CONTROL PREFIXES for Parameter-Efficient Text Generation

Jordan Clive[1], **Kris Cao**[2], **Marek Rei**[1,3]
[1]Department of Computing, Imperial College London, United Kingdom
[2]DeepMind, London, United Kingdom
[3]ALTA Institute, University of Cambridge, United Kingdom
jordan.clive19@imperial.ac.uk, kriscao@deepmind.com, marek.rei@imperial.ac.uk

Abstract

Prefix-tuning is a parameter-efficient and powerful technique for adapting a pre-trained language model to a downstream application. However, it uses the same dataset-level tuned set of parameters for all examples in the dataset. We extend the framework with a dynamic method, CONTROL PREFIXES, which allows for the effective inclusion of input-dependent information, thereby demonstrating how prefix-tuning can be used for controlled text generation tasks. The method incorporates attribute-level learnable representations into different layers of a pre-trained Transformer, enabling the generated text to be guided in a particular direction. We provide a systematic evaluation of the technique and apply it to five datasets from the GEM benchmark for natural language generation (NLG). Using only 0.1–2% additional trainable parameters, we show CONTROL PREFIXES can even outperform full fine-tuning methods, and present state-of-the-art results on several data-to-text datasets, including WebNLG. We also examine the common case where input-dependent information is unavailable at test time and show CONTROL PREFIXES can excel in this setting also.

1 Introduction

Approaches in text generation have been dominated by adapting pre-trained language models (PLM) to various downstream tasks. As the scale of PLMs continue to climb, the cost of updating *all* the PLM parameters per task, and resultant overhead of entirely new parameter-sets per task becomes impractical. Furthermore, full fine-tuning has been shown to result in catastrophic forgetting where knowledge learnt from the pre-training task is lost and natural language understanding overwritten (Peters et al., 2019).

Recent work has demonstrated that it is possible to train these models by optimizing a negligible fraction (0.01-2%) of additional parameters while leaving the base PLM parameters unchanged (Houlsby et al., 2019; Lester et al., 2021). Such parameter-efficient transfer learning (PETL) can achieve performance comparable to fine-tuning. Prefix-tuning (Li and Liang, 2021), which trains a prefix of additional key-value pairs at each layer, and adapters (Rebuffi et al., 2017) are the two current most popular PETL methods. Another alternative is in-context learning (ICL) (Brown et al., 2020; Schick and Schütze, 2020), which supplies hand-written prompts and requires no gradient-based training. ICL has however shown to result in poor performance as the number of fine-tuning examples increases beyond a handful (Lester et al., 2021). We therefore believe that PETL methods provide a more promising direction for study.

A weakness of most PETL methods is that the same additional parameters are used for all examples within a single task. As yet, there has been little research exploring PETL methods that incorporate input-dependent parameters (Liu et al., 2021a) for finer-grained control. Our work closes this gap by introducing a novel framework which extends prefix-tuning and demonstrates the utility of controlling parameter-efficient learning for data-to-text tasks. The method uses multiple modular *control prefixes*, trained simultaneously, which can change alongside the input according to the guidance signal. These dynamic prefixes operate together with the static prefix parameters and allow for finer-grained control over the frozen PLM. The chosen attributes can provide additional context about the input, for example, the sub-domain of a data-to-text triple, or specify some aspect of the desired output, such as the target length for text simplification.

Controlled text generation aims to guide generation towards the desired attributes, by incorporating various types of guidance (e.g. highlighted phrases (Grangier and Auli, 2018)). Previous work has focused on directly updating all the existing model's

Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 363 - 382

parameters (Keskar et al., 2019) or using a discriminator to guide generation (Dathathri et al., 2020). Other methods aim to generate text with specific target qualities, independent of overall task performance (Yu et al., 2021). In contrast, our proposed method is designed for maximizing downstream task performance through controlled text generation, while also doing it in a way that is parameter-efficient and compatible with PETL.

The resulting parameter-efficient architecture outperforms previous approaches, many of them based on full fine-tuning, when evaluated on the WebNLG (Gardent et al., 2017), WebNLG+ 2020 (Castro Ferreira et al., 2020), DART (Radev et al., 2020) and E2E Clean (Dušek et al., 2019) data-to-text datasets using the official evaluation scripts. We also show that although these modular prefixes are formed from shared reparameterizations and operate at every layer, they provide a level of interpretability, as similar control prefix representations are learned by the model for semantically similar attribute labels. This fact allows us to employ a zero-shot technique to deal with the more common case in controlled generation, where attribute-level information is absent at inference time. In addition, we show the superiority of the architecture to an alternative architecture of introducing identical guidance signal into prefix-tuning. In total, we evaluate CONTROL PREFIXES on five popular datasets from the GEM benchmark (Gehrmann et al., 2021, 2022) for natural language generation and demonstrate the technique is easily extendable to new tasks.[1]

2 CONTROL PREFIXES

2.1 Background

To evaluate our architecture, we focus on the data-to-text generation task, where structured data (such as database fields or tuples from a knowledge graph) is transformed into natural language. The objective is to model the conditional probability $P(Y|X)$ with X representing the structured input and Y representing the tokenized output sequence. As is done for current state-of-the-art (SOTA) systems (Ribeiro et al., 2020; Radev et al., 2020), we linearize the structured table or graph input into a tokenized sequence. For example, with the WebNLG dataset, X is the linearized graph and Y is a lexicalization of this graph—descriptive text expressing all and only the information in the input. However, the data also contains additional information we can exploit: WebNLG is clustered semantically into 15 different subdomains, and we can use the subdomain of each example as an explicit input-dependent attribute for our model.

In this work, we experiment with T5-large (Raffel et al., 2020) and $\text{BART}_{\text{LARGE}}$ (Lewis et al., 2020) as the underlying pre-trained LMs with parameters ϕ. As we consider *fixed LM* methods, these parameters ϕ are always kept frozen. Both are Transformer encoder-decoder where decoding proceeds auto-regressively. They have been pre-trained with the denoising objective, so they are good candidates for the data-to-text task. They have also been employed by top performers in public challenges such as the WebNLG+ 2020 Challenge (Castro Ferreira et al., 2020).

2.2 Intuition

Using a frozen PLM that captures broad natural language understanding provides the model with a parameter-efficient starting point that already has capacity for linguistic fluency. Combining these frozen parameters with a trainable task representation for data-to-text allows the model to learn how to use the LM to lexicalize graphs. Moreover, introducing attribute-level parameters, such as the subdomain of the data-to-text input, allows us to guide the generation further into a required direction relevant to all inputs associated with that domain.

The general task-specific parameters can themselves adapt to the modular *control prefixes*, which change according to the guidance signal for each input X. CONTROL PREFIXES can therefore leverage input-level information while being a parameter-efficient tuning method.[2] For this work, we only consider discrete labels as attributes for the guidance signal.

2.3 Description

A prefix (Li and Liang, 2021) is a set of additional learned key-value pairs at every layer. Our model uses a general task prefix P_θ ("task-specific parameters") and also trains a set of control prefixes C_θ that change depending on the input ("attribute-level parameters"). This requires attribute-level information or guidance G, to indicate which control prefixes to be used while processing a given

[1] We open-source CONTROL PREFIXES at https://github.com/jordiclive/ControlPrefixes.

[2] We use the term parameter-efficient to denote methods that update <2% of a base LM's parameters.

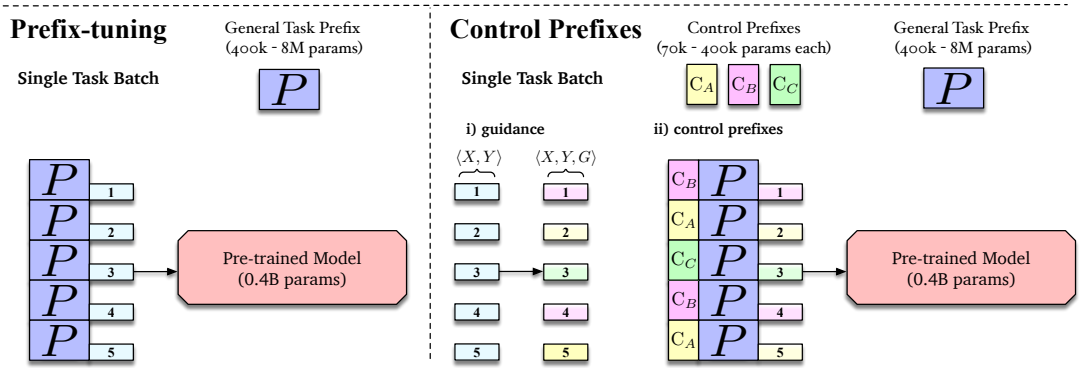


Figure 1: Prefix-tuning and CONTROL PREFIXES in the single-task setup for a PLM such as $\text{BART}_{\text{LARGE}}$. The same single-task batch (examples 1,2,3,4 and 5) is considered for both setups. Left: Prefix-tuning has one general prefix P for all examples. Right: CONTROL PREFIXES utilizes additional attribute information at the input-level, G, in **i)**. This conditional information is used in **ii)** to dictate which control prefix (C_A, C_B, C_C) to use for a particular example in a batch. This takes advantage of prefix-tuning's capacity to include different prefixes in one forward pass.

input X.[3] Let us consider the parallel corpus $\mathcal{Z} = \left\{\left\langle X^j, Y^j, G^j\right\rangle\right\}_{j=1,..,N}$, where G^j indicates all the conditional attribute-level information for the sample j. The goal is to optimize through gradient descent the final inference parameters, θ, whilst the underlying ϕ parameters of the pre-trained LM remain frozen:

$$\theta^* = \arg\max_{\theta} \sum_{j=1}^{N} \log\ p\left(Y^j \mid X^j, G^j; P_\theta, C_\theta, \phi\right). \tag{1}$$

Encoder-decoder We use d to represent the hidden state dimension and L the number of layers. We use (E, Dc, Ds) to denote the three classes of attention present in each layer: self-attention in the encoder (E), decoder cross-attention (Dc) and decoder self-attention (Ds). For an attention computation in the l-th layer, the query, key and value matrices are denoted $Q_l \in \mathbb{R}^{N\times d}$, and K_l, $V_l \in \mathbb{R}^{M\times d}$, where N is the number of tokens in the series relating to queries, and M is the number of tokens in the series relating to keys and values.

General Prefix For each attention class (E, Dc, Ds), a distinct prefix of key-value pairs is learnt, $P = \{P_1, \ldots, P_L\}$, where $P_l \in \mathbb{R}^{\rho\times 2d}\ \forall l \in \{1, \ldots, L\}$. $P \in \mathbb{R}^{\rho\times 2dL}$ and ρ is the prompt length, i.e. the number of additional key-value pairs in each attention computation. In prefix-tuning[4], for an attention computation in the l-th layer, K_l and V_l are augmented to become

$$K_l' = [P_{l,K}; K_l]\ , V_l' = [P_{l,V}; V_l] \tag{2}$$

where $K_l', V_l' \in \mathbb{R}^{(\rho+M)\times d}$. The overall general prefix, parameterized by θ, is $P_\theta = \left\{P^E, P^{Dc}, P^{Dm}\right\}$, where $P_\theta \in \mathbb{R}^{\rho\times 6dL}$.

Control Prefixes In addition to the general prefixes, we introduce control prefixes that change depending on the input attribute value. Let us consider one attribute, for example the domain of the input table (e.g. sports team, athlete etc.) with R possible values: $C_\theta = \{C_{\theta,1}, \ldots, C_{\theta,R}\}$, where $C_{\theta,r} \in \mathbb{R}^{\rho_c\times 6dL}, \forall r \in \{1 \ldots . R\}$. $C_{\theta,r}$ represents the control prefix learnt for the r-th attribute label and the parameter ρ_c denotes the control prompt length for this particular attribute.[5] Let $\mathcal{A}$ be a function which returns the corresponding control prefix for the attribute label indicated by G. Using CONTROL PREFIXES, the attention keys K_l and values V_l are augmented to become:

$$\begin{aligned} K_l'' &= [\mathcal{A}(G)_{l,K}; P_{l,K}; K_l]\ , \\ V_l'' &= [\mathcal{A}(G)_{l,V}; P_{l,V}; V_l] \end{aligned} \tag{3}$$

where $K_l'', V_l'' \in \mathbb{R}^{(\rho_c+\rho+M)\times d}$.

Shared Re-parameterization Li and Liang (2021) found that prefix optimization is stabilized by increasing the number of trainable parameters. This is achieved by introducing a feed-forward network to re-parameterize the prefix. Rather than one network, we use three distinct two-layered large

[3] We discuss cases where G is not present in §5.2.

[4] There has been confusion in recent work concerning different forms of prefix-tuning (Li and Liang, 2021). For details and observations of the benefits conferred by key-value pair prefix-tuning, see Appendix C.

[5] The method can be generalized to multiple attributes, each with control prefixes of different length.

feed-forward neural networks for each attention class, applied row-wise. For each attention class (E, Dc, Ds), $P = \text{MLP}(\tilde{P})$ where $\tilde{P} \in \mathbb{R}^{\rho \times d}$ is smaller than the matrix $P \in \mathbb{R}^{\rho \times 2dL}$, and each MLP has an intermediate dimension k which we set to 800. Once training is complete, the output of the MLP can be saved as the new prefix and the MLP parameters themselves can be discarded.

As described for the general prefix, P_θ, each control prefix, $C_{\theta,r}$, comprises three constituents for each attention class: $C_{\theta,r} = \{C_r^E, C_r^{Dc}, C_r^{Ds}\}$. The re-parameterization of $C_{\theta,r}$ occurs in the same manner as P_θ, sharing the same MLP^E, MLP^{Dc} and MLP^{Ds}. We found that using shared re-parameterization matrices provided performance improvements and led to more stable learning, while also significantly reducing the total number of parameters.

3 Experimental Setup

3.1 Datasets, Guidance and Metrics

Following Li and Liang (2021), we evaluate on the data-to-text datasets DART (Radev et al., 2020) and WebNLG (Gardent et al., 2017). In addition, we report results on E2E Clean (Dušek et al., 2019)[6], a dataset focused on the restaurant domain. The structured knowledge input in these datasets is in the form of a graph or table and can be linearized for sequence-to-sequence learning.

WebNLG contains graphs from DBPedia (Auer et al., 2007) and the dataset is clustered semantically into different categories. The test set is divided into two partitions: "Seen", which contains 10 DBpedia categories present in the training set, and "Unseen", which covers 5 categories never seen during training.[7] These categories, such as *Airport* or *Food* are used as a guidance attributes for CONTROL PREFIXES (indicated by A_1 in Table 1); our approach for the unseen categories is discussed in §5.2. The intuition of the category providing useful information is supported by studies showing a clear disparity in the performance of different model types between different categories (Moryossef et al., 2019; Castro Ferreira et al., 2020). By providing the category explicitly, the model is able to adjust its generation depending on the required target domain.

DART is an open-domain, multi-source corpus, with six sources: internal and external human annotation of both Wikipedia tables and WikiSQL, as well as the two existing datasets WebNLG and E2E Clean. Radev et al. (2020) showed fine-tuning T5-large on the WebNLG dataset with only the human annotated portion of DART achieves SOTA performance, whilst using the whole DART dataset is not as effective. Nevertheless, this inspired the idea of using the six DART sub-dataset sources as a controllable attribute, represented by A_2 in Table 1. This strategy was inspired by previous work which incorporates auxiliary scaffold tasks in multitask learning to improve span-labeling and text classification (Swayamdipta et al., 2018; Cohan et al., 2019; Cachola et al., 2020). For E2E Clean, which is itself a part of DART, our CONTROL PREFIXES model is trained on the additional components of the DART dataset with the explicit data source labels as guidance to act as scaffold framework.

CONTROL PREFIXES incorporates the attribute knowledge into a parameter-efficient architecture, giving it greater control over the generation process and allowing us to guide the output in a required direction. This provides a way of incorporating information about the data that would otherwise be left unused (such as the source domain of the input), or directing the generated output based on user preferences (for example by specifying the length of a simplified text).

We ensured that the attribute values used at inference time are permitted by all the shared task organizers corresponding to each dataset. In section §5.2 we also investigate settings where the attribute values are previously unseen or unavailable during inference. Also, note that additional training data is permitted by the organizers of the E2E Clean and WebNLG datasets. For example, the SOTA for WebNLG is a T5-large model fine-tuned on WebNLG and the human annotated portion of DART (Radev et al., 2020).

3.2 Metrics and Evaluation using GEM

We use the official evaluation scripts and report BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and TER (Snover et al., 2006) metrics[8]. In support of thorough NLG evaluation, we also report lexical similarity and diversity char-

[6] The same version as in GEM (Gehrmann et al., 2021).

[7] All the training category labels are visible in Appendix D, where we visualize control prefixes, corresponding to each training category.

[8] Additional evaluation script metrics, including machine-learned are found in Appendix A

acterization metrics, including machine-learned metrics, from the GEM (Gehrmann et al., 2021) evaluation suite in Appendix Tables 8,9.

Although we use data-to-text and text simplification datasets to demonstrate the technique is effective, it can be applied to any generation task cast as a sequence-to-sequence problem which similarly benefit form parameter-efficient control.

3.3 Training Details

We implement prefix-tuning and CONTROL PREFIXES for T5-large rather than GPT-2, as T5-large provides a stronger baseline and enables comparison with SOTA systems.[9] For the data-to-text datasets, we follow Ribeiro et al. (2020) and linearize the triples that form the input graph, prepending the special tokens <H>, <R>, and <T> before the subject, predicate, and object of an individual triple. The embeddings relating to these special tokens are the only embeddings we train, as our work is focused on fixed LM methods. We also prepend "translate Graph to English: " to every input (Raffel et al., 2020). We provide full training and hyperparameter details in Appendix E.

4 Data-to-Text Results

We indicate the guidance signal(s) used by each CONTROL PREFIXES model with A_1 for the WebNLG subdomain category and A_2 for the DART sub-dataset source.

Results in Table 1 show that for DART, both CONTROL PREFIXES (A_2) and prefix-tuning attain higher performance than the current SOTA, which is T5-large fined-tuned (Radev et al., 2020), by 1.29 and 0.54 BLEU points respectively. Note the results in the main body of the GEM paper (Gehrmann et al., 2021) are reported on the validation set rather than the test set as is done here.

The SOTA for WebNLG is a T5-large model fine-tuned on WebNLG and the human annotated portion of DART (Radev et al., 2020). Compared to this model, CONTROL PREFIXES achieves a 0.83 higher BLEU overall, and 1.33 on the Seen categories. Notably, CONTROL PREFIXES (A_1) outperforms CONTROL PREFIXES (A_1,A_2) on the Seen component of the dataset, but does not generalize as well to the unseen categories, indicating the benefit of using both controllable attributes. The prefix-tuning model with additional DART data, like the SOTA, is trained on only the human annotated portion and yields a minor performance increase of 0.05 BLEU compared to prefix-tuning solely trained on WebNLG. We believe this indicates that for fine-tuning, training on a complementary type of additional data allows the PLM to maintain more NLU by not over-fitting a narrow distribution, leading to better LM generalization. In contrast, for prefix-tuning, much of this gain has already been realized by retaining the original frozen parameters.

The SOTA (Harkous et al., 2020) for E2E Clean consists of a fine-tuned GPT-2 with a semantic fidelity classifier trained on additional generated data. CONTROL PREFIXES (A_2), which can leverage the heterogeneous DART datasets, outperforms this model in terms of the BLEU score. We also report results on the less popular WebNLG+ 2020 (Castro Ferreira et al., 2020) dataset (GEM), the second official WebNLG competition, in Appendix D.

5 Zero-shot Learning

5.1 Visualizing Control Prefixes

We experiment with visualizing the optimized control prefixes, in order to investigate what patterns they have learned. For this, we train a model for the task of text simplification, using the relative text compression rate as an attribute for the control prefix (additional details of this experiment in §7). Fig. 2 displays t-SNE (Maaten and Hinton, 2008) visualizations of the learned control prefix parameters in the decoder self-attention. A clear monotonic pattern emerges, showing that control prefixes for similar compression rate values are close to each other in the representation space. This property can be useful for investigating different attributes or inferring representations for unseen attribute values. In Appendix F we present additional graphs for control prefixes in the encoder and the cross-attention of the model.

5.2 Unseen WebNLG Categories

The control prefix parameters are optimized during training for each attribute value. However, in some settings we may need to handle attribute values that were not present in the training data and therefore have no matching control prefixes available. For example, the category attributes in the WebNLG *Unseen* subset are all novel and were not repre-

[9] $\text{BART}_{\text{LARGE}}$ exhibits inferior performance to T5-large on data-to-text; for example, 9.7 BLEU points lower on WebNLG Unseen (Ribeiro et al., 2020).

	ϕ%	DART			ϕ%	WebNLG			ϕ%	E2E Clean	
		BLEU	METEOR	TER ↓		S	U	A		BLEU	METEOR
T5-large fine-tuned	100	50.66	40	43	100	64.89	54.01	59.95	100	41.83	38.1
SOTA	100	50.66	40	43	100	65.82	56.01	61.44	100	43.6	39
Prefix-tuning	1.0	51.20	40.62	43.13	1.0	66.95	55.39	61.73	1.0	43.66	39.0
CONTROL PREFIXES (A_1)	-	-	-	-	1.4	**67.32**	55.38	61.94	-	-	-
+Data: DART											
Prefix-tuning	1.0	51.20	40.62	43.13	1.0	67.05	55.37	61.78	1.0	43.04	38.7
CONTROL PREFIXES (A_2)	1.1	**51.95**	**41.07**	**42.75**	1.0	66.99	55.56	61.83	1.0	**44.15**	**39.2**
CONTROL PREFIXES (A_1,A_2)	-	-	-	-	1.4	67.15	**56.41**	**62.27**	-	-	-

Table 1: Data-to-text test set results reported on the respective official evaluation scripts. ϕ% denotes the trainable parameters as a % of the fixed-LM parameters required at inference time. T5-large fine-tuned results for WebNLG are from Ribeiro et al. (2020) and for DART are from Radev et al. (2020). Several of the baseline results were only reported to the significant figures shown. A_1 signifies models trained with control prefixes for the *WebNLG category* attribute, and A_2 with control prefixes for the DART *sub-dataset source* attribute. For WebNLG, S, U and A refer to BLEU scores for the *Seen*, *Unseen* and *All* portions of the dataset. The DART results are reported on the official evaluation script for v1.1.1, the same version as the official leaderboard. A CONTROL PREFIXES model attains state-of-the-art results for each dataset.

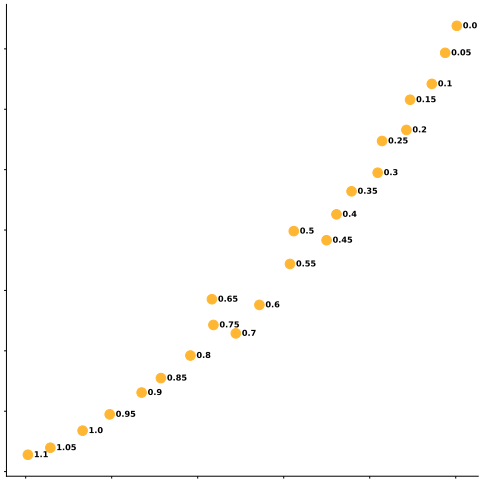


Figure 2: t-SNE visualizations for the decoder self-attention constituent of the simplification model's length compression control prefixes. Each circle represents a control prefix corresponding to each length ratio (bins of fixed width 0.05, from 0 to 1.1).

sented in the training set. While no suitable control prefixes exist for these categories, they each have a textual label available in the dataset. Experiments in Fig. 2 also established that similar attribute labels learn similar parameter values in their control prefixes. This gives us some prior on the properties of the unseen categories, which we show is enough to perform zero-shot transfer with control prefixes.

We first map the textual label of each WebNLG category to a Glove embedding(Pennington et al., 2014).[10] Then for each *Unseen* category, we find

[10]Glove Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors).

	OOV	Zero-shot
WebNLG	56.35	**56.41**
WebNLG+ 2020	50.02	**50.39**

Table 2: A comparison of the BLEU performance on the *Unseen* portions for WebNLG test sets, with i) a single OOV Control Prefix used for all samples from unseen categories, or ii) the zero-shot transfer approach outlined, utilizing the available textual labels.

the *Seen* category with the highest cosine similarity in the embedding space, and use its learned control prefix to also represet the corresponding *Unseen* category. For example, the control prefix for the seen category *SportsTeam* is used for examples relating to the unseen category *Athlete*.[11]

Table 2 shows results for the zero-shot transfer method on both WebNLG datasets. For comparison, we also report results using a single out-of-vocabulary (OOV) control prefix for all unseen categories. This OOV control prefix is trained by randomly selecting 2% of the categories in each training batch and replacing them with a general OOV category. These results indicate that zero-shot transfer based on word embeddings and textual labels provides an advantage over learning a single OOV representation.

[11]Appendix I displays model output for WebNLG along with the zero-shot procedure.

	ϕ%	ASSET		TurkCorpus	
		SARI	FKGL ↓	SARI	FKGL ↓
Gold Reference	-	44.87	6.49	40.04	8.77
$\text{BART}_{\text{LARGE}}$ with ACCESS†	100	43.63	6.25	42.62	6.98
$\text{BART}_{\text{LARGE}}$ fine-tuned	100	39.91*	7.73*	39.55*	7.73*
Prefix-tuning	1.8	40.12	7.28	39.06	**7.28**
CONTROL PREFIXES	1.8	**43.58**	**5.97**	**42.32**	7.74

Table 3: Simplification results on ASSET and TurkCorpus test sets. †This model is from Martin et al. (2020), where the authors fine-tuned $\text{BART}_{\text{LARGE}}$ model alongside control tokens for the four attributes. The CONTROL PREFIXES model is trained with control prefixes for these same four attributes. Prefix-tuning and CONTROL PREFIXES use $\text{BART}_{\text{LARGE}}$ as the fixed LM. The * denotes baseline results calculated in this study—the model outputs of Martin et al. (2020) are publicly available. The $\text{BART}_{\text{LARGE}}$ with ACCESS and CONTROL PREFIXES model are the average test set results over 5 random seeds. We bold the best results of parameter-efficient models in the results tables, while fully fine-tuned models and human performance are reported for reference.

6 Token-level control

For comparison, we also investigated a simpler architecture: prefix-tuning combined with control tokens (Keskar et al., 2019). In this setting, the model receives the same guidance signals as CONTROL PREFIXES, but instead uses trainable control tokens for representing the attribute values. The main model is kept frozen, while the general prefix is optimized along with embeddings for the control tokens, allowing us to benchmark against a different parameter-efficient architecture. Note, we chose to compare against a prefix-tuning based architecture as the fully fine-tuned models lag behind prefix-tuning in Table 1.

The results for this experiment are included in Appendix G. We found that CONTROL PREFIXES consistently outperformed control tokens on all three data-to-text datasets. This indicates that CONTROL PREFIXES is a superior parameter-efficient framework for leveraging additional information, whilst maintaining the *fixed-LM* property. Control tokens lack the shared re-parameterization of static and dynamic parameters. They are only able to inject information at the embedding level, making them less expressive than the CONTROL PREFIXES method.

CONTROL PREFIXES fundamentally depends on the strength of the guidance signal. This is reflected in the constraint of attribute information being available with the dataset. However, we show that CONTROL PREFIXES is a powerful general method which can utilize this signal to achieve a consistent improvement across an array of tasks.

7 Applicability to other tasks

Finally, we investigate the application of CONTROL PREFIXES to generation tasks beyond the data-to-text setting. For these experiments, we integrate the method with a sequence-to-sequence model trained for text simplification on the WikiLarge (Zhang and Lapata, 2017) dataset. Following Martin et al. (2020), the model uses four simplification-specific attributes as control prefixes: the length compression ratio, replace-only Levenshtein similarity, aggregated word frequency ratio and dependency tree depth ratio. [12]

In Table 3 we report SARI (Xu et al., 2016) and FKGL (Kincaid et al., 1975) metrics.[13] For comparison, we report results for $\text{BART}_{\text{LARGE}}$ with ACCESS (Martin et al., 2020), which is a fully fine-tuned model that also integrates the same four attributes but uses control tokens instead. The results show that CONTROL PREFIXES is able to outperform the fully fine-tuned BART on the task of simplification, even though it optimizes only 1.8% of the parameters. When compared to $\text{BART}_{\text{LARGE}}$ with ACCESS, the results for CONTROL PREFIXES are competitive while still being substantially more parameter-efficient. Note this model has the benefit of full fine-tuning and we are already at maximum performance for the datasets as assessed by these metrics. This is indicated by the *Gold Reference* scores, which evaluates against other human annotators.

[12] Refer to Martin et al. (2020) for full attribute details.

[13] We use the FKGL and the latest version of SARI implemented in EASSE (Alva-Manchego et al., 2019) which is also used by Martin et al. (2020).

8 Related Work

Controlled generation aims to incorporate various types of guidance beyond the input text into the generation model (Kikuchi et al., 2016). Johnson et al. (2016) trained a translation model with control tokens to encode each language, and Keskar et al. (2019) pre-trained a 1.63B parameter model, alongside conditional control tokens demonstrating these learnt to govern stylistic aspects. In addition to having the benefit of updating all model parameters, these methods only act at the embedding level.

Alternatives exist, such as using a plug-and-play mechanism to perturb the LM hidden states towards a target attribute (Dathathri et al., 2020). Strategies such as these are computationally intensive, resulting in a slow generation speed and the shift in conditional probability has been shown to increase text degeneration (Holtzman et al., 2020; Gehman et al., 2020). GSum (Dou et al., 2020) is an example of work that has explored using learned guidance prediction models at test time. However, both the prompt and LM parameters are tuned.

There has been little work using controlled generation in the data-to-text domain. Su et al. (2021) were able control both the intra-text sentence and inter-sentence structure of generated output. This architecture exhibits inferior performance to our method on the mutual evaluation dataset WebNLG. Additionally, CONTROL PREFIXES uses fewer additional parameters and can incorporate multi-attribute control with prefixes of varying sizes.

Several successive works (Logeswaran et al., 2020; Liu et al., 2021b; Lester et al., 2021) employ prompt tuning, where unlike the discrete text prompts in ICL, trainable soft embeddings are prepended to the input. Again the technique acts only at the embedding level, thus limiting any control that can be exerted from data-point guidance. This shortcoming also exists with ICL and multi-task prompting (Sanh et al., 2021; Qin and Eisner, 2021). Prefix-tuning is more expressive and, along with Vedd et al. (2021), serves as inspiration for this work. However, prefix-tuning trains each prefix separately and no relationship between prefixes is modelled. The parameters are static with no mechanism to incorporate guidance.

There have been few works exploring input-dependent parameters trained alongside static prompt parameters (Liu et al., 2021a). Perhaps most similar to our work is Yu et al. (2021), who use an attribute alignment function to encode tokens of attributes. Unlike our work, there are no dedicated task parameters and the method aims to generate text with specific target attributes, independent of task performance. With CONTROL PREFIXES, the intention is to also maximize task-specific performance, which is why we maintain a large static component to specify the task itself, which is directly learnt simultaneously with the dynamic parameters in a shared framework.

9 Conclusion

We have proposed CONTROL PREFIXES, a general framework for integrating attribute-level information into pre-trained language models. In addition to the general prefix for the overall task, special prefixes are optimized for each attribute value and incorporated into different levels of the Transformer. This allows for finer-grained control over generated text, either by providing additional context about each input example or by allowing the user to specify some aspect of the desired output. The main language model parameters are kept frozen while only the multiple prefixes are optimized for a particular task, providing a very parameter-efficient method.

Our experiments show that CONTROL PREFIXES outperforms all existing methods for several data-to-text tasks including WebNLG and DART. This is in spite of learning less than 2% of the base LM's parameters and using signal from attribute level information that is available for the tasks. CONTROL PREFIXES also achieves higher results when compared to an alternative prefix-tuning architecture that makes use of the same attribute-level information, showing that the proposed framework is better able to integrate the additional signals with the rest of the model.

We also saw that the method can still be applied when suitable prefixes do not exist for a particular attribute value, by constructing the required prefix based on semantic similarity. Experiments on text simplification also verified that CONTROL PREFIXES can be applied on other tasks and datasets beyond the data-to-text setting.

In future work, additional guiding attributes can be investigated for text generation, such as the desired formality and sentiment. In addition, this method can be integrated with a wider range of model architectures, beyond text generation applications, that require parameter-efficient methods of control.

10 Limitations & Ethical Impact

Evaluating NLG is notoriously challenging (Celikyilmaz et al., 2020). For example, Freitag et al. (2020) and Mathur et al. (2020) find that when comparing two high-quality systems, differences according to a metric may also reflect how the references are written or flaws in the metric itself. To combat this, in addition to using the official task evaluation scripts, we report an array of GEM Gehrmann et al. (2021) metrics that represent lexical similarity and semantic equivalence in Table 8. We are also conscious that NLG models intrinsically trade off diversity and quality. We therefore report diversity and system characterization results in Table 9.

The technique described requires data-point information in the form of discrete categorical variables. Future work would look to investigate how best to integrate continuous information. In addition, as highlighted throughout CONTROL PREFIXES fundamentally depends on the strength of the guidance signal. The success of the zero-shot procedure depends on how well the semantic category labels are written for the unseen categories.

The technique is also limited by the predictive capabilities of the base frozen language model. One benefit, however, is that optimizer states for the base language model do not need to be stored during training, making training more computationally efficient.

We acknowledge that biases pose a huge problem in the Machine Learning and NLP community. We conducted experiments with BART and T5. Both models are trained on large amounts of textual data such as news, books, and web text, which may contain any kinds of biases. Although our research is conducted under the purview of parameter efficient NLP methods, we still used up to 6 V100-SXM2-16GB GPUs. There is a responsibility for the considerable CO2 emissions in the NLP community and for developing more resource-efficient training and inference methods.

References

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. Easse: Easier automatic sentence simplification evaluation. *arXiv preprint arXiv:1908.04567*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

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. *CoRR*, abs/2005.14165.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S. Weld. 2020. TLDR: extreme summarization of scientific documents. *CoRR*, abs/2004.15011.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *CoRR*, abs/2006.14799.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. *CoRR*, abs/1904.01608.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. Gsum: A general framework for guided neural abstractive summarization. *CoRR*, abs/2010.08014.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proc. of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. *CoRR*, abs/2004.06063.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on*

Natural Language Generation, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur P. Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672.

Sebastian Gehrmann, Abhik Bhattacharjee, Abinaya Mahendiran, Alex Wang, Alexandros Papangelis, Aman Madaan, Angelina McMillan-Major, Anna Shvets, Ashish Upadhyay, Bingsheng Yao, Bryan Wilie, Chandra Bhagavatula, Chaobin You, Craig Thomson, Cristina Garbacea, Dakuo Wang, Daniel Deutsch, Deyi Xiong, Di Jin, Dimitra Gkatzia, Dragomir Radev, Elizabeth Clark, Esin Durmus, Faisal Ladhak, Filip Ginter, Genta Indra Winata, Hendrik Strobelt, Hiroaki Hayashi, Jekaterina Novikova, Jenna Kanerva, Jenny Chim, Jiawei Zhou, Jordan Clive, Joshua Maynez, João Sedoc, Juraj Juraska, Kaustubh Dhole, Khyathi Raghavi Chandu, Laura Perez-Beltrachini, Leonardo F. R. Ribeiro, Lewis Tunstall, Li Zhang, Mahima Pushkarna, Mathias Creutz, Michael White, Mihir Sanjay Kale, Moussa Kamal Eddine, Nico Daheim, Nishant Subramani, Ondrej Dusek, Paul Pu Liang, Pawan Sasanka Ammanamanchi, Qi Zhu, Ratish Puduppully, Reno Kriz, Rifat Shahriyar, Ronald Cardenas, Saad Mahamood, Salomey Osei, Samuel Cahyawijaya, Sanja Štajner, Sebastien Montella, Shailza, Shailza Jolly, Simon Mille, Tahmid Hasan, Tianhao Shen, Tosin Adewumi, Vikas Raunak, Vipul Raheja, Vitaly Nikolaev, Vivian Tsai, Yacine Jernite, Ying Xu, Yisi Sang, Yixin Liu, and Yufang Hou. 2022. Gemv2: Multilingual nlg benchmarking in a single line of code.

David Grangier and Michael Auli. 2018. QuickEdit: Editing text & translations by crossing words out. 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 272–282, New Orleans, Louisiana. Association for Computational Linguistics.

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. *CoRR*, abs/2004.06577.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.

Wendell Johnson. 1944. Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.

N. Keskar, B. McCann, L. R. Varshney, Caiming Xiong, and R. Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

J. Peter Kincaid, Robert P Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels

of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691.

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

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

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

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. GPT understands, too. *CoRR*, abs/2103.10385.

Lajanugen Logeswaran, Ann Lee, Myle Ott, Honglak Lee, Marc'Aurelio Ranzato, and Arthur Szlam. 2020. Few-shot sequence learning with transformers. *CoRR*, abs/2012.09543.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

L. V. D. Maaten and Geoffrey E. Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020. Multilingual unsupervised sentence simplification. *CoRR*, abs/2005.00352.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Amit Moryossef, Ido Dagan, and Yoav Goldberg. 2019. Improving quality and efficiency in plan-based neural data-to-text generation.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Nivranshu Pasricha, Mihael Arcan, and Paul Buitelaar. 2020. NUIG-DSI at the WebNLG+ challenge: Leveraging transfer learning for RDF-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 137–143, Dublin, Ireland (Virtual). Association for Computational Linguistics.

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

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying lms with mixtures of soft prompts. *CoRR*, abs/2104.06599.

Dragomir R. Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Nazneen Fatema Rajani, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Murori Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, and Richard Socher. 2020. DART: open-domain structured data record to text generation. *CoRR*, abs/2007.02871.

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(140):1–67.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30, pages 506–516. Curran Associates, Inc.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2020. Investigating pretrained language models for graph-to-text generation. *arXiv*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. Multitask prompted training enables zero-shot task generalization. *CoRR*, abs/2110.08207.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *CoRR*, abs/2009.07118.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and Ralph Weischedel. 2006. A study of translation error rate with targeted human annotation. In *In Proceedings of the Association for Machine Transaltion in the Americas (AMTA 2006*.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. *CoRR*, abs/2108.13740.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *EMNLP*.

Nihir Vedd, Zixu Wang, Marek Rei, Yishu Miao, and Lucia Specia. 2021. Guiding visual question generation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Dian Yu, Kenji Sagae, and Zhou Yu. 2021. Attribute alignment: Controlling text generation from pretrained language models. *CoRR*, abs/2103.11070.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

A Additional Results

Additional results using the official evaluation scripts for the data-to-text datasets are reported in Tables 4,5,6 to supplement the results in Table 1.

B GEM Automatic Evaluation

Supporting results using the GEM package for model evaluation (https://github.com/GEM-benchmark/GEM-metrics) are provided in Tables 8,9.

C Prefix-tuning

We make two previously unremarked upon observations of the benefits conferred by using the key-value pair prefix-tuning described in §2.3 compared to prefix-tuning involving augmenting the activations directly (Hu et al., 2021) or prompt-embedding tuning of prompt length ρ. i) The form discussed does not restrict the input length of the base LM. ii) The time complexity at inference time is reduced; for example, if we take a multi-head self-attention computation ($M = N$), the time complexity at inference time is $\mathcal{O}((N+\rho)Nd + Nd^2)$ rather than the greater $\mathcal{O}((N+\rho)^2 d + (N+\rho)d^2)$.

D WebNLG+ 2020 Results

WebNLG+ 2020 is not a component of DART—it was used for the second official WebNLG competition (Castro Ferreira et al., 2020). There are 16 training categories (the 15 categories from WebNLG, but with new examples), alongside 3 unseen categories. Table 7 displays WebNLG+ 2020 results using the same model architectures as used for WebNLG. A similar pattern is revealed, in that CONTROL PREFIXES outperforms prefix-tuning with CONTROL PREFIXES (A_1,A_2) as the top-performing model. This illustrates again the benefit of using both controllable attributes.

In the WebNLG and WebNLG+ 2020 training sets, for the same tripleset, multiple distinct lexicalizations exist. In our experiments, the examples sharing identical tripleset inputs have the same triple order after linearization. This is to aid in comparison with current systems for WebNLG, DART and E2E Clean. Permuting the triples for these examples will introduce a source of randomness for result comparison.

(a) WebNLG

(b) WebNLG+ 2020

Figure 3: t-SNE visualizations for the encoder constituent of control prefixes representing WebNLG categories seen during training. Each circle represents a category seen during training for the CONTROL PREFIXES ($A1$) model. All 15 categories are seen categories in WebNLG+ 2020, along with the category *Company*. WebNLG+ 2020 has 3 additional unseen categories to those shown.

E Additional Training Details

All implementations in this study are built on top of the Transformers library (Wolf et al., 2020). As T5 has relative position biases, we set these in all layers pertaining to offsets where the key is part of a prefix to zero. For $\text{BART}_{\text{LARGE}}$ we adapt the original implementation (Li and Liang, 2021). Table 11 displays the hyperparameters used when training the models reported in this paper.

The general prompt length and each control prompt length are architecture-specific parameters that we choose based on performance on the validation set. We use gradient accumulation across batches to maintain an effective batch size above 64,

a linear learning rate scheduler for all models and beam-search decoding. AdamW (Loshchilov and Hutter, 2017) and AdaFactor (Shazeer and Stern, 2018) were used for optimization. We chose the checkpoint with the highest validation score using BLEU for data-to-text and SARI for simplification. For all tasks, we train our models on single Tesla V100-SXM2-16GB machines, with mixed precision for $\text{BART}_{\text{LARGE}}$ based models (fp16) and full precision for T5-large based models (fp32).

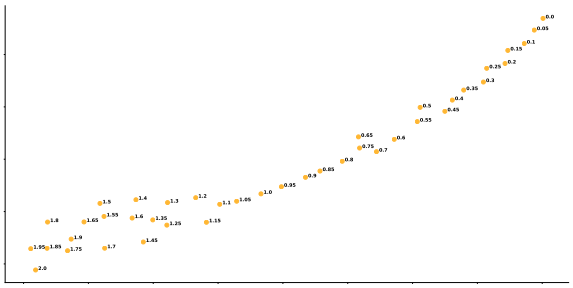
The CONTROL PREFIXES models with the DART *sub-dataset source* attribute (A_2) use DART as additional data and were trained in two stages: i) on DART, ii) solely on the downstream dataset. The WebNLG prefix-tuning model with DART data shown in Table 11 uses only the human annotated portion of DART. The prefix-tuning models using all of the DART data for WebNLG and E2E Clean were similarly trained in two stages, with identical hyperparameters to CONTROL PREFIXES models using A_2. Training prefix-tuning on all of DART for WebNLG yielded lower performance than with only the human-annotated DART portion as additional data, so was not reported in Table 1.

Decoding specific parameters were not tuned—we instead mirrored what the top-performing fine-tuned based system used for the particular LM and dataset. For example, a beam width of 5 as in Ribeiro et al. (2020) for T5-large on all data-to-text datasets.

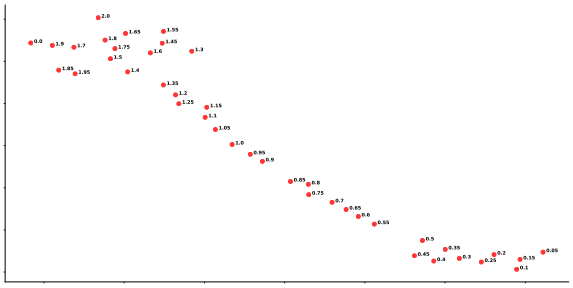
F Simplification Length Control

Figure 4: Histogram illustrating the influence of different target length ratios on the actual length compression ratio output distribution for the simplification CONTROL PREFIXES model on the TurkCorpus validation set.

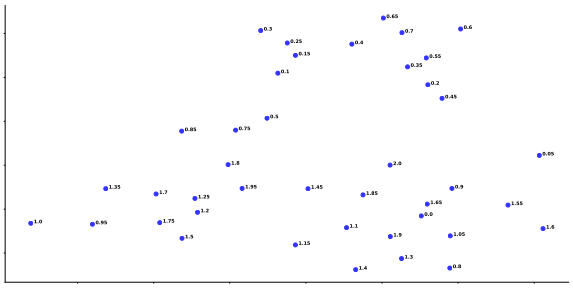
Fig. 4 depicts the length compression ratio output distribution on the validation set for CONTROL PREFIXES, where a length control prefix of a specific attribute value (0.25,0.5,0.75,1.0) is specified. This clearly demonstrates CONTROL PREFIXES is capable of controlling the target length with respect to the input. Table 12 displays example output generations with each of the 0.25,0.5,0.75,1.0 values specified.



(a) Decoder Masked-attention (Dm)



(b) Encoder (E)



(c) Decoder Cross-attention (Dc)

Figure 5: t-SNE visualizations for constituents of the length compression control prefixes learnt as part of the simplification CONTROL PREFIXES model. Each diagram depicts representations of control prefixes corresponding to each length value (41 bins of fixed width 0.05, from 0 to 2) for a particular attention mechanism. The dimension represented on the x-axis is stretched from a 1:1 to 2:1 aspect ratio for labelling clarity.

Fig. 5 is supplementary to §5.1, showing all constituents of the length compression control prefixes for all attribute values. In the WikiLarge training data, there are far fewer training samples where the simplified output is much longer than the complex, original input in WikiLarge. This explains why the representations are not as interpretable for values greater than 1.2.

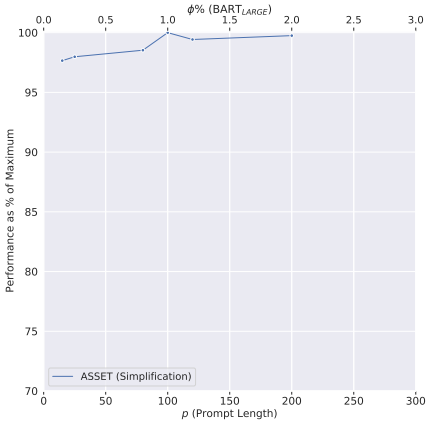
G Prefix-tuning + Control Tokens

We propose another architecture 'prefix-tuning + control tokens', where all of the original LM parameters, ϕ, still remain fixed, including the embedding matrix. Control has to be exerted through the few control embeddings and prefix-tuning's ability to steer the frozen ϕ parameters through $< 2\%$ additional parameters. We use this method to inform the model of the same discrete guidance information as in CONTROL PREFIXES, but with control tokens instead of control prefixes.[14] This alternative method is less expressive than CONTROL PREFIXES, in much the same way as prefix-tuning is more expressive than prompt-embedding tuning. Prefix-tuning + control tokens also does not benefit from the shared re-parameterizations (§2.3) that we argue allow for more effective demarcation of control of the fixed LM in each attention class subspace.

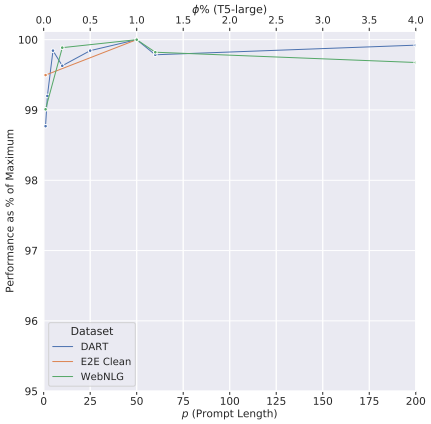
Table 10 reveals that CONTROL PREFIXES outperforms prefix-tuning + control tokens on the data-to-text datasets, while the results are both comparable to the *Gold References* on simplification datasets. This indicates that CONTROL PREFIXES is better able to integrate and leverage guidance signal at the input-level, whilst maintaining the *fixed-LM* property, than prefix-tuning + control tokens.

H Varying Prompt Length

Our research is not solely focused on parameter efficiency, but also on the effectiveness of adapting an already parameter efficient, fixed-LM method (adding <2% additional parameters). The only way to add parameters with prefix-tuning is to increase the prompt length. XSum is the only dataset considered where performance does not plateau when increasing prompt length[15], therefore we ensure CONTROL PREFIXES does not have more parameters than prefix-tuning to ensure a fair comparison. Fig. 6 illustrates how performance saturation is observed—after a certain prompt length performance plateaus. Different datasets require varying prompt lengths to attain near maximum performance in a parameter search for prompt length. For the data-to-text datasets, near maximum performance (>99% of the maximum validation score in the search) is reached with a prompt length of 1 or 2.



(a) $\text{BART}_{\text{LARGE}}$



(b) T5-large

Figure 6: Prefix-tuning results of a model parameter search on several datasets for the optimal prompt length per dataset. These results are for the metric monitored per task on the respective validation sets indicated in the legend. $\phi\%$ denotes the % of additional parameters to the number of fixed-LM parameters required at inference time. The y-axis is a relative measure: the validation set performance as a % of the maximum attained in the parameter search.

I Qualitative Examples

For data-to-text, Table 14 displays example CONTROL PREFIXES output for WebNLG input belonging to unseen categories, along with the zero-shot procedure. Table 14 depicts example CONTROL PREFIXES (A_1,A_2) output alongside prefix-tuning model output for WebNLG+ 2020 input. For simplification, Table 13 compares the fixed-LM guided generations of CONTROL PREFIXES to the fine-tuned $\text{BART}_{\text{LARGE}}$ with ACCESS (Martin et al., 2020).

[14] Only the embeddings pertaining to the controllable attributes and the prefix are trained.

[15] We do not observe performance degradation, such as described by Hu et al. (2021), when utilizing different forms of prefix-tuning. This is shown in H.

	ϕ%	BLEU	METEOR	TER ↓	BERTScore(F1)
T5-large fine-tuned*	100	50.66	40	43	0.95
Prefix-tuning	1.0	51.20	40.62	43.13	0.95
CONTROL PREFIXES (A_1)	1.1	**51.95**	**41.07**	**42.75**	0.95

Table 4: Detailed results on the DART test set to complement Table 1. T5-large fine-tuned is the current SOTA (Radev et al., 2020). We report results on the official evaluation script for v1.1.1, the same version as the official leaderboard, available here: https://github.com/Yale-LILY/dart. *Results for this model were only reported to the significant figures shown. ϕ% denotes the % of additional parameters to the number of fixed-LM parameters required at inference time.

	ϕ%	BLEU			METEOR			TER ↓		
		S	U	**A**	S	U	**A**	S	U	**A**
T5-large	100	64.89	54.01	59.95	46	43	44	34	41	37
SOTA	100	65.82	56.01	61.44	46	43	45	32	38	35
Prefix-tuning	1.0	66.95	55.39	61.73	46.73	42.71	44.87	31.34	39.01	34.86
CONTROL PREFIXES (A_1)	1.4	**67.32**	55.38	61.94	46.78	42.77	44.92	30.96	39.01	34.65
+Data: DART										
Prefix-tuning	1.0	67.05	55.37	61.78	46.69	42.82	44.90	31.36	38.79	34.77
CONTROL PREFIXES (A_2)	1.0	66.99	55.56	61.83	46.67	42.87	44.91	31.37	38.53	34.65
CONTROL PREFIXES (A_1,A_2)	1.4	67.15	**56.41**	**62.27**	46.64	43.18	45.03	31.08	38.78	34.61

Table 5: Detailed results on the WebNLG test set to complement Table 1. S, U and A refer to the *Seen*, *Unseen* and *All* portions of the WebNLG dataset. Several of the baseline results were only reported to the significant figures shown.

	ϕ%	BLEU	NIST	METEOR	R-L	CIDEr
T5-large	100	41.83	6.41	0.381	56.0	1.97
SOTA	100	43.6	-	0.39	**57.5**	2.0
Prefix-tuning	1.0	43.66	6.51	0.390	57.2	2.04
+Data: DART						
Prefix-tuning	1.0	43.04	6.46	0.387	56.8	1.99
CONTROL PREFIXES (A_2)	1.0	**44.15**	**6.51**	**0.392**	57.3	**2.04**

Table 6: Detailed results on the E2E Clean test set to complement Table 1. The SOTA baseline result was only reported to the significant figures shown.

	ϕ%	BLEU	METEOR	chrF++	TER ↓	BLEURT
T5-large*†	100	51.74	0.403	0.669	0.417	0.61
Prefix-tuning	1.0	54.74	0.417	0.693	0.399	0.62
CONTROL PREFIXES (A_1)	1.6	54.97	0.417	0.693	0.398	0.62
+Data: DART						
CONTROL PREFIXES (A_2)	1.0	54.92	0.418	0.695	0.397	0.62
CONTROL PREFIXES (A_1,A_2)	1.6	**55.41**	**0.419**	**0.698**	**0.392**	**0.63**

Table 7: **WebNLG+ 2020.** The overall WebNLG+ 2020 test set results using the official evaluation script. *As the model outputs are publicly available, we are able to run evaluation to achieve the same precision. †Results from Pasricha et al. (2020), who before fine-tuning on the WebNLG+ data, further pre-train T5-large using a Mask Language Modelling objective (with 15% of the tokens masked) on the WebNLG corpus and a corpus of DBpedia. A_1 signifies models trained with control prefixes for the *WebNLG category* attribute, and A_2 with control prefixes for the DART *sub-dataset source* attribute.

Dataset	Model	Metrics (Lexical Similarity and Semantic Equivalence)						
		METEOR	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	BLEURT
DART	Prefix-tuning	0.405	76.7	53.0	61.7	50.2	0.95	0.32
	CONTROL PREFIXES (A_2)	0.410	77.3	53.7	62.4	51.1	0.96	0.33
E2E Clean	Prefix-tuning	0.385	74.5	48.3	55.8	43.7	0.95	0.23
	CONTROL PREFIXES (A_2)	0.387	74.4	48.4	55.9	44.1	0.95	0.23
WebNLG 2017	Prefix-tuning	0.443	81.1	59.6	67.8	60.3	0.96	0.43
	Prefix-tuning + DART	0.443	81.2	59.8	67.8	60.4	0.96	0.43
	CONTROL PREFIXES (A_1)	0.443	81.3	59.9	67.9	60.5	0.96	0.43
	CONTROL PREFIXES (A_2)	0.443	81.3	59.8	68.1	60.5	0.96	0.43
	CONTROL PREFIXES (A_1,A_2)	0.444	81.4	60.0	68.0	60.8	0.96	0.43
WebNLG+ 2020	Prefix-tuning	0.417	79.6	56.2	64.8	56.2	0.96	0.32
	CONTROL PREFIXES (A_1)	0.417	79.5	56.3	65.1	56.3	0.96	0.32
	CONTROL PREFIXES (A_2)	0.418	79.6	56.5	65.3	56.4	0.96	0.33
	CONTROL PREFIXES (A_1,A_2)	0.419	80.0	56.9	65.4	56.8	0.96	0.34

Table 8: The set of additional lexical similarity and semantic equivalence results on the official Data-to-text test sets. These metrics are proposed by Gehrmann et al. (2021) and calculated using the GEM evaluation suite. The hash for BERTScore used is `roberta-large_L17_no-idf_version=0.3.8(hug_trans=3.0.1)` and for BLEURT the version is BLEURT-base-128.

Dataset	Model	Metrics (Diversity and System Characterization)								
		MSTTR	Distinct$_1$	Distinct$_2$	H_1	H_2	Unique$_1$	Unique$_2$	$\|\mathcal{V}\|$	Output Len.
DART	Prefix-tuning	0.45	0.04	0.13	8.1	10.97	1.5k	5.2k	4.8k	21.2
	CONTROL PREFIXES (A_2)	0.45	0.04	0.13	8.11	10.98	1.5k	5.3k	4.8k	21.5
E2E Clean	Prefix-tuning	0.32	0.003	0.01	5.70	7.28	6	57	130	24.8
	CONTROL PREFIXES (A_2)	0.32	0.003	0.01	5.71	7.29	8	73	140	25.3
WebNLG 2017	Prefix-tuning	0.52	0.09	0.26	8.57	11.88	973	4.6k	3.4k	21.1
	Prefix-tuning + DART	0.52	0.09	0.26	8.57	11.87	968	4.6k	3.4k	21.1
	CONTROL PREFIXES (A_1)	0.52	0.09	0.26	8.57	11.89	997	4.7k	3.4k	21.2
	CONTROL PREFIXES (A_2)	0.52	0.09	0.26	8.57	11.88	965	4.6k	3.4k	21.1
	CONTROL PREFIXES (A_1,A_2)	0.52	0.08	0.25	8.52	11.81	962	4.4k	3.4k	21.3
WebNLG+ 2020	Prefix-tuning	0.66	0.04	0.13	8.05	10.94	327	1.8k	1.6k	23.0
	CONTROL PREFIXES (A_1)	0.66	0.04	0.13	8.05	10.92	326	1.8k	1.6k	23.0
	CONTROL PREFIXES (A_2)	0.66	0.04	0.13	8.04	10.92	326	1.8k	1.6k	23.1
	CONTROL PREFIXES (A_1,A_2)	0.66	0.04	0.13	8.05	10.9	300	1.7k	1.5k	23.0

Table 9: The set of additional diversity and system characterization results on the official Data-to-text test sets. These metrics are proposed by Gehrmann et al. (2021) and calculated using the GEM evaluation suite. These include the Shannon Entropy over unigrams and bigrams (H_1, H_2), the mean segmented type token ratio over segment lengths of 100 (MSTTR, Johnson (1944)), the ratio of distinct n-grams over the total number of n-grams (Distinct$_{1,2}$), and the count of n-grams that only appear once across the entire test output (Unique$_{1,2}$, Li et al. (2016)), as well as the vocabulary size over the output ($|\mathcal{V}|$) and the mean output length of a system.

	DART	WebNLG	E2E Clean	ASSET		TurkCorpus	
		BLEU		SARI	QuestEval	SARI	QuestEval
Prefix-tuning + Control Tokens	51.72	61.89	43.57	43.64	0.63	42.36	0.66
CONTROL PREFIXES	51.95	62.27	44.15	43.58	0.64	42.32	0.66

Table 10: **Prefix-tuning + Control Tokens.** Comparison of our best CONTROL PREFIXES model for each dataset with prefix-tuning + control tokens for the same attributes. The guided simplification models are the average test set results over 5 random seeds.

Model	Stage	L-rate	Opt	Warmup-steps	Epochs	Batch Size	Effective Batch	Beam Width	LN-α	Min Target	Max Target	No Repeat Trigram
DART (T5-large)												
Prefix-tuning	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
CONTROL PREFIXES (A_1)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
E2E Clean (T5-large)												
Prefix-tuning	-	8e-5	Ada	2000	50	6	96	5	1	0	384	No
CONTROL PREFIXES (A_2)	1	7e-5	Ada	2000	30	6	96	5	1	0	384	No
	2	5e-5	Ada	2000	50	6	96	5	1	0	384	No
WebNLG (T5-large)												
Prefix-tuning	-	7e-5	Ada	2000	30	6	96	5	1	0	384	No
CONTROL PREFIXES (A_1)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
+Data: DART												
Prefix-tuning	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
CONTROL PREFIXES (A_2)	1	7e-5	Ada	2000	30	6	96	5	1	0	384	No
	2	3e-5	Ada	2000	30	6	96	5	1	0	384	No
CONTROL PREFIXES (A_1, A_2)	1	7e-5	Ada	2000	30	6	96	5	1	0	384	No
	2	3e-5	Ada	2000	30	6	96	5	1	0	384	No
ASSET & TurkCorpus (BART$_{LARGE}$)												
Prefix-tuning	-	5e-5	AdamW	2000	30	8	64	6	0.8	3	100	✓
CONTROL PREFIXES)	-	4e-5	Ada	5000	30	8	64	6	1	3	100	✓

Table 11: **Hyperparameters.** Detailed hyperparameter reporting for the models in this work. If the training procedure is multi-stage, each stage is indicated. L-rate is the learning rate, all learning follows a linear learning rate scheduler; Opt refers to the optimizer, Ada (Adafactor) or AdamW; Effective Batch = Batch size x # of gradient accumulation batches; LN-α refers to the α in length normalization during beam search.

ASSET Corpus	
	Source: The West Coast blues is a type of blues music characterized by jazz and jump blues influences, strong piano-dominated sounds and jazzy guitar solos, which originated from Texas blues players relocated to California in the 1940s.
Gold Reference†	The West Coast blues has jazz and jump blues influences. It also has piano-dominated sounds and jazzy guitar solos, which originated from Texas blues players who moved to California in the 1940s.
CONTROL PREFIXES ($A_{1:4}$) Length Control: 0.25	West Coast blues is a type of blues music from the United States.
CONTROL PREFIXES ($A_{1:4}$) Length Control: 0.5	The West Coast blues is a type of blues music made from Texas blues players who moved to California in the 1940s. It has jazz and jump blues influences.
CONTROL PREFIXES ($A_{1:4}$) Length Control: 0.75	The West Coast blues is a type of blues music that originated in Texas in the 1940s. It is characterized by jazz and jump blues influences, strong piano-dominated sounds and jazzy guitar solos.
CONTROL PREFIXES ($A_{1:4}$) Length Control: 1.0	The West Coast blues is a type of blues music characterized by jazz and jump blues influences, strong piano-dominated sounds and jazzy guitar solos, which originated from Texas blues players who moved to California in the 1940s.

Table 12: **Controlled text simplification**: a qualitative example from the ASSET Corpus validation set, with our corresponding CONTROL PREFIXES model outputs. Only the control prefix for the length control ratio is varied, depicted in red. †Note, this is one random gold reference out of 10.

ASSET Corpus	
	Source: The Great Dark Spot is thought to represent a hole in the methane cloud deck of Neptune.
Gold Reference†	The Great Dark Spot represents a hole in the methane cloud of Neptune.
CONTROL PREFIXES	It is thought that the Great Dark Spot is a hole in Neptune's methane cloud deck.
$\text{BART}_{\text{LARGE}}$ with ACCESS	The Great Dark Spot looks like a hole in the methane cloud deck of Neptune.
	Source: Fives is a British sport believed to derive from the same origins as many racquet sports.
Gold Reference†	Fives is a British sport developed from the same origins as many racquet sports.
CONTROL PREFIXES	Fives is a British sport. It is believed to have its origins in racquet sports.
$\text{BART}_{\text{LARGE}}$ with ACCESS	Fives is a British sport. It is thought to come from the same as many racquet sports.
	Source: Nevertheless, Tagore emulated numerous styles, including craftwork from northern New Ireland, Haida carvings from the west coast of Canada (British Columbia), and woodcuts by Max Pechstein.
Gold Reference†	Tagore copied many styles. These included craftwork from northern New Ireland, Haida carvings from western Canada and woodcuts by Max Pechstein.
CONTROL PREFIXES	Tagore emulated many different styles of art, including Haida carvings from the west coast of Canada (British Columbia), and woodcuts by Max Pechstein.
$\text{BART}_{\text{LARGE}}$ with ACCESS	Tagore copied many styles. He copied craftwork from northern New Ireland, Haida carvings from the west coast of Canada (British Columbia), and woodcuts by Max Pechstein.

Table 13: ***Fixed-LM* vs fine-tuned controlled text simplification**. CONTROL PREFIXES and $\text{BART}_{\text{LARGE}}$ with ACCESS (Martin et al., 2020) generated simplifications chosen from the ASSET Corpus test set. †Note, this is one random gold reference out of 10 for each example. The examples shown for CONTROL PREFIXES and $\text{BART}_{\text{LARGE}}$ with ACCESS are also randomly selected from one of the five model outputs.

WebNLG	
Unseen Category: *Athlete* Zero-shot -> SportsTeam	**Source:** <H> FC Torpedo Moscow <R> season <T> 2014-15 Russian Premier League <H> Aleksandr Chumakov <R> club <T> FC Torpedo Moscow <H> FC Torpedo Moscow <R> manager <T> Valery Petrakov <H> FC Torpedo Moscow <R> chairman <T> Aleksandr Tukmanov
Gold	Valery Petrakov is the manager of FC Torpedo Moscow and its chairman is Aleksandr Tukmanov. Aleksandr Chumakov plays for the club which spent the 2014-15 season in the Russian Premier League.
Prefix-tuning	Aleksandr Tukmanov and Valery Petrakov are the managers of FC Torpedo Moscow. The club played in the Russian Premier League in 2014-15 and their chairman is Aleksandr Tukmanov.
CONTROL PREFIXES (A_1)	Aleksandr Chumakov plays for FC Torpedo Moscow which is managed by Valery Petrakov. The club's chairman is Aleksandr Tukmanov and they played in the Russian Premier League in the 2014-15 season.
Unseen Category: *MeanOfTransportation* Zero-shot -> Airport	**Source:** <H> Costa Crociere <R> location <T> Genoa <H> Costa Crociere <R> parent Company <T> Carnival Corporation & plc <H> AIDAstella <R> operator <T> AIDA Cruises <H> AIDAstella <R> builder <T> Meyer Werft <H> AIDAstella <R> owner <T> Costa Crociere
Gold	Carnival Corporation & plc is the parent company of Costa Crociere in Genoa, who own the AIDAstella. AIDAstella was built by Meyer Werft and is operated by AIDA Cruises.
Prefix-tuning	Costa Crociere is located in Genoa and is owned by Carnival Corporation & plc. AIDAstella is operated by AIDA Cruises and was built by Meyer Werft.
CONTROL PREFIXES (A_1)	Costa Crociere is located in Genoa and is owned by AIDA Cruises. AIDAstella was built by Meyer Werft and is operated by AIDA Cruises. The parent company of Costa Crociere is Carnival Corporation & plc.

Table 14: **WebNLG example generations:** sources are shown in their linearized form, as fed to the T5-large based models, with prefix-tuning output and one of the gold references shown for comparison with CONTROL PREFIXES output. Triplesets are from WebNLG unseen categories and the zero-shot procedure is depicted using the textual category labels. As an example, for the unseen category *Athlete*, the closest Glove embedding belonging to a *seen* category label in embedding space is SportsTeam. Therefore the trained control prefix relating to *SportsTeam* is used for this example at inference time.

WebNLG+ 2020	
WebNLG MeanOfTransportation (Seen with Unseen Entities)	**Source:** <H> Pontiac Rageous <R> production Start Year <T> 1997 <H> Pontiac Rageous <R> assembly <T> Michigan <H> Pontiac Rageous <R> assembly <T> Detroit <H> Pontiac Rageous <R> production End Year <T> 1997 <H> Pontiac Rageous <R> body Style <T> Coupe <H> Pontiac Rageous <R> manufacturer <T> Pontiac
Gold	The Pontiac Rageous was a car with a coupe body style manufactured by Pontiac. Assembled in both Michigan and Detroit, it went into production in 1997, ending in the same year.
Prefix-tuning	The Pontiac Rageous is a coupe manufactured by Pontiac. It is assembled in Detroit, Michigan and began production in 1997.
CONTROL PREFIXES (A_1,A_2)	The Pontiac Rageous is manufactured by Pontiac in Detroit, Michigan. Its production began in 1997 and ended in 1997. The Pontiac Rageous has a coupe body style.
WebNLG (Unseen) Unseen Category: MusicalWork Zero-shot -> Artist	**Source:** <H> Bootleg Series Volume 1: The Quine Tapes <R> genre <T> Rock music <H> Bootleg Series Volume 1: The Quine Tapes <R> preceded By <T> Squeeze The Velvet Underground album <H> Bootleg Series Volume 1: The Quine Tapes <R> record Label <T> Polydor Records <H> Bootleg Series Volume 1: The Quine Tapes <R> recorded In <T> San Francisco
Gold	The Velvet Underground Squeeze album was succeeded by the rock album Bootleg Series Volume 1: The Quine Tapes, recorded under record label Polydor Records in San Francisco.
Prefix-tuning	The record label of Bootleg Series Volume 1: The Quine Tapes is Polydor Records. It was recorded in San Francisco and was preceded by Squeeze The Velvet Underground. Its genre is rock music.
CONTROL PREFIXES (A_1,A_2)	Squeeze The Velvet Underground was preceded by Bootleg Series Volume 1: The Quine Tapes, which was recorded in San Francisco and released by Polydor Records. The genre of the album is rock music.

Table 15: **WebNLG+ 2020 generations:** sources are shown in their linearized form as fed to the T5-large based models. The DART sub-dataset Source control prefix is highlighted, along with the final Category control prefix. The zero-shot procedure is depicted for the Unseen Category *MusicalWork*. The closest embedding belonging to a Seen category in embedding space is Artist.