62.78	84.38	100.00	**49.87**
RF4	**pBBLEU**	60.55	91.63	77.78	42.82
RF5	**pBLEU**	44.22	81.66	75.28	40.08
	TV				
RF1	**DAC, SACC, P(S)**	85.40	96.86	100.00	72.55
RF2	**DAC, SACC, pBLEU, P(S)**	88.19	97.43	100.00	72.55
$RF2_{DA}$	**DAC, SACC, pBLEU, P(S)**	**98.85**	**99.76**	**100.00**	60.51
RF3	**DAC, pBBLEU, P(S)**	73.96	93.87	100.00	**72.89**
RF4	**pBBLEU**	90.14	97.88	99.71	60.51
RF5	**pBLEU**	63.45	91.50	99.57	66.71
	Laptop				
RF1	**DAC, SACC, P(S)**	49.25	86.70	100.00	61.24
RF2	**DAC, SACC, pBLEU, P(S)**	57.29	89.47	100.00	59.39
$RF2_{DA}$	**DAC, SACC, pBLEU, P(S)**	**80.95**	**95.90**	**100.00**	**61.36**
RF3	**DAC, pBBLEU, P(S)**	35.55	80.41	100.00	45.03
RF4	**pBBLEU**	61.79	90.97	98.88	36.32
RF5	**pBLEU**	42.38	84.25	97.77	**61.36**

Table 8: Ranking functions performance.

Ranking Functions. Our results show that our overgenerate-and-rank method has a huge effect on performance as compared to taking the first output from the model. Section A.3 in the Appendix provides more detail, e.g. showing for Viggo, across all the experiments, *Before Ranking* has an average SACC of 65.29% versus an *After Ranking* average of 86.82%, while DAC has an almost a 30% increase with a *Before Ranking* average of 62.11%, and an *After Ranking* average of 91.04%.

Table 8 compares the 5 ranking functions from Section 3 on all three domains for the best prompt so far: TST Vanilla 10. The differences between RF1 and RF2 (addition of pBLEU) are not significant for ViGGO, but are significant for TV (t-test, $p < 0.001$) and Laptop (t-test, $p < 0.001$), with Laptop improving from 49.24 PERF to 57.29 PERF. Note that in all domains ranking by $RF2_{DA}$ results in significantly higher performance across all metrics (t-test, $p < 0.001$): **prioritizing DA correctness results in higher SACC and higher PERF**.

Table 8 also shows that replacing SACC with pBBLEU in RF3 results in a clear drop in performance. As shown in Appendix Section A.2 pBBLEU is the best performing pseudo-metric overall, but there are clear advantages to the domain-specific SACC. Recent work explores automatic methods for training domain-specific semantic fidelity classifiers, but these methods rely on large training corpora making them difficult to apply in few-shot settings (Harkous et al., 2020; Batra et al., 2021).

The baseline RF4 with only the pBBLEU term performs surprisingly well in SACC across all three domains, suggesting that it might be worth examining further combinations of BBLEU with DAC.

MODEL	PROMPT	PERF	SACC	DAC	BLEU
ViGGO					
ChatGPT	TST 10	98.89	95.58	99.44	45.05
ChatGPT	TST 5	94.72	99.34	96.67	40.88
ChatGPT	Def 10	98.89	**100.00**	100.00	42.40
ChatGPT VO	Def 10	95.28	99.85	95.83	14.79
GPT 3	TST 10	95.00	98.49	98.33	40.26
GPT3	TST 5	95.28	98.31	98.89	**54.11**
GPT3	Def 10	**99.44**	99.81	**100.00**	42.75
GPT3 VO	Def 10	95.28	99.83	95.55	9.55
Jurassic	TST 10	85.56	94.70	100.00	40.08
Jurassic	TST 5	83.61	93.88	100.00	32.54
Jurassic	Def 10	82.22	93.51	100.00	15.77
GPT NEO 1.3B	TST 10	17.78	85.32	35.56	25.25
GPT NEO 1.3B	TST 5 dial	64.17	86.74	94.72	43.47
GPT NEO 1.3B	Def 10	35.56	78.27	81.94	15.44
TV					
ChatGPT	TST 10	98.00	99.57	99.93	45.98
ChatGPT	TST 5	91.23	98.14	100.00	38.22
ChatGPT	Def 10	98.00	99.30	99.64	50.97
GPT 3	TST 10	**99.57**	99.91	**100.00**	57.92
GPT3	TST 5	99.07	99.81	100.00	71.80
GPT3	Def 10	99.22	**99.94**	100.00	73.81
Jurassic	TST 10	98.85	99.76	100.00	60.51
Jurassic	TST 5	91.80	98.26	100.00	**74.73**
Jurassic	Def 10	95.01	98.94	100.00	73.66
GPT NEO 1.3B	TST 10	83.15	96.37	100.00	66.28
GPT NEO 1.3B	TST 5 dial	50.78	93.15	73.93	31.95
GPT NEO 1.3B	Def 10	15.74	78.61	65.88	19.29
Laptop					
ChatGPT	TST 10	**97.08**	99.47	99.58	41.45
ChatGPT	TST 5	85.95	97.19	99.43	23.36
ChatGPT	Def 10	67.54	90.37	99.92	36.00
GPT 3	TST 10	84.79	**99.91**	**100.00**	33.20
GPT3	TST 5	94.79	97.14	100.00	32.41
GPT3	Def 10	81.45	92.54	100.00	**85.40**
Jurassic	TST 10	80.95	95.90	100.00	61.36
Jurassic	TST 5	81.55	96.10	99.81	12.94
Jurassic	Def 10	55.98	45.60	100.00	29.12
GPT NEO 1.3B	TST 10	68.89	92.66	100.00	46.21
GPT NEO 1.3B	TST 5 dial	71.89	93.55	100.00	19.49
GPT NEO 1.3B	Def 10	1.33	43.73	99.96	14.59

Table 9: Experiments with additional LLMs, with the top three prompt settings, for ViGGO, Laptop and TV, using the RF2$_{\mathrm{DA}}$ ranking function. We also tested here with the original ViGGO test set, with ChatGPT Def 10 and GPT-3 Def 10, with results shown in cyan, to facilitate comparison with previous work.

Finally, the pBLEU baseline of RF5 reinforces work emphasizing the inadequacies of BLEU as a metric for NLG (Belz, 2008; Liu et al., 2016; Novikova et al., 2017a). We report BLEU for comparison with related work, but Table 8 clearly shows that the highest BLEU score doesn't correspond to the best PERF or SACC, and that even ranking with pBLEU (RF5) doesn't maximize BLEU. RF5 gets the lowest PERF, SACC and DAC scores for ViGGO and TV, and RF2$_DA$ achieves the same BLEU score, with much higher PERF, SACC and DAC for both ViGGO and Laptop.

Experiments with other LLMs. We also compare our results with Jurassic to other LLMs. We select the three best prompt settings, namely TST 10, TST 5, and Definitional Top 10, and experiment with ChatGPT and GPT-3 as large instruction-based models and GPT-Neo 1.3 as a small model.

Table 9 presents the results. Our primary metric is PERF with best PERF shown in bold. Note in the table that the highest PERF score does not necessarily correspond with the highest SACC or highest BLEU. Interestingly, GPT-3 performs slightly better than ChatGPT for both ViGGO and TV while ChatGPT performs best for Laptop. Both ChatGPT and GPT-3 perform significantly better than Jurassic across all three domains. Table 9 shows that the Definitional prompt performs better than TST 10 with both ChatGPT and GPT-3 for Viggo, while TST 10 for TV was comparable to Definitional and performs the best for Laptop in terms of PERF. We add results here for the original ViGGO test set shown in cyan, which has a skewed distribution of DAs with more long Inform DAs, and which appears to be more challenging for DAC but not SACC. Finally, we see much worse performance with GPT Neo, reinforcing results suggesting a model size threshhold for PBL (Wei et al.).

Comparison with SOTA. Table 10 compares our best results with recent work on the VIGGO, Laptop and TV corpora (Xu et al., 2021; Du et al., 2022; Juraska and Walker, 2021; Kedzie and McKeown, 2020; Harkous et al., 2020; Peng et al., 2020). The related work either used fine-tuning or few-shot fine-tuning, rather than PBL. JW21, DT and K-McK are based on fine-tuning. SC-GPT, AUGNLG and ST-SA are all based on FEWSHOTWOZ. In each case, we take the results exactly as reported in the related work. These results are indicative only as e.g. FEWSHOTWOZ does not use the original RNN-NLG test set for Laptop and TV, which we use here. We created our own ViGGO test set to have equal numbers of each DA, but the original test set has many more long *inform* DAs.

Human Evaluation. Given the almost perfect performance reported in Table 9, we conducted a human evaluation to check whether the outputs were indeed perfect (the right DA and the correct semantics), and whether there were any hallucinations. Two expert annotators hand-labelled 100 outputs from ChatGPT with TST-10 Vanilla prompts. Amazingly, neither annotator found any outputs that weren't perfect and neither did they find any hallucinations. They agreed 100% on the results, resulting in a Cohen's Kappa of 1.0.

Model	**Laptop**		**TV**		**ViGGO**	
	BLEU↑	ERR↓	BLEU↑	ERR↓	BLEU↑	ERR↓
Ours	33.20	**0.08**	**73.81**	**0.06**	14.79	**0.15**
JW21	–	–	–	–	**53.60**	0.46
DT					**53.60**	1.68
K-McK	–	–	–	–	48.50	0.46
SC-GPT	32.73	3.39	32.95	3.38	–	–
AUGNLG-SC	34.32	2.83	34.99	5.53	–	–
ST-SA	**35.42**	2.04	36.39	1.63	–	–

Table 10: Ours = Our best model for each domain from Table 9 compared to recent SOTA results. Our VIGGO result is for the ViGGO ORIGINAL test set. JW21 = SeaGuide (Juraska and Walker, 2021). DT = Data Tuner (Harkous et al., 2020). K-McK = (Kedzie and McKeown, 2020). SC-GPT = (Peng et al., 2020). AugNLG = (Xu et al., 2021). ST-SA = (Du et al., 2022). We convert SACC to SER, which other work calls ERR, and report BLEU, and ERR as in that other work. Note that we use our best SACC score from Table 9 to select the row to include here, but this doesn't necessarily correspond to the best BLEU score or the best PERF score.

We also test whether our addition of pBLEU to RF2 has an effect on hallucinations, by testing in general whether pBLEU helps identify hallucinations. We annotate hallucinations for ViGGO, by having 3 annotators label all 360 outputs for each ranking function (6*360) shown in Table 8. The number of hallucinations for RF1 was 34, RF2 was 19, RF3 was 26, RF4 was 40 and RF5 was 14. We compared the mean number of hallucinations of ranking functions with pBLEU, namely RF2, $RF2_{DA}$, and RF5 to those without, namely RF1, RF3 and RF4. We find that the mean number of hallucinations of those with pBLEU is 31.67, while the mean number of those without is 19.67. This difference seems large, but the sample size is small and therefore it's not significant (t = 1.82, p = .14)

6 Conclusion and Future Work

Here we apply an overgenerate-and-rank NLG approach and and provide the first experiments using automatic ranking functions that optimize both DA and semantic accuracy in few-shot prompt-based NLG. We test and compare a combination of prompt formats, sampling methods, and DA representations. We test prompts used for textual style transfer (TST) by treating DAs as styles to be controlled. We also create novel prompts that provide definitions of DAs, For completeness, we fine-tune few-shot models and compare them with the few-shot results. The results show that several prompting styles achieve perfect DA accuracy, and that few-shot methods can achieve semantic accuracy as high as 99.81% with the right ranking function, while 100-shot fine-tuning achieves 97.7%, and performs much worse on DA accuracy (80.6%).

Our contributions include systematic experimentation with different ways of textualizing MRs, providing instructions to the LLM, and ranking outputs. Our results also show that formulating the data-to-text task as textual style transfer using pseudo-references yields the highest performance. We achieve SOTA semantic accuracy with only 10 prompt examples with our best prompt styles, and achieve the surprising results that a ranking function that prioritizes DA correctness results in higher semantic accuracy.

Limitations and Risks One limitation arises from the challenges of prompt-engineering: it is impossible to tell whether another prompt format could perform better, e.g. with smaller LLMs like GPT-Neo, where we get poor comparative results. Another limitation is the need for a high-accuracy DA classifier that works well on out-of-domain model outputs. We address this limitation by releasing our classifiers. Another possible limitation is the use of the overgenerate and rank approach in real-time. In future work we plan to use the high quality (ranked) generated data, to fine-tune a smaller real-time language model, without the need for overgeneration. Another limitation arises from the comparison to few-shot fine-tuning – there are many ways to fine tune and many representations of the MRs, so it is possible that some other method of fine-tuning would lead to better fine-tuning results (Liu et al., 2022). Our main goal here was to show that with a small-number of examples, using reasonable assumptions, few-shot fine-tuning performs worse than PBL.

A potential risk of using LLMs is the possibility of disinformation, often called hallucinations. Control of hallucinations is an active area of research. One of the challenges is that it is very difficult to automatically identify them. Here we experiment with ranking functions for better control of hallucinations, hand-label hallucinations and characterize them. Another potential risk of our work is that some of our dialogue acts like recommend and suggest could be used, in an application context, to persuade a user to buy something. In this context, it is even more important to ensure that the system is not providing false information to users.

References

Srinivas Bangalore, Owen Rambow, and Steve Whittaker. 2000. Evaluation metrics for generation. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 1–8.

Soumya Batra, Shashank Jain, Peyman Heidari, Ankit Arun, Catharine Youngs, Xintong Li, Pinar Donmez, Shawn Mei, Shiunzu Kuo, Vikas Bhardwaj, Anuj Kumar, and Michael White. 2021. Building adaptive acceptability classifiers for neural NLG. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 682–697, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Anja Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. Kgpt: Knowledge-grounded pretraining for data-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wanyu Du, Hanjie Chen, and Yangfeng Ji. 2022. Self-training with two-phase self-augmentation for few-shot dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2770–2784.

Ondrej Dusek, David M Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *INLG*.

Mihail Eric, Nicole Chartier, Behnam Hedayatnia, Karthik Gopalakrishnan, Pankaj Rajan, Yang Liu, and Dilek Hakkani-Tur. 2021. Multi-sentence knowledge selection in open-domain dialogue. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 76–86, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, et al. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120. Association for Computational Linguistics.

Raghav Gupta, Harrison Lee, Jeffrey Zhao, Abhinav Rastogi, Yuan Cao, and Yonghui Wu. 2022. Show, don't tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Vrindavan Harrison, Lena Reed, Shereen Oraby, and Marilyn Walker. 2019. Maximizing stylistic control and semantic accuracy in nlg: Personality variation and discourse contrast. *INLG Workshop on Discourse Structure in NLG 2019*, page 1.

Vrindavan Harrison and Marilyn Walker. 2018. Neural generation of diverse questions using answer focus, contextual and linguistic features. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 296–306.

Devamanyu Hazarika, Mahdi Namazifar, and Dilek Hakkani-Tür. 2021. Zero-shot controlled generation with encoder-decoder transformers. *arXiv preprint arXiv:2106.06411*.

Behnam Hedayatnia, Seokhwan Kim, Yang Liu, Karthik Gopalakrishnan, Mihail Eric, and Dilek Hakkani-Tur. 2020. Policy-driven neural response generation for knowledge-grounded dialogue systems. *arXiv preprint arXiv:2005.12529*.

Peyman Heidari, Arash Einolghozati, Shashank Jain, Soumya Batra, Lee Callender, Ankit Arun, Shawn Mei, Sonal Gupta, Pinar Donmez, Vikas Bhardwaj, Anuj Kumar, and Michael White. 2021. Getting to production with few-shot natural language generation models. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 66–76, Singapore and Online. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *CoRR*, abs/2202.03629.

Juraj Juraska. 2022. *Diversifying Language Generated by Deep Learning Models in Dialogue Systems*. Ph.D. thesis, UC Santa Cruz.

Juraj Juraska, Kevin K Bowden, and Marilyn Walker. 2019. ViGGO: A video game corpus for data-to-text generation in open-domain conversation. In *Proceedings of the 12th International Conference on Natural Language Generation*.

Juraj Juraska, Panagiotis Karagiannis, Kevin Bowden, and Marilyn Walker. 2018. A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 152–162.

Juraj Juraska and Marilyn Walker. 2021. Attention is indeed all you need: Semantically attention-guided decoding for data-to-text NLG. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 416–431, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323.

Chris Kedzie and Kathleen McKeown. 2020. Controllable meaning representation to text generation: Linearization and data augmentation strategies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5160–5185.

Chris Kedzie and Kathleen R McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *INLG*.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.

Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. 2020. The depth-to-width interplay in self-attention. *arXiv preprint arXiv:2006.12467*.

Opher Lieber, Barak Lenz Sharir, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. *Technical report, AI21 Labs*.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Stefano Mezza, Alessandra Cervone, Evgeny Stepanov, Giuliano Tortoreto, and Giuseppe Riccardi. 2018. Iso-standard domain-independent dialogue act tagging for conversational agents. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3539–3551.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

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

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017b. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Shereen Oraby, Vrindavan Harrison, Abteen Ebrahimi, and Marilyn Walker. 2019. Curate and generate: A corpus and method for joint control of semantics and style in neural nlg. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5938–5951.

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

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182, Online. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Owen Rambow, Monica Rogati, and Marilyn Walker. 2001. Evaluating a trainable sentence planner for a spoken dialogue system. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 434–441.

Angela Ramirez, Mamon Alsalihy, Kartik Aggarwal, Cecilia Li, Liren Wu, and Marilyn Walker. 2023. Controlling personality style in dialogue with zero-shot prompt-based learning. ArXiv preprint arXiv:2302.03848.

Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. Increasing faithfulness in knowledge-grounded dialogue with controllable features. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 704–718, Online. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34.

Lena Reed, Vrindavan Harrison, Shereen Oraby, Dilek Hakkani-Tur, and Marilyn Walker. 2020. Learning from mistakes: Combining ontologies via self-training for dialogue generation. In *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2020)*.

Lena Reed, Cecilia Li, Angela Ramirez, Liren Wu, and Marilyn Walker. 2022. Jurassic is (almost) all you need: Few-shot meaning-to-text generation for open-domain dialogue. In *Conversational AI for Natural Human-Centric Interaction: 12th International Workshop on Spoken Dialogue System Technology, IWSDS 2021, Singapore*, pages 99–119. Springer.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. Association for Computational Linguistics.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1702–1723.

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

Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically informative text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4060–4067, Minneapolis, Minnesota. Association for Computational Linguistics.

Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. 2022. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models.

David Traum and James Allen. 1994. Discourse obligations in dialogue processing. In *32nd Annual Meeting of the Association for Computational Linguistics*, pages 1–8.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263.

Xinnuo Xu, Guoyin Wang, Young-Bum Kim, and Sungjin Lee. 2021. Augnlg: Few-shot natural language generation using self-trained data augmentation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1183–1195.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2021. A review on question generation from natural language text. *ACM Transactions on Information Systems (TOIS)*, 40(1):1–43.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

Deyan Zou. 2022. Multi-dimensional consideration of cognitive effort in translation and interpreting process studies. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 416–426, Orlando, USA. Association for Machine Translation in the Americas.

A Appendix

A.1 Full Prompt Descriptions and Examples

Table 11 shows a sample instantiation for each prompt type and template. When this paper is accepted, we will provide all the prompt files and instantiated prompts for all experiments in our github: `https://github.com/aramir62/da-nlg`.

A.2 Semantic Accuracy Pseudo Metrics

We estimate the goodness of the pseudo versions of BLEU, Beyond-BLEU, BERT and BLEURT by examining their correlations with the domain-specific SACC scores on a sample of model outputs from our experiments, as shown in Table 12. The correlations show that the pseudo version of Beyond-BLEU (Wieting et al., 2019) – pBBLEU – performs the best across all three domains. Interestingly, pBLEU, despite BLEU's popularity, performs the worst.

A.3 Before & After Ranking

Our results show that ranking by any ranking function significantly and greatly improves performance, with the greatest performance improvements arising from the RF2_{DA} ranking function for all three domains. We calculate *Before Ranking* by averaging all metrics over the entire set of test outputs (test set size X 10 outputs into ranking). When taking averages across all experiments (per, fine-tuned, and specific), average SACC and DAC are significantly higher after ranking.

Table 13 provides more detail on how the ranking affects the results for few-shot fine-tuning. Comparing Row 1 to Row 4 shows that ranking improves the performance of SACC for 5-shot fine-tuning (85.71) to perform almost as well as 100-shot fine-tuning before ranking (88.71). Ranking also improves the performance of DAC for 100-shot fine-tuning from 57% to 80.56%, a huge improvement.

Table 14 shows more detail for Viggo across all the experimental settings. *Before Ranking* has an average of 65.29% versus *After Ranking* with an average of 86.82% for SACC. DAC has an almost a 30% increase where *Before Ranking* has an average of 62.11%, and *After Ranking* has an average of 91.04%. Table 15 shows the effect of ranking for TV and Laptop, illustrating a similarly large performance improvement due to ranking.

Prompt ID	Example
TST VANILLA	Here is a text: "Worms: Reloaded Steam". Rewrite of the text, which is a suggest dialogue act: "I bet you like it when you can play games on Steam, like Worms: Reloaded, right?"
TST DIALOGUE	Here is a text: "Worms: Reloaded Steam". Rewrite it to be a suggest dialogue act: "I bet you like it when you can play games on Steam, like Worms: Reloaded, right?"
TST PARAPHRASE	Here is a text: "I suggest Worms: Reloaded Steam". Paraphrase of the text: "I bet you like it when you can play games on Steam, like Worms: Reloaded, right?"
DEFINITIONAL	Description of $< suggest >$: A question asking if your friend has any experience with a certain type (based on data) of video games. Use the name of the game in data with 'such as', 'like', etc. The response should consist of a single yes/no question. Generate diverse responses. Data: suggest = yes | name = Worms: Reloaded | available_on_steam = yes. Data to Text for $< suggest >$: I bet you like it when you can play games on Steam, like Worms: Reloaded, right?
PARAPHRASE	I suggest a game Worms: Reloaded Steam. I bet you like it when you can play games on Steam, like Worms: Reloaded, right?
DIALOGIC	Can you suggest a game Worms: Reloaded Steam? I bet you like it when you can play games on Steam, like Worms: Reloaded, right?
PSEUDO	Suggest Worms: Reloaded Steam. I bet you like it when you can play games on Steam, like Worms: Reloaded, right?
S2S	suggest = yes | name = Worms: Reloaded | available_on_steam = yes. I bet you like it when you can play games on Steam, like Worms: Reloaded, right?

Table 11: Prompt IDs and Instantiation of each Prompt Template Type

Measure	ViGGO	Laptop	TV
pBLEU	0.08	-0.12	0.05
pBBLEU	0.52	0.32	0.45
pBLEURT	0.38	0.17	0.26
pBERT precision	0.33	0.14	0.36
pBERT recall	0.03	-0.06	0.14
pBERT F1	0.20	0.04	0.26

Table 12: Pearson correlation between SACC and common semantic preservation measures when applied to pseudo-references. All correlations are statistically significant at $p < 0.001$.

N	SACC		Perf		DAC	
	Before	After	Before	After	Before	After
5	65.57	85.71	9.10	38.88	21.10	54.44
25	76.01	92.19	16.39	62.22	31.10	79.72
50	86.70	96.43	29.10	71.94	42.00	79.44
100	88.71	97.74	40	78.61	57.00	80.56

Table 13: Few-shot fine-tuning performance with increasing training examples per DA - before and after ranking. DAC = DA accuracy.

Format	N	**Perfect**		**SACC**		**DAC**	
		Before	After	Before	After	Before	After
TST Vanilla	10	37.2	**85.6**	76	**94.7**	84.3	**100**
TST Dialogue	10	**39.5**	83.9	**76.7**	94.2	84.7	**100**
S2S	10	32.0	70.6	68.3	86.5	85	100
Pseudo	10	32	75.8	70.3	94.2	84.5	100
Definitional (each)	10	37.2	76.9	73.4	91.2	88.3	100
Definitional (Top)	10	38.2	82.2	72.3	93.5	**88.8**	100
TST Vanilla	5	38.7	**83.6**	76.8	92.6	76.9	98.7
TST Dialogue	5	**40.7**	**83.6**	**76.9**	**93.9**	79.1	100
S2S	5	34.1	66.7	65.5	83.5	77.9	98.7
Pseudo	5	14.7	52.2	47.5	82.6	47.2	88.6
Definitional (each)	5	40.2	80.0	75.1	92.7	81.9	99.4
Definitional (Top)	5	38.4	77.2	74	91.3	**82**	**100**
TST Vanilla	1	25.6	**69.2**	**69.3**	**88.2**	58	92
TST Dialogue	1	25.5	**69.2**	68.2	**88.2**	62.3	93.3
Definitional	1	**25.7**	63.9	67	85.3	**66.2**	**98.3**

Table 14: Results Before and After Ranking

Format	**N**	**SACC**		**Perf**		**DAC**	
		Before	**After**	**Before**	**After**	**Before**	**After**
TV	10	92.59	99.76	65.30	98.85	95.90	100
Laptop	10	80.73	95.90	36.35	80.95	99.71	100

Table 15: Laptop and TV Before and After ranking. DAC = DA Accuracy.