Imperial College London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

CONTROL PREFIXES for Text Generation

Author:
Jordan Clive

Supervisor:
Dr. Marek Rei [a]
Co-supervisor:
Dr. Kris Cao [b]

[a]Imperial College London
[b]DeepMind, London

Submitted in partial fulfillment of the requirements
for the MSc degree in Computing (Machine Learning & A.I.) of Imperial College London

September, 2021

Abstract

The prevailing paradigm of natural language processing comprises using fine-tuning to adapt pre-trained language models to downstream tasks. As we pre-train larger models, retraining all model parameters quickly becomes undesirable. Subsequently, there has been a shift in focus to more parameter efficient alternatives, spawning the field of *prompt learning*. This is where language models are adapted to downstream tasks with the aid of a prompt. Most of the current work on *prompt learning* in text generation uses a shared dataset-level prompt for all examples in the dataset (*static* methods). PREFIX-TUNING is a *static* prompting method that optimizes a small dataset-specific continuous prompt. Based on PREFIX-TUNING, we propose a *dynamic* method: CONTROL PREFIXES, which incorporates conditional input-dependent prompts. CONTROL PREFIXES empowers the model to have finer-grained control during text generation than current *prompt learning* methods by conditioning on attributes of the input. This method is at the intersection of *prompt learning* and *controlled generation*; the latter a field that aims to incorporate various types of guidance beyond the input text into the generation model. We provide systematic evaluation of the technique and apply CONTROL PREFIXES on five of the eleven GEM benchmark datasets, an important benchmark for natural language generation (NLG). We present state-of-the-art results on three of the GEM Data-to-Text tasks, along with state-of-the-art results on WebNLG 2017. We release our implementation at https://github.com/jordiclive/ControlPrefixes/.

Acknowledgements

I would like to thank my supervisor Dr. Marek Rei and my co-supervisor Dr. Kris Cao for their invaluable support and guidance throughout the project. It was beneficial to see this work through two lenses; our frequent catch ups shaped the project direction, and helped me appreciate how this work fits in the larger context.

I would also like to thank, in alphabetical order; Christopher Bryant, Thiago Castro Ferreira, Daniel Khashabi, Sebastian Gehrmann, Lisa Xiang Li, Louis Martin, Mounica Maddela, Thomas Scialom and Anastasia Shimorina, for providing valuable further detail about their work in private communications.

Contents

Section 1: Introduction

Recently, approaches in natural language processing (NLP) have been dominated by adapting one large-scale, pre-trained language model (PLM) to various downstream tasks. Such adaptation was usually done via fine-tuning, which necessitates updating and storing all of the parameters, resulting in multiple new language models (LMs), one for each task. As the trend of larger PLMs continues, this overhead moves from mere inconvenience for models such as BART or T5-large ($<$ 1B parameters) to a formidable deployment challenge for GPT-3 with 175B parameters.

Many researchers sought to alleviate this issue by using *fixed-LM* techniques, where all the parameters of the base LM always remain unchanged. An ever-growing subsection of these methods can be classed under *prompt learning*, where language models are adapted to downstream tasks with the aid of a prompt accompanying the input. PREFIX-TUNING (Li and Liang [2021]) is one of the most promising of these alternatives, which has constituents of a prompt at every layer of a model, and adds only 0.1–2% additional parameters to the base LM.

Controlled generation is a highly related field: aiming to incorporate various types of guidance (e.g. length specifications (Kikuchi et al. [2016])) beyond the input text into the generation model. Most current work on *prompt learning* in text generation shares a single dataset or task-level prompt, meaning the prompting function is *static*. Only very few works have explored *dynamic* prompts, which are input-dependent. *Dynamic* prompts may empower the model to have finer-grained control on generation than current methods by conditioning on attributes of the input. This controlled text generation may provide valuable direction for future work on *prompt learning* (Liu et al. [2021a]).

We propose the *dynamic* prompting method CONTROL PREFIXES, which is built on top of the *static* prompting technique PREFIX-TUNING. Whilst preserving the *fixed-LM* property, CONTROL PREFIXES enables multiple attributes to act as guidance signal at the input-level. Although task agnostic, we focus on text generation tasks and evaluate CONTROL PREFIXES on Data-to-Text, Grammatical Error Correction (GEC), Abstractive Summarization, and Sentence Simplification using either $\text{BART}_{\text{LARGE}}$ or T5-large as the fixed LM. In addition to conducting internal human evaluation for Data-to-Text, we include our Abstractive Summarization submission results on GENIE (Khashabi et al. [2021])—an external platform that organizes and runs the human evaluation.

Thesis Objectives

- To investigate whether PREFIX-TUNING for the encoder-decoder, alone, is an effective technique when applied to tasks other than summarization. To also be more explicit on the exact architecture for the encoder-decoder than the original paper (Li and Liang [2021]).

- To investigate whether a parameter efficient, *fixed-LM* method such as PREFIX-TUNING can itself leverage conditional information at the input-level and still keep the LM fixed. To achieve this, we introduce the novel method CONTROL PREFIXES.
- To demonstrate that CONTROL PREFIXES is effective on an array of NLG tasks. We present state-of-the-art results on several Data-to-Text datasets, and strong performance with CONTROL PREFIXES on other NLG tasks.
- To investigate whether the additional CONTROL PREFIXES parameters are interpretable and whether zero-shot learning is possible for conditioning on input-level information previously unseen during training.

Thesis Structure

Section 2 provides necessary background for this thesis, incorporating a description of the current state of NLG and NLG model evaluation. Section 3 introduces *controlled generation*, related work, and the rapidly developing field of *prompt learning*. Section 4 introduces CONTROL PREFIXES; Section 5 outlines the generation tasks considered and overall experimental setup. Section 6 reports the model setups and main results on an array of generation tasks. Section 7 provides some additional discussion on zero-shot learning with CONTROL PREFIXES and CONTROL PREFIXES interpretability. Section 8 documents human-assessed evaluation of our model's generations, before we offer concluding thoughts and directions for future research in Section 9.

Section 2: Background

2.1 Natural Language Generation Overview

> The limits of my
> language mean the limits of my world.
>
> Ludwig Wittgenstein

Since the transformer encoder-decoder was proposed by Vaswani et al. [2017], there has been a revolution in the field of natural language processing (NLP). The improved efficiency and parallelization of transformer-based language models (LMs) enabled GPT (Radford et al. [2019]) and BERT (Devlin et al. [2019]) to be pre-trained on a vast array of unlabelled data. Once pre-trained, models like BERT and GPT required minimal fine-tuning[1] to surpass previous state-of-the-art (SOTA) records on more than a dozen natural language understanding (NLU) tasks.

Nevertheless, standalone BERT and GPT models have been less effective for natural language generation (NLG) tasks, for example, translation or response generation. Language modelling is the predominant task for pre-training, where the target words are predicted, conditioned on a given context. Therefore, it was intuitive to employ the pre-trained LMs for natural language generation as the pre-training objective naturally accords with the goal of NLG. However, we do not write words from scratch but instead based on a particular context, e.g. the source language sentences for translation or the dialogue histories for response generation. Hence it was pre-trained transformer encoder-decoders [2] that incorporated sequence-to-sequence pre-training, such as BART (Lewis et al. [2020]) which attained top performance on a variety of sequence-to-sequence tasks (Zhang et al. [2019]; Raffel et al. [2020]); Zhang et al. [2020c])

There is a catch—these pre-trained LMs use heavily over-parameterized representations consisting of hundreds of millions or billions of parameters. Fine-tuning requires updating *all* parameters and storing one copy of the fine-tuned model per task. If one could reuse one pre-trained language model, this would reduce substantial storage and deployment costs that hinder the applicability of large-scale PLMs to real-world applications.

Therefore, this work focuses on *fixed-LM* methods that do not alter the pre-trained language model parameters and add $<3\%$ additional parameters. Before describing the architecture of models such as BART, which are used as the 'fixed LMs' for the methods implemented, we explain the exact kind of tasks this thesis will tackle. At the very end of this section we outline the current state of NLG evaluation, which is constantly evolving alongside the development of LMs. Some base knowledge of machine learning is assumed.

[1] Fine-tuning is defined as the task-specific training of a model that has been initialized with the weights of a pre-trained language model.
[2] Also denoted seq2seq transformer models.

Sequence-to-sequence Tasks

Although the techniques we consider are mainly agnostic to NLP domain, we constrain our research to natural language generation (NLG) tasks, specifically sequence-to-sequence tasks. This means tasks are formulated into a "text-to-text" format—the model is fed some text as a sequence for context or conditioning and is then asked to produce some output text sequence. We use the notation LM_ϕ to denote a pre-trained LM with an auto-regressive decoder parameterized by ϕ.

Sequence-to-sequence tasks are defined as a mapping from a tokenized input sequence ($X = \{x_1, \ldots, x_n\}$) to a tokenized output sequence ($Y = \{y_1, \ldots, y_m\}$) of *a-priori* unknown output length m. Auto-regressive language generation is grounded on the assumption that the probability distribution of a token sequence can be decomposed into the product of conditional next token distributions. The model LM_ϕ learns the conditional distribution $P_\phi(Y|X)$ as:

$$P_\phi(Y|X) = P_\phi(y_1, \cdots, y_m | x_1, \cdots, x_n) = \prod_{t=1}^{m} p_\phi(y_t | X, y_1, \cdots, y_{t-1}). \tag{2.1}$$

This framework can provide a consistent training objective both for pre-training and fine-tuning. This enables use of the maximum likelihood objective, teacher forcing and a cross-entropy loss function (Williams and Zipser [1989]).

Maximum Likelihood Objective

Although there are alternatives to the maximum likelihood objective for text generation (e.g. Guo et al. [2021]), it is by far the predominant objective. Each downstream dataset is represented by a *corpus* of context-target pairs: $\mathcal{Z} = \{\langle X^j, Y^j \rangle\}_{j=1,..,N}$, where both X^j and Y^j are sequences of tokens. The generation process is regarded as a sequential multi-label classification problem. It can be directly optimized by the negative log likelihood (*NLL*) loss. Therefore, the objective of a text generation model via maximum likelihood estimation (MLE) for a corpus is formulated as:

$$\mathcal{L}_{NLL}(\phi) = -\log p_\phi(\mathcal{Y}|\mathcal{X}) = -\sum_{j=1}^{N} \sum_{t=1}^{m} \log \Big(p_\phi(y_t^j | y_{<t}^j, X^j) \Big), \tag{2.2}$$

where the notation $y_{<t}$ represents a sequence of tokens from y_0 to y_{t-1}, i.e. $(y_0, \cdots, y_{t-1})$.

2.2 Transformer Encoder-decoder Models

Currently, the Transformer is the de facto architecture of choice for processing sequential data and is starting to be applied to vision applications (e.g. Dosovitskiy et al. [2020]). All the techniques we discuss use transformer encoder-decoder models as the fixed LM. Therefore, unless explicitly stated, any reference to an LM can be assumed to be a transformer encoder-decoder. The fixed LMs used in this work largely follow the original implementation (Vaswani et al. [2017]). PREFIX-TUNING and CONTROL PREFIXES both modulate how the fixed LM processes input. In this section we elaborate on the components pertinent to how this is done.

Overview

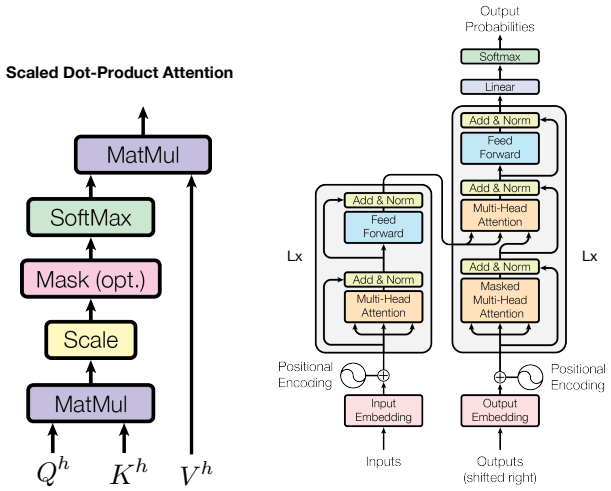


Figure 2.1: Left: Visualization of scaled-dot product attention. Right: The original Vaswani et al. [2017] transformer encoder-decoder architecture. Note there are L encoder layers and L decoder layers. Adapted from Vaswani et al. [2017].

There is an E of the LM[3] that maps the tokenized sequence $[X,Y]$ to a sequence of embeddings $[\mathbf{e}(X),\mathbf{e}(Y)]$. This sequence of embeddings is then propagated up through the layers of the model. The encoder learns to encode a variable length sequence into a fixed length vector representation. The decoder is tasked with decoding a given fixed length vector representation into a variable length sequence[4].

PREFIX-TUNING for the encoder-decoder adapts these models. PREFIX-TUNING is more expressive than additional continuous embedding prompts at the input, by virtue of having constituents of the prompt at every layer of an LM. We introduce multi-head attention in detail

[3]The embedding matrix does not have to be shared across the encoder and the decoder, but we consider LMs that share the embedding matrix.

[4]Pre-training solely the encoder outperforms pre-training solely the decoder (Liu and Lin [2019]). Although no unanimous theory for this phenomenon exists, it does suggest that PLMs are more robust when acting as representation extractors, whilst more sensitive to the change of context when acting as conditional language generators.

(Fig. 2.1, Right), as this is exactly where PREFIX-TUNING modulates the encoder-decoder in each layer of the model.

2.2.1 Attention in Encoder-Decoders

Scaled Dot-product Attention

The scaled dot-product attention (Fig. 2.1, Left) was introduced by Vaswani et al. [2017] and forms the cornerstone of the transformer architecture. The set of equations are:

$$\text{Attention}(Q^s, K^s, V^s) = a\left(\frac{Q^s K^{s\top}}{\sqrt{d_q}}\right)V^s, \tag{2.3}$$

where

$$\begin{aligned} Q^s &= (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N)^\top \in \mathbb{R}^{N \times d_q} \\ K^s &= (\boldsymbol{k}_1, \ldots, \boldsymbol{k}_M)^\top \in \mathbb{R}^{M \times d_q} \\ V^s &= (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M)^\top \in \mathbb{R}^{M \times d_v}. \end{aligned} \tag{2.4}$$

To give a high-level intuition, the set of equations in 2.3 and 2.4 can be understood from an information retrieval perspective. Query vectors, represented by $\boldsymbol{q}_i \in \mathbb{R}^{d_q}$, are submitted to the system and the system will check the degree of 'matching' against key vectors, represented by $\boldsymbol{k}_j \in \mathbb{R}^{d_q}$. Each key vector, $\boldsymbol{k}_j$ is associated with a value vector $\boldsymbol{v}_j \in \mathbb{R}^{d_v}$, so that if $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ are aligned, the value vector $\boldsymbol{v}_j$ will be weighted strongly.

Here $a(\cdot)$ is an activation function applied row-wise. In our work it is always the softmax function. Incorporating $\sqrt{d_q}$ prevents the variance of the dot product increasing linearly with the dimensionality, which would push the softmax into regions with negligible gradients[5].

Multi-head Attention

Vaswani et al. [2017] noted improved performance with multi-head attention, which facilitates multiple alignment processes by projecting the inputs (hidden states) into different sub-spaces and then performing dot-product attention in such sub-spaces (an attention head). The outputs of each attention head are concatenated and projected to produce the final output. During training, each head extracts a different pattern of the sequence, such as specific positional offsets, objects of prepositions and delimiter tokens (Voita et al. [2019]). Therefore, multiple complementary patterns in the sequence can be captured, in contrast to when using a single attention head.

[5]This is based on the assumption that the elements in $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ are independently distributed with variance 1, therefore the variance of $\langle \boldsymbol{q}_i, \boldsymbol{k}_j \rangle$ would be d_q.

For clarity we present only the multi-head attention, with associated dimensions, that is utilized by the fixed LMs in our work[6]. We use the superscript to denote the head $h \in \{1,...,H\}$[7]. In detail, the mathematical form is the following:

$$\begin{aligned} Q^h &= \left(\boldsymbol{q}_1^h,...,\boldsymbol{q}_N^h\right)^\top \in \mathbb{R}^{N\times\tilde{d}}, \boldsymbol{q}_i^h = \mathbf{h}_i W^{Q,h} \quad \forall i \in \{1,...,N\} \\ K^h &= \left(\boldsymbol{k}_1^h,...,\boldsymbol{k}_M^h\right)^\top \in \mathbb{R}^{M\times\tilde{d}}, \boldsymbol{k}_j^h = \mathbf{s}_j W^{K,h} \quad \forall j \in \{1,...,M\} \\ V^h &= \left(\boldsymbol{v}_1^h,...,\boldsymbol{v}_M^h\right)^\top \in \mathbb{R}^{M\times\tilde{d}}, \boldsymbol{v}_j^h = \mathbf{s}_j W^{V,h} \quad \forall j \in \{1,...,M\}, \end{aligned} \tag{2.5}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is the hidden state of the i^{th} token in the sequence that relates to the queries and $\mathbf{s}_j \in \mathbb{R}^d$ is the hidden state of the j^{th} token in the sequence relating to the keys and values. The projection matrices $W^{Q,h}, W^{V,h}, W^{K,h} \in \mathbb{R}^{d\times\tilde{d}}\ \forall h \in \{1,...,H\}$ and $W^O \in \mathbb{R}^{d\times d}$ are learnt. To keep the costs close to performing single-head attention in the original space, $\tilde{d} = d/H$.

The Attention function is computed H times in parallel and the outputs concatenated to obtain the final output of the multi-head attention:

$$\text{head}_h = \text{Attention}\left(Q^h, W^h, V^h\right) \tag{2.6}$$

$$\text{MHA}(Q,K,V) = \text{ Concat }\left(\text{ head }_1,\cdots,\text{ head }_H\right)W^O, \tag{2.7}$$

where we denote

$$\begin{aligned} Q &= \text{ Concat }\left(Q^1,\cdots,Q^H\right) \in \mathbb{R}^{N\times d} \\ K &= \text{ Concat }\left(K^1,\cdots,K^H\right) \in \mathbb{R}^{M\times d} \\ V &= \text{ Concat }\left(V^1,\cdots,V^H\right) \in \mathbb{R}^{M\times d}. \end{aligned} \tag{2.8}$$

Multi-head Attention for PREFIX-TUNING

These equations are especially important for the PREFIX-TUNING implementation we consider (Li and Liang [2021]). In PREFIX-TUNING the K and V matrices are augmented for every multi-head attention calculation, enlarging the context window [8]. For PREFIX-TUNING: $K, V \in \mathbb{R}^{(M+\rho)\times d}$, where ρ is a PREFIX-TUNING parameter.

This means the time complexity for the multi-head attention calculations increases from $\mathcal{O}(MNd)$ to $\mathcal{O}((M+\rho)Nd)$ and the space complexity increases from $\mathcal{O}(MN+Nd)$ to $\mathcal{O}((M+\rho)N+Nd)$. During the forward pass of $[X;Y]$ the PREFIX-TUNING key-value pairs are independent of previous model computations, hence why PREFIX-TUNING enables batching across tasks. Therefore, the other associated space and time complexity considerations for multi-head attention (for the input projections and in combining the outputs) are the same as without these additional key-value pairs.

[6] All fixed-LMs considered: $\text{BART}_{\text{LARGE}}$, T5-large and $\text{PEGASUS}_{\text{LARGE}}$ have $d_q = d_v = d = 1024$, and $H = 16$.
[7] We use different notation to the original implementation to assist in explaining PREFIX-TUNING.
[8] Alternatively, models like the Longformer (Beltagy et al. [2020]) minimize the context window.

Positional Encodings

Attention (eqn 2.8) is equivariant to row permutations in the query matrix Q, and as ordering information is necessary for natural language generation tasks, this needs to be remedied. In the original Transformer architecture, absolute positional encodings (2.1) solve this problem and are computed through a deterministic function. Absolute positional encodings can also be learnt, as is the case for BART. However, as absolute positional embeddings do not capture relative word order, models such as T5 (Raffel et al. [2020]) use relative position encodings (Shaw et al. [2018]) that produce a different learned embedding according to the offset between the "key" and "query" being compared in the attention computation[9].

Decoder Block

Bi-directional multi-head attention, as in each encoder layer, permits the full context to be attended to. In contrast, the decoder layer comprises two attention mechanisms: a masked self-attention (**Dm**) and cross-attention (**Dc**) module. The masked multi-head attention only permits input vectors y_j to attend to y_i, with $i \leq j$.

This is achieved by constructing an "attention mask", $M \in \mathbb{R}^{N \times M}$ replacing the operation in 2.3 with

$$\text{Attention}(Q^h, K^h, V^h) = a\left(\frac{Q^h K^{h\top} + M}{\sqrt{d_q}}\right)V^h \quad \text{where} \quad M_{ij} = \begin{cases} 0, & i \leq j \\ -\infty, & i > j \end{cases} \tag{2.9}$$

Tokens are forbidden from attending to the right context. The cross-attention layer facilitates the decoder to be conditioned on the contextualized encoded sequence ($\overline{X} = \{\overline{\mathbf{x}}_\mathbf{1}, \cdots, \overline{\mathbf{x}}_\mathbf{n}\}$), through having the (K, V) in 2.8 being formed from projections of $\overline{X}$.

2.2.2 BART and T5 as Pre-trained Language Models

BART and T5 both refer to pre-trained transformer encoder-decoders. $\text{BART}_{\text{LARGE}}$ [10] and T5-large are specific model sizes from the respective BART and T5 class of models.

With an *encoder-only* model (e.g Fig. 2.2, **a)** BERT), X is mapped to $\overline{X}$, where, for example $\overline{X}$ is then processed by a classification layer for NLU classification tasks. Encoder-only models can only map an X to a Y of *a priori* output length—making it infeasible to use encoder-only models for seq2seq tasks.

[9] The positional encodings at the input-level shown in 2.1 would be omitted in this case.

[10] The main architectural change to the original Transformer encoder-decoder is that the RELU activation functions are replaced by GELUs (Hendrycks and Gimpel [2016]).

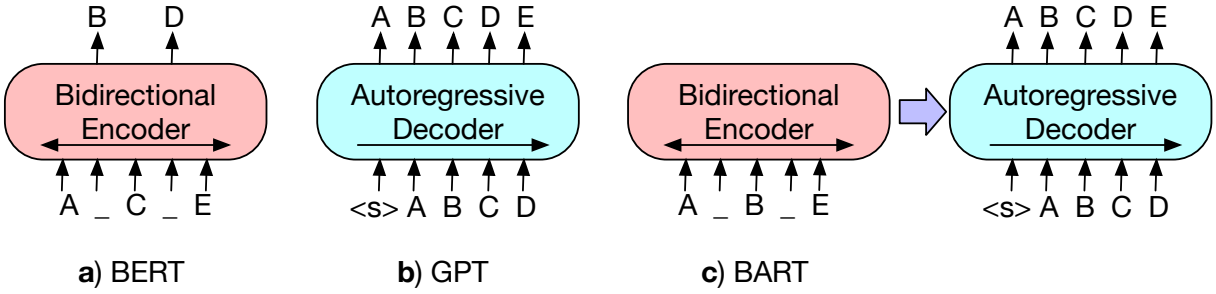


Figure 2.2: BART architecture, shown as a combination of the encoder and decoder-only models: BERT and GPT. **a)** The Cloze Task (Taylor [1953]) is shown as a pre-training task for BERT. **c)** BART pre-training de-noising task: the span [C, D] is masked before encoding and an extra mask is inserted before B, leaving the corrupted document "A _ B _ E" as input to the encoder. Source: Lewis et al. [2020].

With a *decoder-only* model (e.g Fig. 2.2, **b)** GPT) there is only masked self-attention. These models have fueled interest into 'prompting': where an input prompt $y_{<i}$ is fed to the model to yield the conditional distribution $p_\phi(y_i \mid y_{<i})$, whereby sampling then proceeds autoregressively. By setting the input prompt as X, GPT can be used for conditional generation. Only possessing masked self-attention limits the model's representation of the sequence input $\overline{X}$, as $\overline{\mathbf{x}}_\mathbf{i}$ cannot depend on $\overline{\mathbf{x}}_{\mathbf{i+1}}, \forall i \in \{1,...,n\}$.

Fig. 2.2 explains how $\text{BART}_{\text{LARGE}}$'s architecture combines benefits of GPT and BERT. Both $\text{BART}_{\text{LARGE}}$ (400M parameters) and T5-large (740M parameters)[11] were trained with de-noising objectives on different sets of large-scale data [12]. The unsupervised "span masking" objective for $\text{BART}_{\text{LARGE}}$, as shown in Fig 2.2 **c)**, is an example of a de-noising objective, where the model has to learn how to correct corrupted text. Various unsupervised de-noising tasks are given to the models during pre-training. This approach generalizes the original word masking and next sentence prediction objectives of BERT by coercing the model to reason more about overall sentence length. T5-large was also pre-trained on supervised translation, summarization, classification, and question answering tasks [13].

Justification for using these models as the fixed LMs in our *fixed-LM* based techniques is that they are *generalist* pre-trained models: having shown strong performance on the range of generation tasks we consider in the study. Additionally, at the time of writing, $\text{BART}_{\text{LARGE}}$ is the only encoder-decoder model to be used with PREFIX-TUNING[14]. We also trial PREFIX-TUNING

[11] These model sizes ensured the experimentation was practicable with the project's modest resources. We could use T5-large for the smaller Data-to-Text datasets (<70k samples).

[12] $\text{BART}_{\text{LARGE}}$ uses the same pre-training data as Liu et al. [2019]—namely, 160GB of news, books, stories, and web text, whilst T5-large is trained on C4.

[13] Interestingly, not Data-to-Text tasks, which are the tasks we confine our experiments with T5-large to, and where T5-large has been shown to excel (Kale [2020]).

[14] Only trialled for the Abstractive Summarization XSum dataset in Li and Liang [2021]

with $\text{PEGASUS}_{\text{LARGE}}$ (Zhang et al. [2019]) for summarization; for the pre-training tasks of $\text{PEGASUS}_{\text{LARGE}}$ were purely focused on Abstractive Summarization[15].

2.3 Generation in Practice

This section highlights how text is actually generated by these models.

During text generation, the decoder generates a text sequence through a decoding strategy. *Beam search* is a decoding strategy that does breadth-first search, one token per tree level, but with a limited bandwidth. At each level of the search tree, beam search keeps track of n (the 'beam width') best candidates and expands all successors of these candidates in the next level. Beam search halts expansion if a node arrives at the EOS (end-of-sentence) token. The hypothesis is chosen which has the overall highest probability in the final beam. This maximization-based decoding strategy does not guarantee the most likely output or even high-quality generation (Fig 2.3).

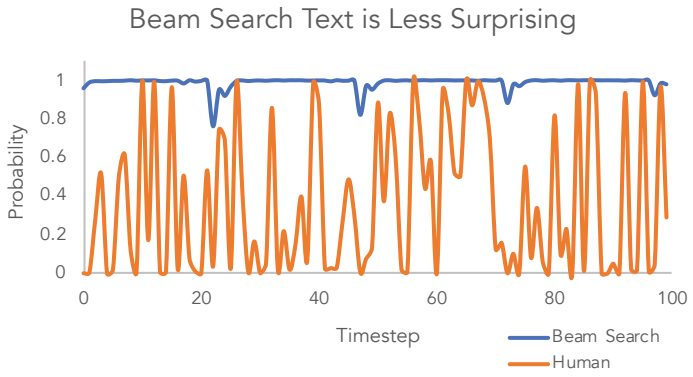


Figure 2.3: The probability assigned to tokens generated by beam search and humans, given the same context. Note the increased variance that characterizes human text, in contrast with the endless repetition of text decoded by beam search. Source: Holtzman et al. [2019].

Beam search is still the predominant decoding strategy, excelling where the length of the desired generation is predictable; for instance, summarization or error correction. However, for open-ended generation, where the desired output length can vary significantly (dialogue or story generation), beam search suffers from repetitive generation. This is remarkably challenging to control with n-gram or other penalties [16]. In this work, all decoding is done with beam search; although, we note its limitations such as lack of n-gram diversity.

Decoding hyper-parameter considerations do not change any of the trainable parameters, but do have a profound affect on the generation quality. In this work we consider the *beam width*,

[15] The only difference in the base architecture of PEGASUS compared to BART is that PEGASUS uses sinusoidal positional encodings.

[16] This is why another decoding strategy *top-k sampling* is used in work such as Radford et al. [2019] (GPT-2).

the *length normalization alpha* [17] (Wu et al. [2016]), the *minimum length*, *maximum length* and the n*-gram penalty*. The *minimum length* can be used to obligate the model to not produce an EOS token before the *minimum length* is reached. This is used frequently in summarization, or simplification where there is a strong prior on the output length. The n*-gram penalty* introduced by Paulus et al. [2017] and Klein et al. [2017] ensures no n-gram appears twice—we consider the choice of up to which n-grams the penalty should be incurred to be a hyper-parameter.

2.4 Evaluation in NLG

NLG is one of the most active research fields in NLP, and subsequently, the number of available datasets, metrics and evaluation strategies is rapidly expanding[18]. NLG evaluation is notoriously challenging—principally because many NLG tasks are so open-ended. For example, a dialogue or story generation system can generate multiple plausible responses for the same user input. This section outlines the NLG system evaluation measures we use to assess model performance and the associated limitations.

2.4.1 Human-centric Metrics

Human evaluation, assuming the execution meets certain stipulations [19], remains the gold standard for almost all NLG tasks. Typically the evaluation involves crowdsourcing where qualified workers assess *intrinsic dimensions* (e.g. fluency, coherence, correctness) of a generated text.

However, human evaluation is expensive and is customarily conducted by individual researchers with varying design decisions that impact quality control and consistency; thereby, making results largely incomparable across evaluations (Celikyilmaz et al. [2020]). To enforce consistency one can outsource the human evaluation from the individual researchers to an evaluation platform, by people hosting a shared task or leaderboard, such as GENIE (Khashabi et al. [2021]).

2.4.2 Automatic Metrics

Researchers often resort to *automatic metrics* for quantifying model performance. These metrics are principally chosen to correlate well with human judgement, and are often calculated using human written references. The success of these metrics ultimately depends on how open-ended the underlying task is and both the quality and quantity of references. Multiple references are shown to improve the correlation of automatic metrics with human judgments (Han

[17] Values < 1.0 encourage the model to generate shorter sequences, whilst values > 1.0 encourage the model to produce longer sequences.

[18] For a comprehensive description of recent developments in NLG evaluation, we refer readers to the survey by Celikyilmaz et al. [2020].

[19] Namely the sample size, human annotator training and inter-annotator agreement.

and Wong [2016]; Läubli et al. [2020]) for NLG tasks. Although, notably 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.

Automatic metrics can be classed into *untrained metrics* and *machine-learned metrics*—based on machine-learned models. Untrained metrics are efficient to compute and are still the prevailing approach to quantify the day-to-day progress of model development. Despite this, there is an ongoing evolution of incorporating *machine-learned metrics* that rely on PLMs and have shown improved correlations with human judgements (Zhang et al. [2020b]). The field concerning *untrained metrics* is undergoing rapid evolution—there is still little consensus on reporting standards.[20] There are additional concerns about the implicit circularity of evaluating models that may utilize the same pre-trained embeddings (Celikyilmaz et al. [2020]).

Untrained Automatic Metrics

BLEU (Papineni et al. [2002]) is an n-gram overlap metric, used to evaluate the degree of matching between a candidate translation of text to one or more reference translations. BLEU is a weighted geometric mean of n-gram precision scores. There are drawbacks for NLG tasks where contextual understanding and reasoning are key; Caccia et al. [2018] found that generated text with perfect BLEU scores was often grammatically correct but lacked semantic or global coherence. Despite the shortcomings, BLEU has been shown to correlate highly with human judgement for Data-to-Text tasks (Castro Ferreira et al. [2020]; Semeniuta et al. [2019]), and is still the main yardstick for this task, along with the metric **METEOR** (Lavie and Agarwal [2007])—which measures not just precision but the harmonic mean of the unigram precision and recall.

ROUGE (Lin [2004]) is a set of metrics for mainly evaluating automatic summarization of long texts. ROUGE's reliance on n-gram matching can be problematic, most notably for long-text generation tasks (Kilickaya et al. [2017])—it doesn't provide information about the narrative flow, grammar, or topical flow of the generated text.

Assessing Diversity of Generation

One issue in evaluating text generation systems is the diversity of generation, particularly when the text to evaluate is long. The generated text can be fluent, valid given the input and informative for the user, but may still lack lexical overlap with the reference text (Gao et al. [2019]; Welleck et al. [2019]). We address this issue by reporting additional metrics focused on generation diversity, which is discussed at the end of the section.

[20] We note the importance of reporting the accompanying hash (e.g. for BERTScore: `roberta-large_L17_no-idf_version=0.3.8(hug_trans=3.0.1)`, which encodes the underlying PLM information, as the same metric with different hashes can produce vastly different scores.

Machine-learned Metrics

Machine-learned models are trained on human judgement data aiming to mimic human judges in assessing the quality of output, such as factual correctness, naturalness, fluency and coherence.

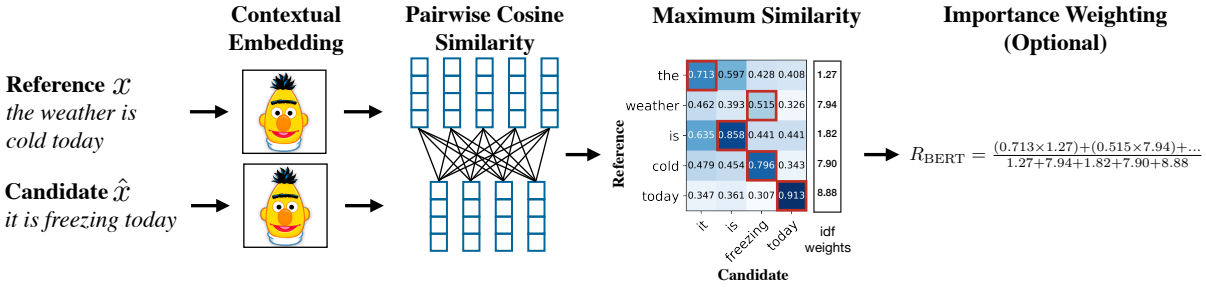


Figure 2.4: The computation of the BERTScore recall metric, $\mathrm{R_{BERT}}$. Source: Zhang et al. [2020b].

We report BERTScore (Zhang et al. [2020b]), a promising BERT based metric, appropriate for many NLG tasks. BERTScore can compute precision, recall, and F1 measures and is based on the similarity of sentence embeddings. As illustrated in Fig. 2.4, the metric employs the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences via cosine similarity. However, BERTScore is still a token-level metric, so although robust to synonyms and sentence structure, it suffers from token misalignment; thus, sharing the limitation of being very dependent on the number of references available. We additionally report the BERT-based metric, BLEURT (Sellam et al. [2020]), on all datasets which has exhibited high correlation with human judgement on a number of NLG tasks.

QA Metrics Another class of metric that we consider for Simplification is Question and Answering (QA) metrics. These measures can provide an alternative perspective to model evaluation that does not highly correlate with other evaluation approaches (Fabbri et al. [2020a]).

We use QuestEval (Scialom et al. [2021]), which was originally proposed for summarization, to assess our text simplification models. It evaluates if a summary is factually consistent with respect to a source document. It does this by i) generating a list of questions and ii) retrieving the corresponding answers from the source document and the summary. If the answers are similar, the summary is deemed appropriate. Table 2.1 depicts an example of the procedure, with the recent simplification specific modification: BERTScore is used to measure the answer similarity rather than F1 score. By harnessing its dense representations, a smoother similarity function than the F1 can be computed, which permits synonyms to be taken into account—an important capability for Simplification.

Source Text: In the Soviet years, the Bolsheviks demolished two of Rostov's principal landmarks- St Alexander Nevsky cathedral (1908) and St George cathedral in Nakhichevan (1783-1807).			
Simplification: The Bolsheviks destroyed St. Alexander Nevsky cathedral and St. George cathedral in Nakhichevan during the Soviet years.			
Generated Question	**Answers** On Source	On Simplif.	**Score** BERTScore
When did the Bolsheviks demolish St George cathedral?	the Soviet years	Soviet years	0.89
Who demolished St Alexander Nevsky cathedral?	demolished	destroyed	0.82
How many of Rostov's main landmarks were demolished?	two	Unanswerable	0.0
0.0			
What cathedral was demolished in 1908?	Rostov	Unanswerable	0.0
[...]	[...]	[...]	[...]

Table 2.1: Example of questions automatically generated and answered by the QA metric *QuestEval* given a source text and its simplification. Adapted from Scialom et al. [2021].

Result Reporting for our Experiments

There are complex interactions between sometimes contradicting measures. In light of this, we report additional evaluation metrics, including machine-learned metrics for all our models. This is to align with our objective of systematically evaluating CONTROL PREFIXES and vanilla PREFIX-TUNING. We report the GEM (Gehrmann et al. [2021])[21] metrics that represent lexical similarity and semantic equivalence, available in Appendix Tables A.1,A.2. We are also conscious that NLG models intrinsically trade off diversity and quality. We therefore report diversity and system characterization results, also proposed by GEM, for all NLG tasks in Appendix Tables A.3, A.4.

[21] GEM is an NLG benchmark with an evaluation framework that can be applied to all NLG tasks. The framework is maintained by GEM organizers.

Section 3: Related Work

This section contrasts fine-tuning with parameter efficient alternatives in adapting a pre-trained LM to a specific task or dataset, emphasizing the limitations of fine-tuning. We use related work to motivate the rationale for why *fixed-LM* techniques, as well as both the fields of *prompt-learning* and *controlled generation*, possess desirable properties we wish to exploit.

3.1 Task-specific Adaptation of Pre-trained LMs

3.1.1 Fine-tuning

The predominant strategy for NLG state-of-the-art (SOTA) systems has been the *pretrain-then-finetune* strategy applied to encoder-decoders. However, large models are costly to share and serve, making the ability to reuse one frozen model for various downstream tasks desirable—particularly as the scale of pre-trained language models continues to climb. Moreover, full fine-tuning has been shown to be unnecessarily profligate—it has been quantitatively established by Radiya-Dixit and Wang [2020] and Li et al. [2018] that fine-tuned networks do not deviate substantially from the pre-trained one in parameter space. Aghajanyan et al. [2020] used intrinsic dimension to show that there exists a low dimension re-parameterization that is as effective for fine-tuning as the full parameter space.

In addition to requiring multiple downstream models that share the same NLU, the mismatch in data distribution and objective between the two training stages may actually result in performance degradation and instabilities in training (Dodge et al. [2020]; Peters et al. [2019]). The NLU garnered during pre-training is overwritten—this is termed the *catastrophic forgetting* problem. The *catastrophic forgetting* problem is why the performance of parameter efficient alternatives have been shown to be superior to fine-tuning in certain scenarios, especially in the low data-regime and generalization settings (Li and Liang [2021]). Here the NLU assembled by the models in pre-training is retained rather than the models using their capacity to overfit a narrow distribution.

3.1.2 Parameter Efficient Alternatives

To combat the problems outlined, various alternatives have been proposed such as training a subset of parameters or excising model weights by training a binary mask over model parameters (e.g. Radiya-Dixit and Wang [2020]). Other works can be classed as *low-rank* methods, building on the fact that the overall fine-tuning parameter updates have a low intrinsic dimension (Aghajanyan et al. [2020]; Li et al. [2018]). Recent works such as Hu et al. [2021] and Mahabadi et al. [2021] achieve strong performance by injecting trainable low-rank

task-specific weight matrices. These methods have the additional benefit of not adding much inference latency; however, they still do not enable efficient batching of inputs across tasks.

***Fixed-LM* methods**

Another line of research considers keeping the weights, ϕ, of the original pre-trained LM_ϕ, completely unchanged. For example, Zhang et al. [2020a] trained a light-weighted side network, fused with the base LM_ϕ. Adapter-tuning (Rebuffi et al. [2017]; Houlsby et al. [2019]) inserts small bottleneck task-specific layers between frozen pre-trained network layers. Houlsby et al. [2019] achieved comparable performance to BERT-Large on important NLU benchmarks (Lin et al. [2020]), whilst only appending 2-4% additional parameters. Mahabadi et al. [2021] is both a *low-rank* and adapter-tuning technique, which adds only 0.047% of a pre-trained model's parameters and performs comparably to fine-tuning on GLUE (an NLU benchmark).

For comparison PREFIX-TUNING and prompt-embedding tuning (*prompt learning* techniques we consider in the next section) can achieve comparable performance to fine-tuning on certain tasks by appending 0.01-2% task-specific parameters [1]. Although adapter-tuning provides a parameter efficient way to inject vital knowledge, the technique dictates that batching cannot occur across tasks; for the insertable bottleneck layer alters the underlying structure of LM_ϕ. This ability to batch across task-specific parameters is a capability present with PREFIX-TUNING that we utilize in this study for *controlled generation*.

3.2 Prompt Learning

The nascent field of *prompt learning* (Liu et al. [2021a]) has been propelled on the back of excitement surrounding GPT-3 (Brown et al. [2020]). This section introduces the field and gives a formal description of two *prompt learning*[2] techniques: prompt-embedding tuning and PREFIX-TUNING —the latter forms the foundation for our method CONTROL PREFIXES. We also demonstrate the inter-relation of these two techniques, which is relevant for the discussion of a simpler method we propose: 'PREFIX-TUNING + control tokens'.

Manual Text Prompts

A frozen GPT-3's behaviour can be conducted by short text prompts: a series of tokens $P = \{p_0, p_1, ..., p_n\}$—typically a task description with several canonical examples. P is prepended to the input X, so that the model maximizes the likelihood: $p_\phi(Y \,|\, [P;X])$, whilst keeping ϕ fixed. These prompts are manually designed to steer the generation to desired tasks. Schick and Schütze [2020] demonstrate the effectiveness of predefined prompt templates in few-shot learning and text classification settings. The advantage of this method is that it is *fixed-LM* and only requires a suite of appropriate prompts to solve a great number of tasks

[1] The tasks trialled were Data-to-Text for PREFIX-TUNING and the SuperGLUE benchmark (NLU) for prompt-embedding tuning.

[2] For a recent survey on *prompt learning*, we refer the reader to Liu et al. [2021a].

(Brown et al. [2020]; Sun et al. [2021]). Although this approach reuses a frozen model, there are notable caveats:

- Creating these prompts is time-consuming and takes experience, particularly for some complicated tasks such as semantic parsing and can fail to find the optimal prompt (Shin et al. [2020]).
- Transformers can only condition on a bounded-length context; therefore, this hinders a pre-trained model's capacity to adapt to a task.
- Task description is error-prone and requires human involvement; manual prompts are specific to large models such as GPT-3 and do not generalize to smaller models such as BART.
- The downstream task performance of these models on benchmarks such as SuperGLUE lags behind that of tuned models (Lester et al. [2021]).

3.2.1 Prompt Engineering

The limitations outlined above have stimulated the rise of prompt engineering, which is the process of creating a prompting function that results in proficient performance on the downstream task. One design consideration that we consider in this work is whether the prompting function is *static*, using essentially the same prompt for each input in a dataset, or *dynamic*, generating a custom prompt for each input (Liu et al. [2021a]). Prompts can also be either discrete or continuous.

AutoPrompt (Shin et al. [2020]) is an example of optimising *discrete prompts*: it searches for discrete trigger words in the embedding space to concatenate with each input, evoking sentiment or factual knowledge from a masked language model. However, *continuous prompts* (soft prompt vectors), as well as being more expressive, enable direct gradient-based optimization. Subramani et al. [2020] revealed a pre-trained LM can reconstruct arbitrary sentences by optimizing a continuous vector for each sentence. One drawback is that a continuous prompt vector is less interpretable than prompts pertaining to discrete tokens. On the other hand, one can argue the discrete tokens as used in AutoPrompt can also be seen as uninterpretable—for they rarely relate to a human-readable ordering of words.

3.2.2 Prompt-Embedding Tuning

Several successive works: Logeswaran et al. [2020], Liu et al. [2021b] (P-tuning), and Lester et al. [2021] (prompt-tuning), share the similar idea of explicitly training continuous prompt embeddings. We class these methods as 'prompt-embedding tuning' (i.e. tuning *soft prompts*) and consider the *fixed-LM* case for the encoder-decoder. Lester et al. [2021] found prompt-embedding tuning on the NLU benchmark: SuperGLUE becomes more impressive with scale, and directly comparable to fine-tuning with language models ($>$ 10B parameters),

such as T5 XXL. Yet, the technique only uses 0.01% additional parameters whilst being *fixed-LM*. This result may be indicative of the low intrinsic dimension of the task adaptation update for an LM shown by Aghajanyan et al. [2020].

It should be noted Lester et al. [2021] executed a further pre-training stage with a prompt LM objective [3]. This type of priming may be an interesting research direction as the field of *prompt learning* matures. Figure 3.2.2 depicts how the gap between fine-tuning and prompt-embedding tuning narrows as the size of the Pre-trained LM increases.

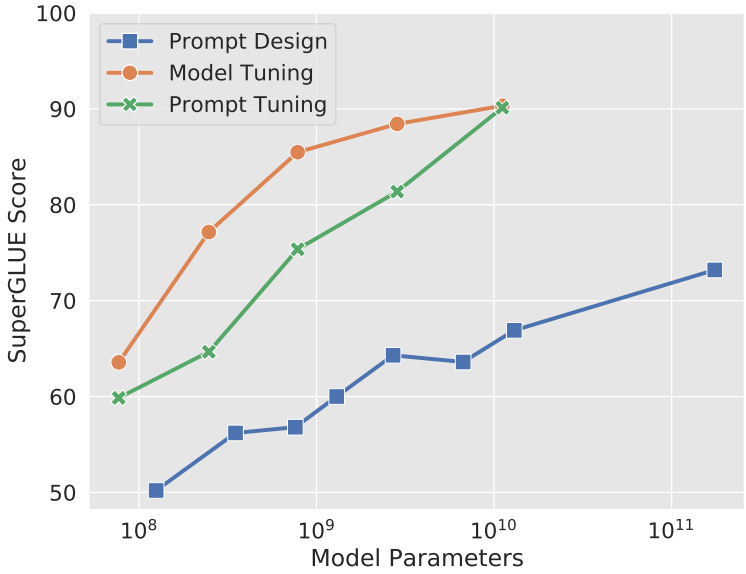


Figure 3.1: The x-axis is the total number of model parameters on a log- scale. For example, each plot of Model Tuning is determined by a size of T5 (eg. the right most plot being T5 XXL). **Model Tuning** refers to fine-tuning of T5-based models, and achieves strong performance but requires storing separate copies of the model for each end task. **Prompt Tuning** refers to prompt-embedding tuning of T5-based models, and matches the quality of fine-tuning as LM size increases. **Prompt Design** refers to few-shot prompt-design using GPT-3, the right most plot is GPT-3 175B. Source: Lester et al. [2021].

Formal Description of Prompt-Embedding Tuning

The pre-trained embedding matrix of the fixed model, $E \in \mathrm{LM}_\phi$, is used to map $[X,Y]$ to $[\mathbf{e}(X),\mathbf{e}(Y)]$. The idea of prompt-embedding tuning is to steer a frozen LM_ϕ by prepending a sequence of continuous embeddings to the embedding sequence [4] as so:

$$\mathbf{p_0},\ldots,\mathbf{p_i},\mathbf{e}(X),\mathbf{p_{i+1}},\ldots,\mathbf{p_m},\mathbf{e}(Y) \tag{3.1}$$

where $\{\mathbf{p_i} \,|\, i \in \{1,\ldots,N\}\}$ are trainable embedding vectors. This sequence is then processed by LM_ϕ—therefore, the *soft prompts* do limit the model's usable sequence length. Finally,

[3] This adaptation step is still generalist, transforming T5 XXL into a model more similar to GPT-3—a "few-shot learner")—see Lester et al. [2021] for full details.

[4] Lester et al. [2021] simplify the recipe by dispensing with the decoder prefix $\mathbf{p_{i+1}},\ldots,\mathbf{p_m}$.

with the downstream loss function $\mathcal{L}$ (2.2), optimization can be achieved by gradient descent:

$$\mathbf{p}^*_{\mathbf{0:m}} = \underset{\mathbf{p_{0:m}}}{\operatorname{argmin}} \ \mathcal{L}(LM_\phi(\mathcal{X},\mathcal{Y};\mathbf{p_{0:m}})).$$

There are variants on how to initialize each prompt. Many approaches initialize the prompt embedding with embeddings drawn from the model's vocabulary relevant to the task of interest. The intuition here being that these continuous prompts regulate the frozen model, akin to the textual context seen during pre-training.

3.2.3 Prefix-Tuning

In this section we give explicit description of PREFIX-TUNING[5], before demonstrating how prompt-embedding tuning fits in a PREFIX-TUNING framework.

Li and Liang [2021] [6] discovered that prompt-embedding tuning was inferior in terms of performance to PREFIX-TUNING for text generation in their supplementary experiments; reasoning that this is due to the inferior expressiveness of prompt-embedding tuning. In essence, PREFIX-TUNING has constituents of the prompt at every layer of an LM, endowing more degrees of freedom to control an LM. These constituents can steer the LM at every layer, rather than only being able to steer an LM through the input.

In PREFIX-TUNING instead of the base LM_ϕ computing activations at each layer, propagated from the original prompt-embedding $\mathbf{p_i}$, *trainable* key-value pairs are prepended to every layer. These pairs are fixed across examples at every layer. This does increase the number of parameters for a given prompt/prefix length; but in doing so, Li and Liang [2021] argues that this magnifies the ability of LM_ϕ to be adapted to a downstream task.

Formal Description of PREFIX-TUNING

Let us consider $\text{BART}_{\text{LARGE}}$ where $d_q = d_v = d = 1024$ (eqn 2.8), and there are the same number of decoder and encoder layers, L (Fig 2.1). Let ϕ refer to the parameters of the frozen LM_ϕ, and θ refer to the final inference PREFIX-TUNING parameters, and $\tilde{\theta}$ refer to the training parameters, where $\theta(\tilde{\theta})$, $|\theta| < |\tilde{\theta}|$.

The prefix, parameterized by θ, can be viewed as a set of trainable matrices for each class of attention: self-attention in the encoder (**E**) and cross-attention (**Dc**) and masked-attention

[5] Following the original implementation for $\text{BART}_{\text{LARGE}}$ by Li and Liang [2021], found here `github.com/XiangLi1999/PrefixTuning` for the encoder-decoder. There are some details omitted in the original paper, which focused mostly on PREFIX-TUNING for the *decoder-only* model, GPT-2.

[6] At the time of writing, the only other paper that utilizes PREFIX-TUNING is Hu et al. [2021] in their comparison studies, using the *decoder-only* LMs: GPT-2 and GPT-3. Perhaps symptomatic of the rapidly evolving nature of *prompt learning* in 2021, several recent works mis-characterize PREFIX-TUNING as prompt-embedding tuning (e.g. Mahabadi et al. [2021] refer to prompt-embedding tuning as the successor to PREFIX-TUNING).

(**Dm**) in the decoder:

$$P_\theta = \{P_\theta^E, P_\theta^{Dc}, P_\theta^{Dm}\}, \tag{3.2}$$

where $P_\theta^E, P_\theta^{Dc}, P_\theta^{Dm} \in \mathbb{R}^{\rho \times \omega}$. The dimension ρ is the prefix length, and $w = 2dL$, ρ is a model parameter that determines the length of the prefix of key-value pairs[7].

Here we use the notation $\left(K_i^{(l)}, V_i^{(l)}\right)$ to denote all key-value pairs from the l-th layer of the attention mechanism: $i \in \{E, Dc, Dm\}$. In the normal LM forward pass: $K_i^{(l)}, V_i^{(l)} \in \mathbb{R}^{M \times d}$, whereas with PREFIX-TUNING: $K_i^{(l)}, V_i^{(l)} \in \mathbb{R}^{(M+\rho) \times d}$, where M is the length of the sequence corresponding to the keys and values being considered.

Every row of P_θ^E is used to augment $[(K_E^{(1)}, V_E^{(1)}), \cdots, (K_E^{(L)}, V_E^{(L)})]$ in the encoder: i.e. the first indexed key-value pair in every layer is drawn from the first row of P_θ^E. This happens for the first ρ indexed key-value pairs in every layer, they are drawn from each of P_θ^E's respective rows. Similarly, the ρ rows of $P_\theta^{Dc}, P_\theta^{Dm}$ are used to augment the respective $[(K_{Dc}^{(1)}, V_{Dc}^{(1)}), \cdots, (K_{Dc}^{(L)}, V_{Dc}^{(L)})]$ and $[(K_{Dm}^{(1)}, V_{Dm}^{(1)}), \cdots, (K_{Dm}^{(L)}, V_{Dm}^{(L)})]$, so the first ρ indexed key-value pairs in every layer are trainable vectors extracted from these matrices. This means there is both a prefix of ρ key-value pairs always in the left context of the encoder and always in the left context of the decoder for both attention mechanisms.

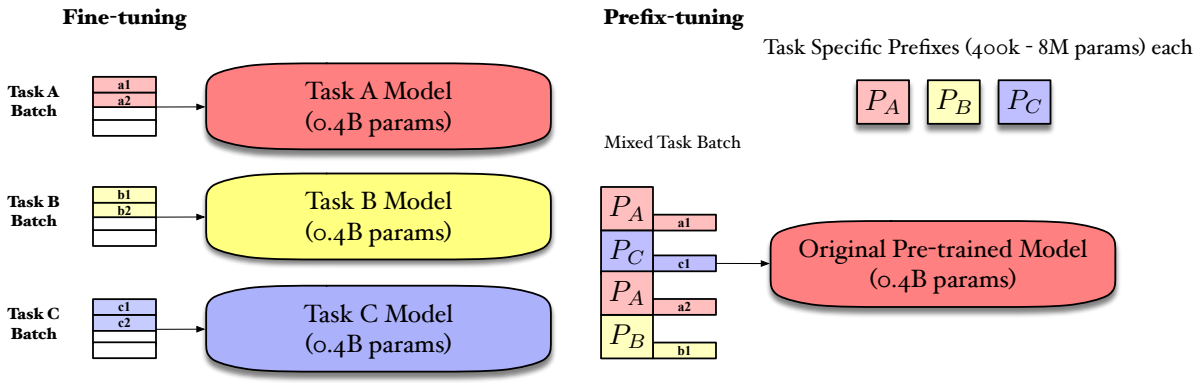


Figure 3.2: Schematic demonstrating the benefits of prefix-tuning for a PLM, (# parameters are shown for $\text{BART}_{\text{LARGE}}$) in a multi-task setup. Left: **Fine-tuning** requires multiple task-specific $\text{BART}_{\text{LARGE}}$ models for each respective task. Right: **PREFIX-TUNING**, reuses the original pre-trained $\text{BART}_{\text{LARGE}}$ checkpoint, only training prefixes P_A, P_B, P_C, and is a *fixed-LM* method. For example, P_A, relates to training a set of $\{P_A^E, P_A^{Dc}, P_A^{Dm}\}$ matrices. Despite involving modifications to all attention mechanisms in all layers, PREFIX-TUNING can be batched, unlike adapter tuning.

[7] ρ does not have to be the same for each class of attention. For example there can be no decoder prefix matrices. However, in this work we consider the case where ρ is the same across each class of attention.

Over-parameterization Trick

To stabilize optimization, an important step is carried out which results in a significant increase in the trainable parameters: $\tilde{\theta}$. This is achieved by re-parameterizing P_θ^i as:

$$P_\theta^i = \eta^i(P_{\tilde{\theta}}^i), \quad \forall i \in \{E, Dc, Dm\}, \tag{3.3}$$

where $P_{\tilde{\theta}}^i \in \mathbb{R}^{\rho \times k}$ is a smaller matrix ($k < \omega$). $\eta^i(\cdot)$ is a two layered large MLP:

$$\eta^i(\cdot) = \Big(\sigma\Big((\cdot)W_{\tilde{\theta},1}^i + B_{\tilde{\theta},1}^i\Big)\Big)W_{\tilde{\theta},2}^i + B_{\tilde{\theta},2}^i, \tag{3.4}$$

where $W_{\tilde{\theta},1}^i, B_{\tilde{\theta},1}^i \in \mathbb{R}^{k \times d_m}$, $W_{\tilde{\theta},2}^i$, $B_{\tilde{\theta},2}^i \in \mathbb{R}^{d_m \times \omega}$ and $B_{\tilde{\theta},1}^i, B_{\tilde{\theta},2}^i$ are formed from respective bias vectors, and σ is an activation function[8]. The dimensions k and d_m, can be considered training specific parameters. The idea is to augment the trainable parameters significantly—in this study we use $k = d = 1024$ and $d_m = 800$ for all models[9]. Whereas $w = 2dL$, so $w = 24{,}576$ for $\text{BART}_{\text{LARGE}}$ and $w = 49{,}1526$ for T5-large[10]. Once training is complete, the final P_θ^i can be saved for use at inference and the re-parameterization parameters ignored.

Shared Re-parametrizaton

There is no concrete theoretical justification for why the over-parameterization is important: recent work by Buhai et al. [2020] show that over-parameterizing can artificially smooth the optimization landscape.

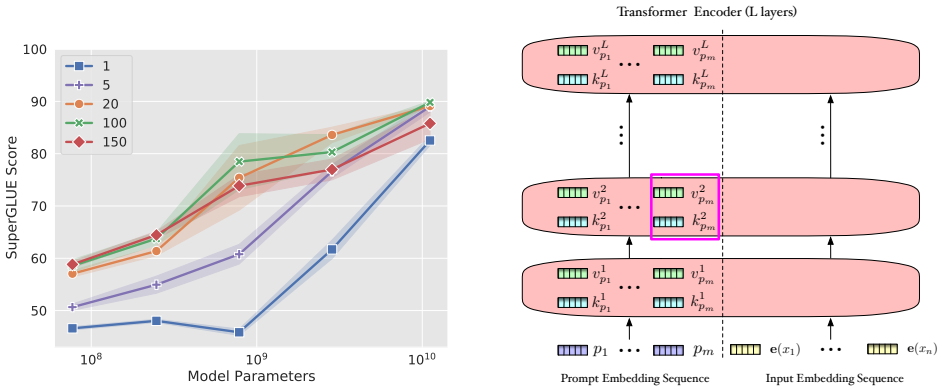
In this work we further explore these re-parameterizations through CONTROL PREFIXES. Consider one element of the prefix for one attention mechanism (pertaining to the same row of P_θ^i and $P_{\tilde{\theta}}^i$) . We argue having $d_m < k$ (eqn 3.4), which constrains each element to lie in a bottleneck representation and then having a shared, large learned non-linear η^i to map this representation to L key-value pairs: P_θ^i, compels each element to coordinate steerability for that mechanism. For example, the rows of P_θ^{Dm}, may lie in a vector space better coordinated for steering the output sequence than the vector space P_θ^E sits in, by virtue of η^E learning a mapping coordinated for moderating the processing of the input sequence, X.

Generalizing Soft Prompts with PREFIX-TUNING

Fig. 3.3: **(a)** (Lester et al. [2021]) illustrates that for prompt-embedding tuning, past a certain prompt length, m, *performance saturates*. It does not matter how many additional parameters are added by increasing prompt length. At this point we can increase the steerability by overwriting any selection of key-value pairs that pertain to a prompt-embedding index with

[8] We use tanh for the activation function σ.
[9] These are the same values used by Li and Liang [2021] for $\text{BART}_{\text{LARGE}}$ applied to the task of Abstractive Summarization.
[10] L=12 for $\text{BART}_{\text{LARGE}}$ and L=24 for T5-large.

trainable vectors. Therefore, they are not constrained by the model. The unselected key-value pairs can be computed by LM_ϕ. Any selection of key-value pairs in the encoder and decoder can be overwritten with trainable vectors. A depiction of this is shown in Fig. 3.3: **(b)**.



(a) Altering Prompt Length **(b)** Prompt-embedding Tuning

Figure 3.3: (a) Depicts experiments conducted by Lester et al. [2021] (the source of the figure)—for a certain T5 model size (x-axis), various prompt lengths are trialled (shown in the figure legend). Performance degrades past a certain length, m. **(b)** Illustrates how prompt-embedding tuning can be viewed as part of a more general PREFIX-TUNING framework. For simplicity, only the encoder of the frozen LM_ϕ is shown. The key-value pairs in each layer, corresponding to the indexed prompt vectors are highlighted. We could overwrite the key-value pair shown in the pink square with two trainable vectors, and train these in addition to $\mathbf{p_{0:m}}$. This key-value pair would not be constrained by the model, enhancing the degree of control.

This in theory only expands the capability of the model, providing additional degrees of freedom, while maintaining the NLU of the frozen LM_ϕ. PREFIX-TUNING is the extreme case whereby, all key-value pairs in every layer are overwritten, and none are constrained by the model. This also means that the key-value pairs in the cross-attention layer in the decoder are not an identical contextualized representation for each layer. This general case also has the added benefit of not restraining the model's usable sequence length, by only augmenting K,V. Hu et al. [2021] cite PREFIX-TUNING as reducing the LMs usable input length; however, this is clearly not the case for the implementation under discussion.

Additional benefits of PREFIX-TUNING and Prompt-embedding Tuning

Dataset complexity measure In the task sub-space we can measure the capacity required to capture a specific task by varying prefix length, akin to experiments conducted by Aghajanyan et al. [2020]. This gives us an idea of the task and dataset complexity, whilst permitting us to appreciate which classes of task require more training data or greater capacity PLMs. Li and Liang [2021] and Lester et al. [2021] both found *performance saturation* or even *performance degradation* after increasing the respective ρ and the prompt-length to a certain threshold.

Examining the task sub-space may also allow us to measure the similarity of labelled datasets, identifying which tasks complement one another.

Domain shifts and low-data regime Freezing the parameters of the base LM restricts the model from altering its general understanding. This reduces the model's ability to overfit a dataset by memorizing specific lexical cues. This suggests why Li and Liang [2021] and Lester et al. [2021] reported PREFIX-TUNING and prompt-embedding tuning as more robust to domain shifts—where the distribution of inputs varies between training and evaluation; as well as performing better in the low data regime. Adapter tuning also has this benefit, but does not leave the original language model as intact by interleaving the structure with the adapter modules.

3.3 Controlled Text Generation

A highly related field to *prompt learning* is *controlled generation*, which aims to incorporate various types of guidance signal at the input-level of the model, while the underlying task remains the same. This section introduces related work in *controlled generation*, motivating CONTROL PREFIXES which is at the intersection of *controlled generation* and *fixed-LM prompt learning*. A recent survey on *prompt learning* (Liu et al. [2021a]) highlights the dearth of research in this intersection, and how valuable this research could be for the future of *prompt learning*, citing Tsimpoukelli et al. [2021][11] as a sole example of such work.

Guidance Signals

The guidance signals can be length specifications (Kikuchi et al. [2016]), or even highlighted phrases or sentences (Grangier and Auli [2018]) to plan the content of generated texts. This control can happen at decoding time, for example by guided weight decoding (GeDi, Krause et al. [2020]).

GSum (Dou et al. [2020]) employ a *prompt + fine-tuning* strategy: both the prompt's and pre-trained LM's parameters are tuned. The authors could guide $\text{BART}_{\text{LARGE}}$ successfully for Abstractive Summarization using oracle extracted sentences as guidance. However, this has the undesirable quality that the $\text{BART}_{\text{LARGE}}$ checkpoint weights do not remain fixed.

Control Tokens

Control tokens have been a popular way to extend control, even before the advent of the Transformer. Control tokens are artificial tokens that do not relate to a real word. The resultant embeddings are trained to indicate conditional information to the model. For example, in Machine Translation, Johnson et al. [2016] trained a multilingual model and showed the efficacy of adding a control token to encode the target language at the beginning of each source sentence. CTRL (Keskar et al. [2019] (1.63B parameters), the largest LM at

[11] The authors train a vision encoder to represent each image as a sequence of continuous embeddings, so that a pre-trained, *fixed-LM* prompted with this prefix generates the appropriate caption.

the time, was trained alongside conditional control tokens and demonstrated these learnt to govern style, content, and task-specific behaviour.

These examples are not *fixed-LM* techniques—the whole underlying LM can adapt alongside the control tokens. This is a crucial distinction to make as in our supplementary experiments, where we implement PREFIX-TUNING + control tokens models, the underlying LM remains fixed.

Prompt Learning* and *Controlled Generation

In contrast with *controlled generation*, the main motivation for using prompts for text generation is to specify the task itself and better utilize the pre-trained model. Thus, there are commonalities in these two genres: both add extra information to the input text to ameliorate generation and these additional signals are usually learnable parameters.

This thesis wants to explore if *controlled generation* can be applied to a parameter efficient *fixed-LM* method, where the parameters of the underlying LM cannot themselves adapt to varying conditional information. Input-level control has to be extended via the small number of input-level parameters and additional task-specific parameters only.

Input vs. Target Control

It is important to differentiate if the control is extended over the input or the target generation. In Rabinovich et al. [2016] the authors use domain adaptation techniques to help retain the original gender traits in machine translation—this is an example of *input-level* control. If the control concerns the *output* such as length specification, evaluation can be especially challenging as a lot of these techniques involve oracle information which is not present during evaluation. Therefore, the methods have to rely on additional models to predict the correct guidance information.

Control over the *input* and control over the *target* are both considered in this work.

3.3.1 Auxiliary Scaffold Tasks

Incorporating auxiliary scaffold tasks via multitask learning has been studied before for improving span-labeling and text classification (Swayamdipta et al. [2018]; Cohan et al. [2019]). Cachola et al. [2020] demonstrate that control tokens can be used to effectively incorporate scaffold tasks (i.e. title generation tasks) in conjunction with the main task (i.e. TL;DR summary) for $\text{BART}_{\text{LARGE}}$. Inspired by this form of data augmentation, we apply a similar procedure with CONTROL PREFIXES when using a Data-to-Text dataset formed from an accumulation of heterogeneous Data-to-Text sub-datasets. We provide guidance to the model on which sub-dataset the training example belongs, in order for the model to better leverage these heterogeneous sources.

Section 4: CONTROL PREFIXES

This section provides a formal description of our principal method, CONTROL PREFIXES. This section relies on details introduced in Section 3.2.3. CONTROL PREFIXES naturally extends the explicit framework used to describe PREFIX-TUNING for the encoder-decoder.

Problem Statement

To demonstrate how a parameter efficient[1], *fixed-LM* method such as PREFIX-TUNING can itself leverage sub-task information at the input-level effectively.

Intuition

CONTROL PREFIXES is inspired by how both traditional *controlled generation* and even word-embeddings, themselves, can guide an LM into generating very disparate target sentences. As NLG systems can be very poor at generating part of a target distribution, we want to investigate if PREFIX-TUNING can successfully leverage a guidance signal that is fed into the model alongside the source document X—in order to encourage more fine-grained control. We believe explicitly demarcating the parameters into "generalist parameters", "task-specific parameters" and "attribute-level parameters" has a range of benefits. The general task-specific parameters can themselves adapt to modular *control prefixes* which change along with the guidance signal, for each input X. For this work we only consider guidance signals that consist of discrete values for a particular categorical variable.

4.1 Description

The idea is to have a general task prefix P_θ ("task-specific parameters") which remains static, and train at the same time C_θ ("attribute-level parameters"), a set of prefixes that change depending on the input. This requires conditional information or guidance G, to be fed into the model in addition to the source document. G[2] indicates which *control prefixes* to use for a particular attribute. Let us consider the parallel corpus $\langle \mathcal{X}, \mathcal{Y}, \mathcal{G} \rangle$, where $G^j \in \mathcal{G}$ indicates all the conditional 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:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{\langle X^j, Y^j, G^j \rangle \in \langle \mathcal{X}, \mathcal{Y}, \mathcal{G} \rangle} \log p\big(Y^j \mid X^j, G^j; P_\theta, C_\theta, \phi\big). \tag{4.1}$$

[1] We use the term parameter efficient to denote methods adding <3% additional parameters to a fixed LM's parameters.

[2] We discuss cases where G is not present for a subset of examples later on.

CEFR Levels as Motivating Example

Let us consider Grammatical Error Correction (GEC) as a motivating example. In this work we approach GEC as a typical text-to-text task, albeit a machine translation[3] of non-standard language input (ungrammatical sentence), X, to standard language output (grammatical sentence) Y. We consider the case, as is with the W&I Corpus Dataset (Bryant et al. [2019], where the data is annotated with a proficiency level: the CEFR level, that can be A, B or C in order of increasing proficiency.

In this scenario, there are three *classes*[4] (A,B,C), for one *attribute*: CEFR level. Therefore G^j is just the CEFR level for that particular corpus pair $\langle X^j, Y^j \rangle$. In this simplistic case

$$C_\theta = \{C_{\theta,A}, C_{\theta,B}, C_{\theta,C}\}. \tag{4.2}$$

If we take $C_{\theta,A}$ for example, $C_{\theta,A}$ relates to three matrices for each class of attention (E, Dc, Dm). This is in the same manner as the matrices that make up P_θ:

$$C_{\theta,A} = \{C^E_{\theta,A}, C^{Dc}_{\theta,A}, C^{Dm}_{\theta,A}\} \tag{4.3}$$

$$P_\theta = \{P^E_\theta, P^{Dc}_\theta, P^{Dm}_\theta\}. \tag{4.4}$$

$P^E_\theta, P^{Dc}_\theta, P^{Dm}_\theta \in \mathbb{R}^{\rho \times \omega}$ is as defined in Section 3.2.3 for PREFIX-TUNING, where ρ is the length of the general task prefix that remains static for all examples. Whereas $P^E_\theta, P^{Dc}_\theta, P^{Dm}_\theta \in \mathbb{R}^{\rho \times \omega}$, $C^E_{\theta,A}, C^{Dc}_{\theta,A}, C^{Dm}_{\theta,A} \in \mathbb{R}^{\rho_c \times \omega}$ where ρ_c denotes the control prefix length, which is kept constant between classes in a given attribute. Typically $\rho_c < \rho$, and $\omega = 2dL$ as before. The other classes $C_{\theta,B}$ and $C_{\theta,C}$ describe three distinct matrices each in the same manner.

4.1.1 Sharing Sub-spaces

For PREFIX-TUNING, the MLPs $\eta^E, \eta^{Dc}, \eta^{Dm}$ (3.3,3.4) are used in the re-parameterization of the general prefix:

$$P^i_\theta = \eta^i(P^i_{\tilde{\theta}}), \quad \forall i \in \{E, Dc, Dm\}, \tag{4.5}$$

where $P^i_{\tilde{\theta}} \in \mathbb{R}^{\rho \times k}$ and $k < \omega$. These same MLPs are shared amongst *all* CONTROL PREFIXES. For example, if we take $C_{\theta,A}$:

$$C^i_{\theta,A} = \eta^i(C^i_{\tilde{\theta},A}), \quad \forall i \in \{E, Dc, Dm\}, \tag{4.6}$$

where $C^i_{\tilde{\theta},A} \in \mathbb{R}^{\rho_c \times k}$ is a smaller matrix ($k < \omega$), k being the same outer dimension as $P^i_{\tilde{\theta}}$. Therefore, training one additional attribute class for control prefixes amounts to only an additional $k\rho_c$ training parameters.

[3] Many of the top-performing systems for GEC borrow ideas originally developed for machine translation (Bryant et al. [2019]; Stahlberg and Kumar [2021]; Takahashi et al. [2020]).

[4] We use this terminology in order to avoid confusion with WebNLG categories, which are used as a conditional attribute in Section 6.

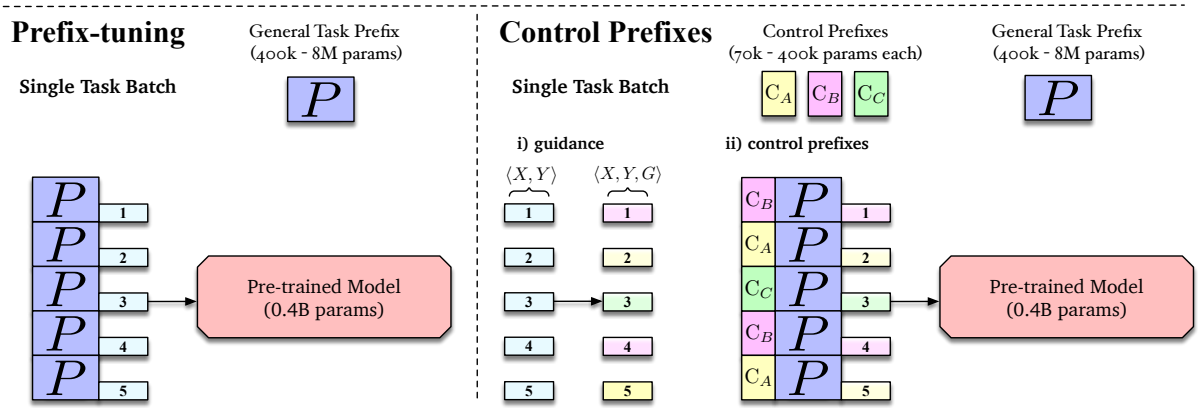


Figure 4.1: High-level diagram contrasting PREFIX-TUNING and CONTROL PREFIXES in the single-task setup described for GEC. The same single-task batch is considered for both setups, which is formed from examples 1,2,3,4 and 5. Left: PREFIX-TUNING has one general prefix P for all examples. Right: CONTROL PREFIXES utilizes additional conditional information at the input-level in **i)**, where this conditioning is indicated by the change in colour. 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. Note: the parameters θ are omitted to aid clarity.

As alluded to in 3.2.3, we believe these re-parameterizations are important for vanilla PREFIX-TUNING. For CONTROL PREFIXES, sharing these functions encourages each η^i to coordinate the final overall collection of prefixes in the subspace relating to the particular attention mechanism $i \in \{E, Dc, Dm\}$. For example, η^E compels P_θ^E and $C_{\theta,A}^E, C_{\theta,B}^E, C_{\theta,C}^E$ to learn to share properties to moderate the processing of the input sequence on the encoder side.

This belief is backed up by empirical observation. When using disjoint re-parameterizations for CONTROL PREFIXES, performance degrades and training instability arises. Disjoint re-parameterizations would also augment the number of training parameters $\tilde{\theta}$ significantly.

4.1.2 Generalizing to Other Attributes

One can easily see how this example of CONTROL PREFIXES generalizes to A[5] attributes.[6] As we consider discrete values of a categorical variable, each attribute $a \in \mathcal{A}$, possesses R_a classes. The control prefix length $\rho_{c,a}$ is kept constant between classes of an attribute. For example, the control prefixes for the classes of attribute a would be

$$C_{\theta,a} = \{C_{\theta,a,1}, \ldots, C_{\theta,a,R_a}\}, \tag{4.7}$$

where each of these elements consists of a corresponding set of three matrices for each attention mechanism. We run experiments with multiple attributes and with differing control prefix lengths.

[5] Here A denotes the number of attributes rather than the CEFR level A.
[6] In this case, $|G^j| = A$ for the sample $\langle X^j, Y^j, G^j \rangle$ of the corpus.

CONTROL PREFIXES provides a way to capture attribute-level inductive biases given a suitable guidance signal. This technique fundamentally depends on the strength of the conditioning information. The guidance signal for this implementation has to be the form of discrete categorical variables.

4.1.3 Controlling Prefix-Tuning

Control tokens (introduced in Section 3.3) can also be used as a means to condition on a guidance signal. Furthermore, as discussed in Section 3.2.3, PREFIX-TUNING can be regarded as a type of generalization of prompt-embedding tuning, possessing more degrees of freedom to steer a frozen LM. Similarly, if we take the *fixed-LM* and PREFIX-TUNING parts out of the equation, CONTROL PREFIXES can be viewed as a generalization of control tokens. It possesses many more degrees of freedom to guide an LM in light of a particular conditional attribute than control tokens. This could promote research into Prompt+LM fine-tuning methods, with CONTROL PREFIXES better able to integrate conditional information in the traditional scenario of LM fine-tuning.

PREFIX-TUNING + Control Tokens

We propose another architecture titled 'PREFIX-TUNING + control tokens', which in Table 6.4 and Appendix E we refer to as 'control tokens'. In PREFIX-TUNING + control tokens, all of the original LM parameters, ϕ, still remain fixed, including the Embedding matrix, E. We use this method to inform the model of the same discrete guidance information G, as in CONTROL PREFIXES. Only the embeddings pertaining to the controllable attributes and the prefix are trained. Therefore, there are $< 2\%$ additional parameters to the fixed LM.

In this study, we benchmark the top-performing CONTROL PREFIXES for each dataset against PREFIX-TUNING + control (except for Summarization). This alternative method is less expressive than CONTROL PREFIXES, so we argue that if the performance is greater with CONTROL PREFIXES than PREFIX-TUNING + control tokens, this gives indication that the conditional information is more complicated to leverage. One advantage of the PREFIX-TUNING + control tokens architecture over CONTROL PREFIXES is that it adds fewer parameters to the fixed LM, assuming the general task prefix remains the same size. How much less is determined by i) the control prefix lengths and ii) how many attribute classes there are in the downstream dataset.

Section 5: Experimental Setup

A principal objective of this thesis is to systematically evaluate the performance of CONTROL PREFIXES. In order to do this, we need an array of diverse datasets and evaluation measures, where we can robustly test generation performance. This section introduces the tasks and datasets used to evaluate our systems. Here we also provide *motivation* for the dataset-specific guidance information used for each CONTROL PREFIXES model.

Which datasets and tasks to evaluate on?

Section 2.4 motivated the importance of selecting datasets with robust measures and the nature of these measures. The datasets should be challenging, but feasible to evaluate on. For example, creative generation tasks like story generation and poetry generation, and even dialogue systems suffer from having an extreme number of plausible responses, making automated evaluation ineffective.

Therefore, our chief selection criteria were i) clean datasets, to remove the effect of confounding model mistakes with learned noise and ii) datasets with multi-references, and those shown to produce more robust automatic evaluation.

5.0.1 NLG Evaluation Frameworks

Two recent endeavours to mitigate the problems outlined in Section 2.4, as well as develop reproducible standards, are the GEM benchmark (Gehrmann et al. [2021][1])—a substantial coordinated NLG evaluation effort—and the GENIE (Khashabi et al. [2021]) platform, which aims to automate and standardize the human evaluation of NLG systems. Our work is conducted with much of the ethos outlined in these works[2]. This is also why we select five of the eleven GEM datasets, and also why we elected to submit our summarization model to the GENIE human evaluation platform.

[1]The organizers wish GEM to be an evolving benchmark. Although, for the five GEM datasets we use, the test sets are currently identical for GEM and for the official evaluation test sets, this may change. All our results are reported on the official test sets and official evaluation scripts; except for XSum, where no official evaluation framework exists.

[2]GEM is currently developing its own human evaluation approaches, using the infrastructure provided by GENIE to run its human evaluation.

5.1 Tasks & Datasets

5.1.1 Data-to-Text: Source & Category

The objective of Data-to-Text generation is to produce fluent text from structured input. Applications include weather forecasts (Konstas and Lapata [2012]) and summarization (Fan et al. [2019]). The datasets may take various formats; although, in this work we ensure all source formats are a set of RDF triples, called a *tripleset* (Gardent et al. [2017]). This means each dataset can be constructed as a *corpus* of <triplesets,sentences> pairs. An RDF triple is a set of three entities that codifies a statement about semantic data in the form of subject–predicate–object, for example: 'Serie A-champions-Juventus F.C' in Fig. 5.1.1.

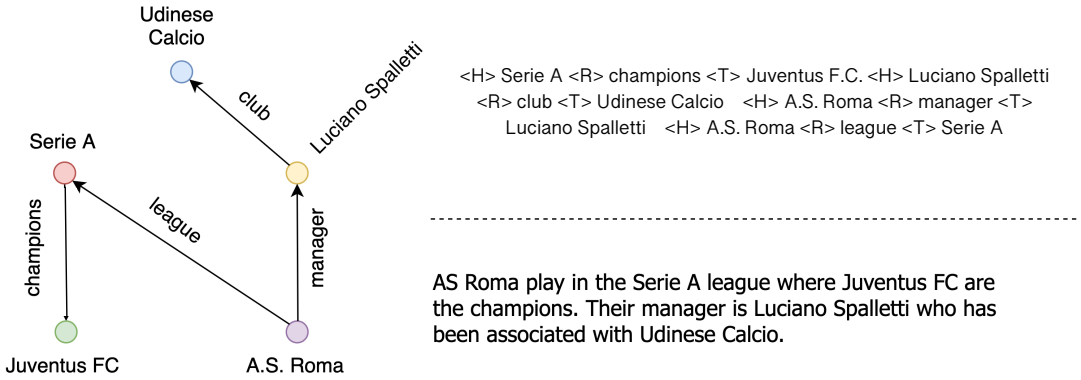


Figure 5.1: An example of the linearization procedure for a WebNLG 2017 example from the category *SportsTeam*. Left: The set of RDF triples (tripleset) arranged in a graph. Top right: The linearized input with <H>, <R>, <T> demarcation tokens. Bottom right: the target, i.e. a fluent textual description of the graph. Adapted from Kale [2020].

DART (Radev et al. [2020], a GEM dataset) is an open-domain, multi-source Data-to-Text corpus. Fig. 5.2 depicts the multiple sources: human annotation of Wikipedia tables; automatic extraction from WikiSQl; as well as two existing datasets, WebNLG 2017 and E2E Clean—these existing datasets are restricted to specific domains. In contrast, Radev et al. [2020] 's chief aim was to amass diverse predicates (Table 5.1) over a larger number of domains. We supply details on the nature of triplesets for each DART sub-dataset source.

Note: The DART sub-dataset sources are particularly important for our work as we use each source as a conditional guidance signal in various model setups that use DART for training.

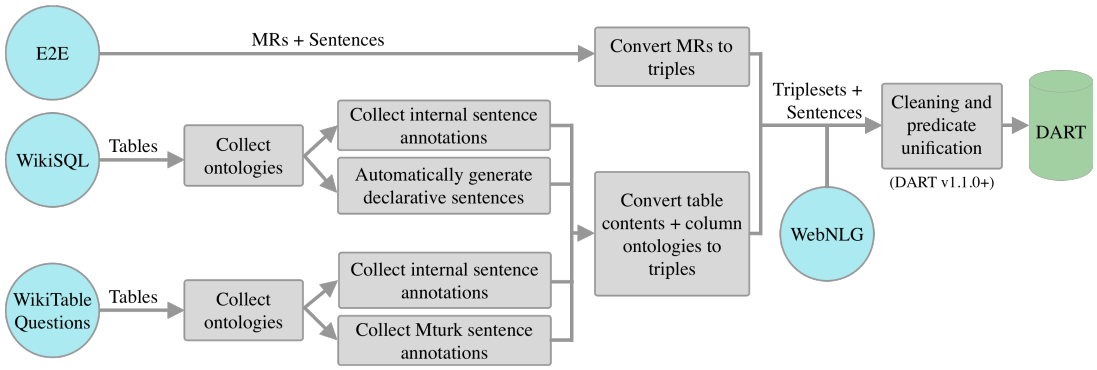


Figure 5.2: Data collection pipeline for DART. Source: Radev et al. [2020].

	DART: 62,659 train / 6,980 dev / 12,552 test					
	WikiTableQuestions		WikiSQL		WebNLG 2017	E2E Clean
	Internal	MTurk	Internal	Declarative		
Domains	Wikipedia (open-domain)				DBPedia Categories (Seen: 10, Unseen 5)	Restaurants
Unique Predicates	1,950	1,403	493	2,008	347	7
Unique Triples	13,505	5,541	1,648	7,787	3,220	946
Tripleset-Sentence Pairs	4,902	2,120	772	4,204	27,731	42,462
Triples per Tripleset (min, med, max)	1, 3, 10	1, 3, 7	1, 2, 7	1, 2, 10	1, 3, 7	1, 4, 7
Vocab Size	13.4K	8.9K	3.0K	10.7K	8.0K	3.0K
Words per SR	15.2	16.5	14.0	12.6	22.5	22.9
Sentences per SR	1.0	1.1	1.0	1.0	1.4	1.6

Table 5.1: Detailed statistics of the constituents of DART, partitioned by the different collection methods. DART exhibits a great deal of topical variety in terms of the number of unique predicates, the number of unique triples, and the vocabulary size. Note the number of unique predicates of the existing datasets WebNLG and E2E Clean are lower than the Radev et al. [2020] curated triplesets, despite having a greater number of <tripleset, sentence> pairs. Adapted from: Radev et al. [2020].

In addition to experiments on DART, we run experiments on the **E2E Clean** dataset. This dataset was procured by Dušek et al. [2019] via automatically fixing the dialogue acts to account for omissions and hallucinations in the text of the original E2E dataset (Novikova et al. [2017]). All examples are from the restaurant domain, where the source is in the form of a meaning representation (MR). An example is as follows:[3]

Source: name[Cotto], eatType[coffee shop], food[English], customer_rating[low], area[city centre], near[The Portland Arms]

Gold Reference: Cotto is a cheap coffee shop with one-star located near The Portland Arms.

WebNLG Datasets

Each instance of WebNLG 2017 (Gardent et al. [2017]) contains a tripleset from DBPedia (Auer et al. [2007]) and a target text with one or multiple sentences that describe the tripleset. 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. Fig.

[3]For details on what exactly an MR is and the MR-to-tripleset conversion process, see Radev et al. [2020]. For the two E2E Clean test set examples excluded from DART and lacking a NAME slot, we convert the slot to an empty string.

5.1.1 depicts an example from the WebNLG 2017 training set for the category *SportsTeam*. These categories are used as guidance signal in our experiments, other categories include *Airport*, *City* and *Food*.[4]

WebNLG+ 2020 (a GEM dataset) is not a constituent 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 2017, but with *new* examples), alongside three unseen categories. The authors split the dataset into *Unseen* for samples from unseen categories. However, examples from seen categories are further split into *Seen* (examples based on seen categories and seen entities during training) and *Unseen Entities* (examples based on seen categories, but unseen entities). Table 5.2 shows a comparison of the WebNLG 2017 and WebNLG+ 2020 test splits.

	WebNLG+ 2020	WebNLG2017
Seen categories	490(28%)	971(52%)
Unseen categories	896(50%)	891(48%)
Unseen entities	393(22%)	-
Total	1,779	1,862

Table 5.2: Comparison of test set category composition for WebNLG 2017 and WebNLG+ 2020.

Data-to-Text Guidance

DART sub-dataset Source as guidance

Radev et al. [2020] revealed fine-tuning T5 on the WebNLG 2017 dataset with dart_human_annotated (all human annotated components of DART in Fig. 5.2, Table 5.1), achieved SOTA performance on WebNLG 2017. Using the whole DART dataset was not as effective—nevertheless, this inspired the idea of using all the sources of DART as a data augmentation strategy with CONTROL PREFIXES as discussed in Section 3.3.1, to better leverage these heterogeneous sources.

Can the WebNLG DBpedia Categories guide CONTROL PREFIXES to produce better textual representations of triplesets?

Neural pipeline models (Moryossef et al. [2019]; Castro Ferreira et al. [2020]) achieve strong performance in the *unseen* portion of WebNLG 2017. On the other hand, fully end-to-end models (Ribeiro et al. [2020]; Schmitt et al. [2020]) exhibit strong performance on the *Seen* data and usually perform poorly on *Unseen* data. This disparity in performance on different categories encouraged the idea of providing the category explicitly as guidance, in the hope that each control prefix could capture inductive biases relating to properties of triplesets belonging to a specific WebNLG category.

[4]Every training category label can be seen in Fig. 7.2, where we visualize control prefixes corresponding to each training category.

We run experiments on the two official WebNLG competitions: WebNLG 2017 (Gardent et al. [2017]) and WebNLG+ 2020 (Castro Ferreira et al. [2020]), a GEM dataset). A central objective of this thesis is to focus on whether control prefixes are interpretable and if zero-shot learning is possible. Therefore, in Section 7 we visualize the control prefixes corresponding to each training category for both WebNLG 2017 and WebNLG+ 2020 in Fig. 7.2.

Examples in addition to Fig. 5.1.1 of the linearized inputs, gold references, WebNLG categories and model outputs for both WebNLG 2017 and WebNLG+ 2020 can be found in Table 7.2, and the Appendix C.0.2.

5.1.2 Grammatical Error Correction

We revisit GEC, introduced in the motivating example for CONTROL PREFIXES in Section 4.

	A	B	C	N	Total
Train					
Texts	1,300	1,000	700	–	3,000
Sentences	10,493	13,032	10,783	–	34,308
Tokens	183,684	238,112	206,924	–	628,720
Dev					
Texts	130	100	70	50	350
Sentences	1,037	1,290	1,069	998	4,384
Tokens	18,691	23,725	21,440	23,117	86,973
Test					
Texts	130	100	70	50	350
Sentences	1,107	1,330	1,010	1,030	4,477
Tokens	18,905	23,667	19,953	23,143	85,668

Table 5.3: The granular CEFR level statistics for the W&I dataset. The N (native samples) are from the LOCNESS corpus which is used as part of the W&I test set. Source: Bryant et al. [2019]

The BEA-2019 task (Bryant et al. [2019]) is a popular task to benchmark GEC performance. In the restricted track, participants are permitted to use a specific set of annotated corpora including W&I as annotated training sources. This does not prohibit the use of large synthetically generated datasets[5]—which is an important ingredient for top-performing GEC systems (e.g. Takahashi et al. [2020]). Although pre-trained LMs have been exposed to large amounts of text, the LM has not specifically been exposed to the gamut of error types in the input of GEC test sets. On the other hand, synthetic data has been found to cover a wide range of error types (Stahlberg and Kumar [2021]).

Can the CEFR level guide CONTROL PREFIXES to improve error correction?

Unlike the rest of the restricted track, W&I Corpus has CEFR level annotations: A, B, C and the test set (BEA-test) has an additional N proficiency level for native level text (Table 5.3). By only training with W&I Corpus, a direct assessment can be made of any advantage of explicitly guiding CONTROL PREFIXES with the proficiency level of the input. ERRANT (Bryant

[5] Even the low resource track does not forbid synthetic data. We did not experiment with synthetic data due to lacking sufficient computational resource.

et al. [2017]) is an automatic scorer which scores through edit overlap, in terms of both its token offsets and correction string, producing a $F_{0.5}$ value[6].

The original purpose of the CEFR annotations were to enable granular analysis of participating systems in the shared task—and it was found that some systems were much more biased towards different CEFR levels than others (Bryant et al. [2019]). While punctuation errors are rare at levels A and B, they are much more common at levels C and N. Conversely, noun number errors are common at levels A and B, but are rarer at levels C and N. These findings encourage using CONTROL PREFIXES with the CEFR level, and that CEFR levels correspond to very distinct error-tag distributions.

In addition to the small training set, a limitation in our study is that CEFR levels are originally assigned at the essay level. This means sentence level CEFR labels are an approximation and it is possible that the same sentence might receive a different label in a different text (Bryant et al. [2019]).

5.1.3 Summarization

Text Summarization is the task of generating accurate and concise summaries from input document(s). Extractive Summarization merely reproduces and stitches together informative fragments from the input. Conversely, Abstractive Summarization may generate novel phrases and sentences. A good abstractive summary captures the salient information of the input and is linguistically fluent.

We report results on the XSum dataset (Narayan et al. [2018], a GEM task), which comprises 226,711 British Broadcasting Corporation (BBC) articles coupled with their single sentence summaries. The dataset is split into three subsets: training (90%, 204,045), validation (5%, 11,332), and test (5%, 11,334) sets. Each article has an average length of 431 words, whilst the hand written summaries have a length of 23.3 words. This extreme summarization—by virtue of condensing the summary into one sentence—is not compliant to extractive strategies, so is a good test of generative systems.

Can the BBC News URL Sub-directories guide CONTROL PREFIXES to produce better summaries?

Each XSum sample corresponds to a BBC online article with a unique URL. The URL has information on whether the sub-directory is from the BBC Sport or BBC News page, and further sub-directory information, for example ('sport', 'formula1') or ('news', 'science').

The motivation for using this as guidance is that different sub-directories are likely to share properties relating to how the information is presented; journalists are also usually confined to one domain. We believe explicitly providing this information will enable CONTROL PREFIXES

[6]For a description of token-based detection vs. span-based detection, we refer the reader to https://www.cl.cam.ac.uk/research/nl/bea2019st/#eval.

to modulate both the processing of the document and generation of the summaries, whilst taking into consideration the news domain.

Examples of the source documents, the sub-directories used as guidance, gold references and generated summaries can be found in Appendix C.0.1.

5.1.4 Simplification

Sentence Simplification is an NLG task that aims to make a sentence more comprehensible by reducing its lexical and syntactic complexity while preserving its original meaning. Simplification can benefit non-native speakers (Paetzold and Specia [2016]), as well as people with cognitive disabilities such as dyslexia (Rello et al. [2013]).

The training sets predominantly used are based on automatic alignments of Simple English Wikipedia. WikiLarge (Zhang and Lapata [2017]) has been a popular training set, it houses 296k simplification pairs; and WikiAuto (Jiang et al. [2020]), which is larger (488,332 pairs), and featured as the Simplification training dataset in the GEM benchmark.[7] For evaluation, the two common benchmarks (both in GEM) are TurkCorpus (Xu et al. [2016]) and ASSET (Alva-Manchego et al. [2020]). Both benchmarks are composed of the same 2000 validation source and 359 test source sentences. TurkCorpus is associated with 8 human-written simplifications, whilst ASSET focuses on a more diverse set of rewriting simplifications and provides 10 references per source.

Text Simplification has been commonly evaluated with SARI (Xu et al. [2016]), which compares model-generated simplifications with the source sequence and gold references. SARI averages F1 scores for addition, keep, and deletion operations.[8] We report readability scores using the FKGL score (Kincaid et al. [1975]), a linear combination of sentence lengths and word lengths, and also report QuestEval (Section 2.4.2). Scialom et al. [2021] recently showed in a study with 9000 human ratings, that only FKGL and QuestEval are able to significantly measure Simplicity and Meaning Preservation (human evaluated dimensions in the study).

Examples of the complex sources, gold reference simplifications and generated summaries can be found in Appendix C.0.3, as well as Fig. 6.8.

Can Target specifications guide CONTROL PREFIXES?

ACCESS (Martin et al. [2019]) is a *controlled generation* technique, conditioning on simplification-specific control tokens for a given attribute, e.g. to indicate the amount of compression of the target sequence relative to the source sequence (Length Compression). We use

[7] We also experimented with a second version of WikiAuto, where the simple sentences are mapped to the same complex sentence as sentence splits. The rationale is that sentence splits are one of the varied rewriting operations present in the ASSET evaluation corpus (Alva-Manchego et al. [2020]). No performance improvement was observed.

[8] We use the latest version of SARI implemented in EASSE (Alva-Manchego et al. [2019]) which fixes bugs and inconsistencies from the traditional implementation of SARI.

the identical controllable attributes in this work: length compression, Levenshtein similarity, aggregated word frequency ratio, and dependency tree depth ratio [9]. We do use an adapted Levenshtein similarity control that only considers replace operations as in Martin et al. [2020].

The control ratios are discretized into bins of fixed width 0.05 in experiments and capped to a maximum ratio of 2. At inference time, once the model has been trained with these oracle controls, the control ratios are set to desired values. As in Martin et al. [2020], for ASSET and TurkCorpus the control ratios used on the respective test set are calculated by tuning on the respective validation set, with the zero-order optimization NEVERGRAD library (Rapin and Teytaud [2018]). Martin et al. [2020] fine-tuned a $\text{BART}_{\text{LARGE}}$ with control tokens using this procedure. Following this procedure enables us to directly compare *fixed-LM controlled generation* with CONTROL PREFIXES to *controlled generation* with full fine-tuning.

5.2 Implementation Details & Hyper-parameters

This section describes specific implementation details and hyper-parameters for our models. This thesis advocates reproducability—it is important for researchers, in NLG especially,[10] to state the exact hyper-parameters and implementation details used. We report our hyper-parameters in Appendix Table B.1.

We use $\text{BART}_{\text{LARGE}}$ for PREFIX-TUNING and CONTROL PREFIXES on all datasets except the Data-to-Text datasets. These models are always compared against fine-tuning $\text{BART}_{\text{LARGE}}$ baselines as well as the current top-performing systems. For PREFIX-TUNING with $\text{BART}_{\text{LARGE}}$, we use the original PREFIX-TUNING implementation (Li and Liang [2021])[11], with $d_m = 800$ and $k = 1024$.

PREFIX-TUNING for T5-large

$\text{BART}_{\text{LARGE}}$ exhibits inferior performance to T5 on Data-to-Text; for example, 9.7 BLEU points lower on WebNLG 2017 *Unseen* (Ribeiro et al. [2020]). In the WebNLG+ 2020 competition, Castro Ferreira et al. [2020] observed models based on fine-tuned $\text{BART}_{\text{LARGE}}$ compared to T5-large showed significantly lower performance across human evaluation measures. For this reason, we thought it prudent to implement T5 PREFIX-TUNING for the Data-to-Text datasets—this enables comparison against the current state-of-the-art.

For T5-large, we implement PREFIX-TUNING in a similar manner as for $\text{BART}_{\text{LARGE}}$, keeping $d_m = 800$, $k = 1024$. As T5 has relative position biases, we set these in all layers pertaining to offsets where the key is part of a prefix (general or control) to zero.

All implementations in this study are built on top of the Transformers library, developed by Hugging Face (Wolf et al. [2020]). This library houses the publicly available pre-trained language model checkpoints that our *fixed-LM* techniques utilize: $\text{BART}_{\text{LARGE}}$ and T5-large.

[9] We refer the reader to the original paper (Martin et al. [2019]) for details on ACCESS and how those values are computed.
[10] Due to decoding specific parameters during generation discussed in Section 2.3.
[11] According to `github.com/XiangLi1999/PrefixTuning` rather than the paper.

For T5-large we did not use V1.1, which includes improvements over the original model[12], as most systems we wanted to compare against use the original.

Data-to-Text in Practice

Recent neural approaches for triplesets-to-text generation linearize the triples as input for seq2seq models (Kale [2020]; Ribeiro et al. [2020]) as shown in Fig. 5.1.1 . Ribeiro et al. [2020] revealed that models such as T5-large fine-tuned are very successful, outperforming pipelines that explicitly encode the graph structure. Ribeiro et al. [2020] cite that PLMs are aided by the factual knowledge from pre-training; we follow the authors and prepend <H>, <R>, and <T> tokens before the subject, predicate, and object of an individual triple (Fig. 5.1.1). The embeddings relating to these special tokens are the only embeddings we train [13], as our work is focused on *fixed-LM* methods. We also prepend "translate Graph to English: " to every input (Ribeiro et al. [2020]) [14].

Hyper-parameters

We use gradient accumulation across batches to maintain an effective batch size (keeping this above 64 for all models). We employ a linear learning rate scheduler for all models. The hyper-parameters we consider are principally the learning rate, but also the number of epochs (which affects the scheduler), and the optimizer: AdamW (Loshchilov and Hutter [2017]) or AdaFactor (Shazeer and Stern [2018]).[15]

Decoding specific parameters were not tuned—we instead mirrored what the top-performing fine-tuned based system used for the particular LM and dataset (e.g. for T5-large on all Data-to-Text datasets a beam width of 5 as in Ribeiro et al. [2020]). We chose the checkpoint with highest validation set score with what we considered to be the principal metric (BLEU for Data-To-Text, $\mathrm{F}_{0.5}$ for GEC, SARI for Simplification, and ROUGE-2 for Summarization).

As Section 6 examines, we consider prefix length, ρ, and each control prefix length as architecture-specific parameters that we vary to try and maximize performance on the validation set. For all tasks, we train our models on single Tesla V100-SXM2-16GB machines, with mixed precision for $\text{BART}_{\text{LARGE}}$ models (fp16) and full precision for T5-large models (fp32).

[12] e.g. GEGLU activations in feed-forward hidden layers, rather than ReLUs.
[13] Except for 'PREFIX-TUNING + control tokens' models, but E of the LM remains fixed in all our models.
[14] The authors found improved performance with this instruction during fine-tuning. T5 was initially trained with such style of instructions (Raffel et al. [2020]).
[15] We experimented with further adaption of the learning rate schedulers depending on how often a class of an attribute is seen. This did not yield an observable improvement.

Section 6: Main Results

The objectives of this section are first to demonstrate that PREFIX-TUNING for the encoder-decoder is a strong baseline for a variety of NLG tasks, not just Summarization. The second objective is to demonstrate that CONTROL PREFIXES can further improve performance over PREFIX-TUNING according to official evaluation measures. This section incorporates descriptions of both the model set-ups as well as reporting on the results.[1]

Our research is not solely focused on parameter efficiency, but more on the effectiveness of adapting an already parameter efficient, *fixed-LM* method. Although for the reported models, CONTROL PREFIXES may have more additional parameters than vanilla PREFIX-TUNING, a fair comparison is ensured by attempting to maximize performance with each technique. We show in Section 6.5 for all the datasets, except XSum, the chosen prefix length, ρ, is in the realm of *performance saturation* and the only way to add parameters with PREFIX-TUNING is to increase ρ. We use the ρ that had maximum performance on the validation set in our parameter search, and demonstrate that using a higher ρ does not lead to an increase in performance. For XSum we ensure that CONTROL PREFIXES does not possess more parameters than PREFIX-TUNING.

Reporting Convention

- $\downarrow$ Signifies lower is better for that particular metric; if not specified, higher is better.
- $\phi\%$ to denote the % of additional parameters to the number of fixed LM parameters required at inference time for the downstream dataset.
- When reporting for a vanilla PREFIX-TUNING model, we report PREFIX-TUNING (ρ) to indicate the prefix length used.
- When reporting a CONTROL PREFIXES model, we report CONTROL PREFIXES (ρ;ρ_{c1},..ρ_{cn}) where ρ is the general prefix length and ρ_{ci} are the control prefix length(s)

6.1 Data-to-text: Source & Category

6.1.1 Data Augmentation

Can CONTROL PREFIXES be used as a data augmentation technique?

As DART is a conglomeration of 6 sub-datasets, we use CONTROL PREFIXES to inform the model from which sub-dataset an example belongs to. CONTROL PREFIXES (DART) (48;2) has a general prefix length of 48 ($\rho = 48$) and a control prefix for the attribute *sub-dataset source*, with $\rho_{c,1} = 2$. This control prefix switches alongside each training sample that belongs to a

[1]This layout is to aid the reader in keeping track of the various guidance signals and control prefixes for each dataset.

different sub-dataset. For DART, both CONTROL PREFIXES and PREFIX-TUNING attain higher performance (Table 6.1) than the current state-of-the-art—which is T5-large fined-tuned (Radev et al. [2020])—by 1.29 and 0.54 BLEU points respectively[2].

	$\phi\%$	BLEU	METEOR	TER ↓	BERTScore(F1)
Baselines & SOTA (Fine-tuning)					
T5-large* (Radev et al. [2020])	-	50.66	40	43	0.95
Our Proposed Methods (T5-large)					
PREFIX-TUNING (50)	1.0	51.20	40.62	43.13	0.95
CONTROL PREFIXES (DART) (48;2)	1.1	**51.95**	**41.07**	**42.75**	0.95

Table 6.1: Metrics on the DART test set (Radev et al. [2020]). We report results on the official evaluation script for v1.1.1, the same version as the official leaderboard, which is available here: https://github.com/Yale-LILY/dart. CONTROL PREFIXES outperforms vanilla PREFIX-TUNING. *Results for this model were only reported to the significant figures shown.

6.1.2 WebNLG

Explicit WebNLG RDF-triple Category Information as Guidance Signal

Alongside using CONTROL PREFIXES to leverage additional data, we investigate if providing the WebNLG category explicitly through CONTROL PREFIXES improves downstream performance.

For CONTROL PREFIXES (DART, WebNLG) models there are two control prefixes. The first is as before, to aid the model in leveraging additional data by annotating the DART source sub-dataset, and the latter to provide guidance on which WebNLG category, a WebNLG tripleset belongs. We also compare this model against CONTROL PREFIXES using *one* of the attributes. For example, CONTROL PREFIXES (WebNLG) is trained only on WebNLG data with only the *WebNLG category* control prefix; CONTROL PREFIXES (DART) is trained with DART as additional data with only the *sub-dataset-source* control prefix. Models that use DART are trained in two stages: i) with DART, ii) solely with WebNLG. The exact training details for each model can be found in B.1.

Note:

(i) For both WebNLG datasets, using additional training data is permitted.

(ii) As discussed in 5.1.1, the current SOTA for WebNLG 2017 is a T5-large fine-tuned model on WebNLG 2017 and a small subset of DART (dart_human_annotated) (Radev et al. [2020]). The authors experimented using all of DART, but this did not improve performance. E2E Clean was found to hurt performance the most—it is the largest component of DART and the most distinct in terms of consistency compared to the rest of DART.

[2]It is not clear if conditioning on the dataset source is permitted for leader-board purposes.

(iii) $\phi\%$ only incorporates the parameters required at inference time for the downstream dataset, i.e. for the DART *sub-dataset source* attribute, only the control prefix for the class of the downstream dataset is needed.

(iv) For CONTROL PREFIXES models with a *WebNLG category* attribute, a zero-shot transfer method is used for unseen categories which we outline in Section 7.2.

WebNLG 2017

	$\phi\%$	BLEU			METEOR			TER↓		
		S	U	**A**	S	U	**A**	S	U	**A**
Baselines & SOTA (Fine-tuning)										
(Kale [2020])										
T5-large*	-	63.9	52.8	57.1	46	41	44	-	-	-
T5-3B*	-	62.8	52.0	54.0	45	42	43	-	-	-
(Ribeiro et al. [2020])										
BART-large*	-	63.71	44.17	54.95	46	39	42	33	51	41
T5-large*	-	64.89	54.01	59.95	46	43	44	34	41	37
+Data: dart_human_annotated										
T5-large* † (Radev et al. [2020])	-	65.82	56.01	61.44	46	43	45	32	38	35
Our Proposed Methods (T5-large)										
PREFIX-TUNING (50)	1.0	66.95	55.39	61.73	46.73	42.71	44.87	31.34	39.01	34.86
+Data: dart_human_annotated										
PREFIX-TUNING (50)	1.0	67.05	55.37	61.78	46.69	42.82	44.90	31.36	38.79	34.77
CONTROL PREFIXES (WebNLG) (48;2)	1.4	**67.32**	55.38	61.94	46.78	42.77	44.92	30.96	39.01	34.65
+Data: DART										
CONTROL PREFIXES (DART) (48;2)	1.0	66.99	55.56	61.83	46.67	42.87	44.91	31.37	38.53	34.65
CONTROL PREFIXES (DART, WebNLG) (48;2,2)‡	1.4	67.15	**56.41**	**62.27**	46.64	43.18	45.03	31.08	38.78	34.61

Table 6.2: Metrics on the WebNLG 2017 test set. S, U and A refer to the *Seen*, *Unseen* and *All* portions of the dataset. CONTROL PREFIXES achieves state-of-the-art results. †dart_human_annotated, is a small subsection of DART. *These results were only reported to the significant figures shown.

Table 6.2 reveals that both CONTROL PREFIXES and PREFIX-TUNING attain higher performance than current methods. CONTROL PREFIXES (DART, WebNLG) outperforms the best vanilla PREFIX-TUNING model, and therefore attains state-of-the-art performance on WebNLG 2017. CONTROL PREFIXES achieves a 0.83 higher BLEU overall, and 1.33 in the *Seen* categories than the current SOTA (Radev et al. [2020]). Notably, CONTROL PREFIXES (WebNLG) outperforms CONTROL PREFIXES (DART, WebNLG) in the *Seen* component of the dataset, but does not generalize as well on the unseen categories. We argue this illustrates the benefit of using both control prefix attributes—as for both WebNLG datasets, CONTROL PREFIXES (DART, WebNLG) is the top-performing model.

PREFIX-TUNING trained on dart_human_annotated 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. Therefore the LM can generalize better. Whilst for PREFIX-TUNING, much of this gain has already been realized by retaining the original frozen ϕ parameters. Training PREFIX-TUNING on all of DART with the

two-stage approach outlined did not yield a performance increase. Table 6.2 includes results by Kale [2020] who fine-tune T5-3B, which achieves a lower performance than fine-tuning T5-large with the same setup [3]. Kale [2020] did not train with the demarcation tokens (<H>,<R>,<T>), nor the T5 text-prompt (explaining the performance mismatch with the result in Ribeiro et al. [2020] for the same model). However, the results are included as they are indicative that fine-tuning larger PLMs does not necessarily pay dividends for Data-to-Text.

WebNLG+ 2020

	ϕ%	BLEU	METEOR	chrF++	TER ↓	BLEURT
Baselines & SOTA (Fine-tuning)						
T5-large* † (Pasricha et al. [2020])	-	51.74	0.403	0.669	0.417	0.61
Amazon AI (Shanghai)* (Guo et al. [2020])	-	53.98	0.417	0.690	0.406	0.62
Our Proposed Methods (T5-large)						
PREFIX-TUNING (50)	1.0	54.74	0.417	0.693	0.399	0.62
CONTROL PREFIXES (WebNLG) (48;2)	1.6	54.97	0.417	0.693	0.398	0.62
+Data: DART						
CONTROL PREFIXES (DART) (48;2)	1.0	54.92	0.418	0.695	0.397	0.62
CONTROL PREFIXES (DART, WebNLG) (48;2,2)	1.6	**55.41**	**0.419**	**0.698**	**0.392**	**0.63**

(a) Overall automatic learned and non-learned metrics reported on the WebNLG+ 2020 test set.

	BLEU			METEOR			TER ↓		
	S	U	UE	S	U	UE	S	U	UE
Baselines & SOTA (Fine-tuning)									
T5-large* † (Pasricha et al. [2020])	58.26	45.57	52.76	0.388	0.416	0.415	0.408	0.438	0.381
Amazon AI (Shanghai)* (Guo et al. [2020])	60.35	49.15	52.25	0.434	0.404	0.413	0.404	0.413	0.394
Our Proposed Methods (T5-large)									
PREFIX-TUNING (50)	**60.63**	49.41	55.10	0.428	0.404	0.425	0.409	0.404	0.364
CONTROL PREFIXES (WebNLG) (48;2)	60.60	49.52	56.09	0.428	0.404	0.427	0.412	0.409	0.358
+Data: DART									
CONTROL PREFIXES (DART) (48;2)	59.95	49.93	56.10	0.427	0.407	0.426	0.408	0.406	0.362
CONTROL PREFIXES (DART, WebNLG) (48;2,2)	60.57	**50.39**	**56.41**	0.428	**0.409**	0.427	**0.402**	**0.400**	**0.360**

(b): Granular results: *Seen* (S), *Unseen Categories* (U) and *Unseen Entities* (UE).

Table 6.3: (a) The overall WebNLG+ 2020 test set results and **(b)** results breakdown across categories. *As the model outputs are publicly available, we are able to run evaluation to achieve the same precision. †Before fine-tuning on the WebNLG+ data, Pasricha et al. [2020] further pre-train T5 using a Mask Language Modelling objective (with 15% of the tokens masked) on two additional datasets: the WebNLG corpus and a corpus of DBpedia.

[3] It would be of interest to see if PREFIX-TUNING could successfully guide larger T5 models without performance degradation for this task, i.e. follow in the same vein as the SuperGLUE prompt-embedding tuning results of Lester et al. [2021].

Note: Many of the official challenge participants made use of the WebNLG 2017 data, which is also in DART. As discussed in Section 5, WebNLG+ 2020 is not part of DART, and is made up of distinct examples to WebNLG 2017.

Similarly for the GEM dataset *WebNLG+ 2020*, Table 6.3 displays that our top-performing model, CONTROL PREFIXES (DART, WebNLG), attains state-of-the-art performance, outperforming PREFIX-TUNING. Both models attain higher overall BLEU and METEOR than the current SOTA (Guo et al. [2020] [4]). This, along with another baseline (Pasricha et al. [2020]) reported use T5-large fine-tuned pipelines, that incorporate additional data. Vanilla PREFIX-TUNING trained with additional un-annotated DART data did not aid performance.

E2E Clean

E2E Clean (a GEM dataset) is a sub-dataset of DART. CONTROL PREFIXES (DART) attains higher *BLEU* than the state-of-the-art model. However, CONTROL PREFIXES reports 0.2 lower ROUGE-L than the incumbent SOTA (Harkous et al. [2020]); where the authors take a two-stage generation-reranking approach, combining a fine-tuned GPT-2 with a semantic fidelity classifier trained on additional generated data. Additionally, there is no difference between CONTROL PREFIXES and PREFIX-TUNING in NIST and CIDEr to the precision specified. Training vanilla PREFIX-TUNING with the additional DART data lowered performance. The model was trained in the same way as CONTROL PREFIXES (DART) with the two-stage approach: i) DART ii) solely E2E Clean.

	ϕ%	BLEU	NIST	METEOR	R-L	CIDEr
Baselines & SOTA (Fine-tuning)						
T5-Large (Gehrmann et al. [2021]) [5]	-	38.74	6.15	0.374	53.0	1.64
DATATUNER_FC (GPT-2)* (Harkous et al. [2020])	-	43.6	-	0.39	**57.5**	2.0
Our Proposed Methods (T5-large)						
PREFIX-TUNING (50)	1.0	43.66	6.51	0.390	57.2	2.04
+Data: DART						
PREFIX-TUNING (50)	1.0	43.04	6.46	0.387	56.8	1.99
CONTROL PREFIXES (DART) (48;2)	1.0	**44.15**	**6.51**	**0.392**	57.3	**2.04**

Table 6.4: Results on the E2E Clean test set: CONTROL PREFIXES outperforms vanilla PREFIX-TUNING, both outperform (except in terms of ROUGE-L) the current SOTA. *Results for this model were only reported to the significant figures shown.

We highlight the nature of the task through the diversity and system characterization results (Table A.3), which are low for both CONTROL PREFIXES and PREFIX-TUNING model outputs.

[4] The authors use a plan-and-pretrain approach—consisting of a relational graph convolutional network (R-GCN) planner and T5-large—this model makes use of the enriched version of WebNLG 2017.

[5] Note: the results presented in Gehrmann et al. [2021] are on the validation-set, but test set outputs for the reported models are available upon request.

The Distinct_1[6] is 0.003 for both, and the total vocabulary size used is 140 and 130 respectively (calculated over 1847 test outputs). Couple this with the method of converting the MR to a tripleset (Radev et al. [2020]), Table A.3 highlights how E2E Clean is an outlier of the DART sub-datasets in terms of consistency.

6.2 Grammatical Error Correction: CEFR level

6.2.1 Full BEA-Corpus, without CEFR

We trained a $\text{BART}_{\text{LARGE}}$ PREFIX-TUNING model, PREFIX-TUNING (100) in Table 6.5, on the BEA-Corpus data to gauge how well PREFIX-TUNING compared against Katsumata and Komachi [2020] who fine-tuned $\text{BART}_{\text{LARGE}}$ on all this data. PREFIX-TUNING $\text{BART}_{\text{LARGE}}$ outperforms fine-tuning (67.1 vs 65.6 $\text{F}_{0.5}$)[7] on BEA-test, indicating that PREFIX-TUNING could be promising in the field of GEC. We did not investigate further because of the training time associated with training of BEA-corpus ($>$ 0.5M samples with exclusions). The W&I Corpus training sample size (34k samples, which is 6% of the corpus used in PREFIX-TUNING (100)) is a limitation of the CEFR conditioning experiment, especially as top-performing systems use 100M+ synthetic examples.

6.2.2 W&I Corpus Only

CEFR level as Guidance for CONTROL PREFIXES

Here we investigate if CONTROL PREFIXES can leverage the CEFR level effectively, by constraining the models to only use the CEFR annotated W&I corpus. Table 6.5 displays BEA-dev[8] and the breakdown across BEA-test for all models. The BEA-test set incorporates samples from native speakers (CEFR: N), a proficiency level unseen during training. Here we can use common knowledge that a CEFR level of C aligns more closely with native speakers to execute zero-shot transfer.

Table 6.5 presents results for PREFIX-TUNING models trained on samples pertaining to a specific CEFR level. Each of these models performs best on the respective CEFR level it was trained on. This supports the ideas discussed in Bryant et al. [2019], and ideas motivating this experiment: samples belonging to specific CEFR levels make up distinct distributions with learnable properties. 'PREFIX-TUNING: Ensemble (15)' is the result of the ensemble of the three individually PREFIX-TUNING models where the 'PREFIX-TUNING: C' model is used for native

[6] The ratio of distinct unigrams over the total number of unigrams.

[7] Katsumata and Komachi [2020] only report for BEA-test $\text{F}_{0.5}$. The difference could be due to our study executing a detokenization preprocessing step the authors omitted. We followed the two-stage training procedure outlined in Lichtarge et al. [2019].

[8] BEA-dev is seldom reported on because only one reference exists compared to five for BEA-test—therefore, the resulting scores are not as reliable.

level samples at inference, and for CEFR A,B,C the respective individual models are used. The PREFIX-TUNING: Ensemble (15) attains lower performance because each of its sub-models only has access to limited data. In contrast to the CONTROL PREFIXES model, knowledge can not be shared across the sub-models. CONTROL PREFIXES (CEFR) (10;5) exhibits a minor increase in performance compared to PREFIX-TUNING (15) (65.21 vs. 64.47 $F_{0.5}$).

	ϕ%	Dev $F_{0.5}$	Test $F_{0.5}$				
			CEFR: A	CEFR: B	CEFR: C	CEFR: N	Overall
Our Proposed Methods (BART$_{LARGE}$)							
+Data: BEA-corpus							
PREFIX-TUNING (100)†	1.8	52.52	66.53	67.61	69.73	64.92	67.07
PREFIX-TUNING (15)	0.3	51.17	62.75	66.09	69.09	61.42	64.47
Individually Trained Models							
PREFIX-TUNING: A (15)	0.3	-	62.00	62.58	64.35	56.47	61.66
PREFIX-TUNING: B (15)	0.3	-	59.68	64.12	63.82	57.43	61.25
PREFIX-TUNING: C (15)	0.3	-	57.97	63.21	64.75	61.73	61.03
PREFIX-TUNING: Ensemble (15)‡	0.9	48.31	62.00	64.12	64.75	61.73	62.96
CONTROL PREFIXES (CEFR) (10;5)	0.5	51.69	63.53	66.70	68.75	63.98	65.21

Table 6.5: ERRANT $F_{0.5}$ results on the BEA-dev and BEA-test sets with the granular breakdown of $F_{0.5}$ per CEFR level. †This is the only model trained on additional data to W&I Corpus. ‡ Is the result of an ensemble of the three individually trained models, where the 'PREFIX-TUNING: C (15)' model is used for 'CEFR: N samples'.

Note: The BEA-test set is still a blind test set with an external leader-board hosted on Codalab, which can be found here. The CEFR Level is not permitted for the official task— however, a full model output breakdown, which includes CEFR level granular analysis is available upon submission. For the purposes of our research, we are able to infer the results of a system that conditions on the CEFR level. This is done by running inference on the external platform three separate times, conditioned on a different CEFR level each time.

6.3 Summarization: XSum, Article Source

Can the BBC News URL Sub-directories guide CONTROL PREFIXES to produce better summaries?

We use the URL as a source of conditioning information: having one coarse control prefix to condition on whether the sub-directory is from the BBC Sport or BBC News page, and then a second control prefix to condition on a further sub-directory: for example ('sport','formula1'), or ('news','science'). We therefore have two classes for the first control prefix, and 40 classes for the second. We use a $\rho_{c,1}$=2,and a $\rho_{c,2}$=1 for our CONTROL PREFIXES (113;2,1) model.

Table 6.6 demonstrates for our setup there is a slight performance increase for CONTROL PREFIXES over PREFIX-TUNING. We were not able to reproduce the results of Li and

	$\phi\%$	R-1	R-2	R-L
Baselines				
$\text{PEGASUS}_{\text{LARGE}}$ (Zhang et al. [2019])	-	47.60	24.83	39.64
$\text{BART}_{\text{LARGE}}$ (Lewis et al. [2020])	-	45.14	22.27	37.25
$\text{BART}_{\text{LARGE}}$				
$\text{BART}_{\text{LARGE}}$ (Li and Liang [2021])	2	43.80	20.93	36.05
Our Proposed Methods				
$\text{PEGASUS}_{\text{LARGE}}$				
PREFIX-TUNING (116)	2.0	46.01	23.62	38.71
$\text{BART}_{\text{LARGE}}$				
PREFIX-TUNING (165)	3.0	43.53	20.66	35.63
CONTROL PREFIXES (113;2,1)	2.8	43.81	20.84	35.81

Table 6.6: Results on the XSum test set. R-1, R-2 and R-L are ROUGE-1 , ROUGE-2 and ROUGE-L. PREFIX-TUNING under-performs fine-tuning of $\text{BART}_{\text{LARGE}}$ and $\text{PEGASUS}_{\text{LARGE}}$ [9]. We also include the original $\text{BART}_{\text{LARGE}}$ result, as reported in Li and Liang [2021].

Liang [2021][10]—however, it is uncertain, what ρ was used in their reported result. The result reported in the paper was cited as corresponding to a $\phi\%$=2. This does not corroborate with their accompanying graph, which shows this result corresponding to ρ=200, a $\phi\%$=3.6.

There are strong limitations on the conclusions we can draw for XSum. This dataset is an outlier, being the only dataset exhibiting a large performance gap with PREFIX-TUNING and fine-tuning, in favour of fine-tuning. XSum is the only dataset where we are not at *performance saturation* for vanilla PREFIX-TUNING. Additionally, XSum is resource-intensive for a modest resource of one Tesla V100-SXM2-16GB machine, taking many days to train a single model[11]. Consequently, we are not able to give a good indication of the maximum performance of PREFIX-TUNING. This is in contrast to the other datasets where we achieve near the maximum performance of a ρ parameter search, as assessed by automatic metrics, with very low ρ (eg. 99% with ρ=2 for the GEC and Data-to-Text datasets).

We submit our CONTROL PREFIXES (113;2,1) model outputs to the GENIE evaluation framework (Section 8.1); where despite low automatic metric scores, this model has higher human-assessed performance than all fine-tuned models. XSum was specifically chosen as one of the four datasets for GENIE because it involves abstractive summaries of a long document, and is therefore difficult to evaluate automatically. Additionally, XSum is the only dataset we consider with one gold reference.

[10]This is the only dataset trialled for the original PREFIX-TUNING encoder-decoder implementation.

[10]There are some known issues with XSum results reproduction, for example in fine-tuning $\text{BART}_{\text{LARGE}}$ and $\text{PEGASUS}_{\text{LARGE}}$, Shleifer and Rush [2020] note they were not able to reproduce results due to different tokenization schemes or other post-processing that impact ROUGE calculations. There is a lack of a standardized evaluation framework.

[11]Training $\text{PEGASUS}_{\text{LARGE}}$ with our setup takes time on the order of weeks.

6.4 Guidance for Simplification: Target Specifications

Can the *fixed-LM* CONTROL PREFIXES technique use oracle conditional information to control Text Simplification output?

Up until this point, we have only discussed guidance signal, G, that concerns a quality about the input. Here we consider guidance signal which aims to specifically influence the output. As described in Section 5.1.4, the guidance signal used is identical to previous work by Martin et al. [2020] [12]. Using the same guidance signal enables various axes of comparison.

Differences with 'BART$_{\text{LARGE}}$ with ACCESS' model (Martin et al. [2020])

'BART$_{\text{LARGE}}$ with ACCESS' is a fine-tuned BART$_{\text{LARGE}}$ model with control tokens, which adjusts the embedding matrix, E as well as all the parameters of BART$_{\text{LARGE}}$. In contrast,CONTROL PREFIXES (96;1,1,1,1) is a *fixed-LM* method and extends *controlled generation* by having a control prefix for each of the controllable attributes. We additionally report on PREFIX-TUNING + control tokens, which is more similar to BART$_{\text{LARGE}}$ with ACCESS; however, the embedding matrix, E is still fixed, along with all of BART$_{\text{LARGE}}$ parameters. With this model, control has to be exerted through the few control embeddings and PREFIX-TUNING's ability to steer frozen ϕ parameters; therefore, even this simpler method still demonstrates a parameter efficient, *fixed-LM* method can exert effective *controlled generation*.

		ASSET			TurkCorpus		
	ϕ%	SARI	FKGL ↓	QuestEval ‡	SARI	FKGL ↓	QuestEval ‡
Baselines & SOTA (Fine-tuning)							
Gold Reference	-	$44.87_{\pm 0.36}$	$6.49_{\pm 0.15}$	$0.63^{*}_{\pm 0.01}$	$40.04_{\pm 0.30}$	$8.77_{\pm 0.08}$	$0.66^{*}_{\pm 0.01}$
BART$_{\text{LARGE}}$	-	39.91^{*}	7.73^{*}	-	39.55^{*}	7.73^{*}	-
BART$_{\text{LARGE}}$ with ACCESS (Martin et al. [2020])	-	$43.63_{\pm 0.71}$	$6.25_{\pm 0.42}$	$0.64^{*}_{\pm 0.01}$	$42.62_{\pm 0.27}$	$6.98_{\pm 0.95}$	$0.66^{*}_{\pm 0.01}$
BART$_{\text{LARGE}}$ (ACCESS+Mined) (Martin et al. [2020])	-	$44.15_{\pm 0.56}$	$6.05_{\pm 0.51}$	-	$42.53_{\pm 0.36}$	$7.60_{\pm 1.06}$	-
Our Proposed Methods (BART$_{\text{LARGE}}$)							
Data: WikiAuto							
PREFIX-TUNING (100) †	1.8	39.84	7.65	-	39.11	7.65	-
Data: WikiLarge							
PREFIX-TUNING (100)	1.8	40.12	7.28	-	39.06	7.28	-
Control Tokens (100)	1.8	$43.64_{\pm 0.17}$	$5.81_{\pm 0.32}$	$0.63_{\pm 0.01}$	$42.36_{\pm 0.30}$	$7.86_{\pm 0.22}$	$0.66_{\pm 0.01}$
CONTROL PREFIXES (96;1,1,1,1)	1.8	$43.58_{\pm 0.36}$	$5.97_{\pm 0.39}$	$0.64_{\pm 0.02}$	$42.32_{\pm 0.25}$	$7.74_{\pm 0.27}$	$0.66_{\pm 0.01}$

Table 6.7: Text Simplification results on ASSET and TurkCorpus test sets, PREFIX-TUNING performs comparably to fine-tuning. The table is adapted from Martin et al. [2020]. All the **baselines & SOTA** results are transcribed from there, except those labelled with *—those results were calculated in this study [13]. Where appropriate we report scores on the test sets averaged over 5 random seeds with 95% confidence intervals to aid in comparison with Martin et al. [2020]. †This model is trained on WikiAuto (all other BART$_{\text{LARGE}}$ models are trained on the WikiLarge (Zhang and Lapata [2017]) dataset, except BART$_{\text{LARGE}}$ (ACCESS+Mined) which is trained on additional mined data). ‡The QuestEval hash we use is contained in this link (due to its length).

[12] Experimenting with different controls, such as sentiment, did not yield any improvement, and maintaining the same guidance enables direct comparison with fine-tuning in Martin et al. [2020].
[13] The model outputs of Martin et al. [2020] are publicly available

Simplification Analysis

In Table 6.7[14] the *Gold Reference* result is computed via a leave-one-out scenario where each reference is evaluated against all others, and then an average is taken. Note the *Gold Reference* produces inferior results for SARI and FKGL compared to several other models. These outputs were themselves computed with those *Gold References*—undermining the credibility of the use of SARI and FKGL for TurkCorpus to assess Simplification.

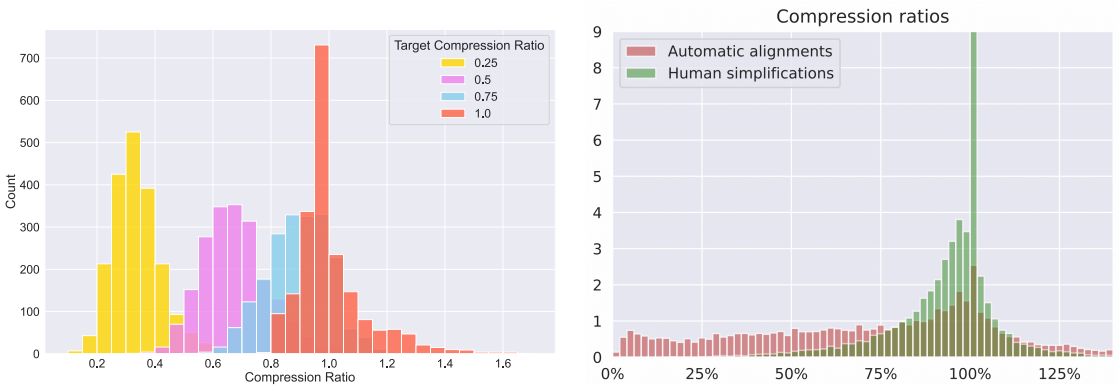


Figure 6.1: Left: This work, illustrating the influence of different target length ratios on the actual length compression ratio output distribution for CONTROL PREFIXES (96;1,1,1,1) on the TurkCorpus validation set. Right: Source: Martin et al. [2019], the density distribution of the **length compression ratios** between the source sentence and the target sentence. The automatically aligned pairs from the WikiLarge train set are spread (red) while human simplifications from the TurkCorpus validation and test set (green) are more clustered with a mean ratio of 0.93 (i.e. nearly no compression).

Performance of our PREFIX-TUNING + control tokens and CONTROL PREFIXES is similar, with PREFIX-TUNING + control tokens having 0.06 and 0.04 higher mean SARI over 5 runs for ASSET and TurkCorpus respectively. However, the ASSET QuestEval result is 0.01 higher for CONTROL PREFIXES. We argue the performance is comparable and this is because the complexity of leveraging the conditioning information is low—most of the gains are from controlling the length ratio, which can be adequately achieved with soft embeddings, even for a *fixed-LM* method.

Fig. 6.4: Right, gives indication of how important tuning for the length compression ratio on the validation set is, partly explaining why models with this guidance achieve more than 2 SARI points on ASSET and TurkCorpus than unguided models. The spread of the training set (red) is near uniform, while the test and validation sets share a more narrow distribution. Fig. 6.4[15]: Left, 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 shows CONTROL PREFIXES is capable of

[14]The difference between using the variants of WikiAuto (Jiang et al. [2020], a part of GEM) and WikiLarge datasets for PREFIX-TUNING were minimal. We opted to use WikiLarge for the other models, which has the added benefit of making a fair comparison with Martin et al. [2020].

[15]In the interest of not depicting derivative results, we only produce this plot. Similar histograms can be found in Martin et al. [2019] for their *fine-tuned LM* method, where they go into detail about having to fix the length compression ratio to more clearly see the other attributes be controlled.

controlling the target length with respect to the input. Table 6.8 displays example output generations with each of the (0.25,0.5,0.75,1.0) values specified.

We also run QuestEval[16] (Scialom et al. [2021],Section 2.4.2), a novel machine-learned reference-less metric [17]. Interestingly the results for QuestEval are higher across models for TurkCorpus compared to ASSET. We argue this is because the test set gold references are on average 114 characters for TurkCorpus, as opposed to 98 for ASSET. Therefore, the ASSET references contain less information to answer the generated queries during QuestEval evaluation; and thus, there is lower performance. We argue this shows a limitation with using QuestEval as a reference-less metric for simplification— by favouring longer generations for Simplification. This is reminiscent of Sun et al. [2019], who showed ROUGE can be improved by increasing the output length of the model.

	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 (96;1,1,1,1) Length Control: 0.25	West Coast blues is a type of blues music from the United States.
CONTROL PREFIXES (96;1,1,1,1) 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 (96;1,1,1,1) 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 (96;1,1,1,1) 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 6.8: *Controlled* Text Simplification: a qualitative example of our CONTROL PREFIXES model from the ASSET Corpus validation Set. Only the control prefix for the length control ratio is varied, depicted in red. The corresponding CONTROL PREFIXES output is displayed. †Note, this is one random Gold Reference out of 10.

Fixed-LM $\text{BART}_{\text{LARGE}}$ vs. fine-tuned $\text{BART}_{\text{LARGE}}$

Table 6.7 reveals that PREFIX-TUNING $\text{BART}_{\text{LARGE}}$ performs comparably to when we fine-tune $\text{BART}_{\text{LARGE}}$. When comparing our *fixed-LM* $\text{BART}_{\text{LARGE}}$ methods to fine-tuned '$\text{BART}_{\text{LARGE}}$ with ACCESS' there is comparable performance in terms of SARI for ASSET, and better FKGL

[16]With machine-learned metrics it is very important to report the hash associated —due to its length we report it as a link in Table 6.4.

[17]Although QuestEval can take references, the authors maintain that any improvement in correlation with human performance is very minor. Additionally, it was of interest to compare TurkCorpus and ASSET outputs independent of their references.

results. However on TurkCorpus both our guided models yield lower performance on average for SARI and FKGL. We highlight the *Gold Reference* scores for TurkCorpus, which indicate that >42 SARI is supposedly greater than human level performance. For ASSET the *Gold Reference* SARI scores are significantly higher (95% confidence) than all guided models, but according to FKGL the performance is significantly worse (95% confidence).

We do not suggest Simplification is a solved task by any means. If we compare Simplification to Data-to-Text, Simplification suffers from having many valid outputs for any particular input, and the validity is more subjective than verbalizing a tripleset graph correctly. Metrics like QuestEval, which take into account synonyms, but are not based on token alignment, offer a promising avenue for future work.

6.5 Complexity

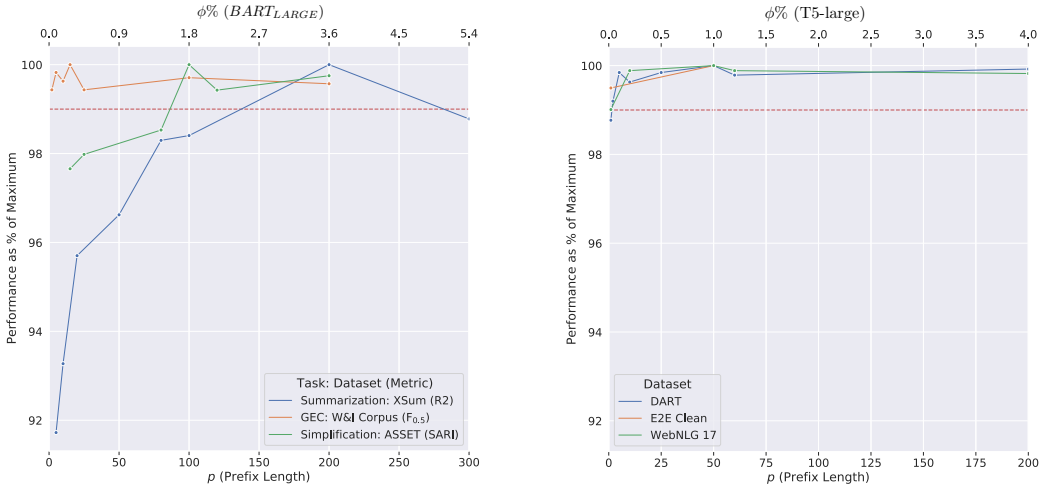
We outline the notion of dataset complexity and present the results of CONTROL PREFIXES against the less expressive PREFIX-TUNING + control tokens model. This less expressive model is more parameter efficient than CONTROL PREFIXES and still aligns with our principal objective of investigating if parameter efficient, *fixed-LM* methods can leverage attribute-level information as guidance signal.

6.5.1 Dataset Complexity

In Section 3.2.3 we introduced the notion of *dataset complexity*, where it is possible to measure the capacity required to capture a specific task by varying prefix/prompt length. Fig. 6.2 illustrates that different datasets require varying prefix lengths to attain near maximum performance in a parameter search for ρ. We can use the length of the prefix as a proxy for *dataset complexity*, akin to experiments conducted by Aghajanyan et al. [2020] and Lester et al. [2021]. We set a proxy indicator as the minimum ρ needed to reach 99% of the maximum validation-set performance attained in a parameter sweep for ρ.

The proxy measure does give an intuitive indication of complexity, i.e. the number of parameters required to steer a pre-trained LM effectively. It suggests that the Data-to-Text datasets and the W&I Corpus hold less complexity than XSum or WikiLarge. This measure also does not necessarily correlate with dataset size, as WikiLarge is larger than XSum (300k vs. 200k samples). The length of inputs for the XSum dataset are whole documents, where a large burden is placed on the encoder, rather than sentences.

There are clear caveats with such a proxy measure: viz., we only have a very small sample of trained models; and the measure is relative to the maximum performance of the sample. For example, the fact that there is a significant gap with summarization performance (according to ROUGE-2) compared to fine-tuning is not encapsulated. It also assumes that a % difference in one metric is comparable to another. The proxy measure is model dependent, hence we plot on



(a) $BART_{LARGE}$ PREFIX-TUNING **(b)** T5-large PREFIX-TUNING

Figure 6.2: Dataset Complexity: PREFIX-TUNING results of a model parameter search on several datasets for the optimal ρ per dataset. These results are on the metric monitored per task on the respective validation sets (except XSum) indicated in the legend. For XSum these data points are reported from Li and Liang [2021] for the test set. The y-axis is a relative measure: the validation set performance as a % of the maximum attained in the parameter search. The red-dotted line indicates 99% of this maximum performance: a proxy indicator for the *dataset complexity*. **(a)** $BART_{LARGE}$ based models. **(b)** T5-large based models.

separate graphs, as the number of prefix parameters depends on L and d of the model. This is especially important when considering the prompting at scale result by Lester et al. [2021], which suggest with larger fixed LMs, smaller prompts more effectively steer the frozen parameters ϕ.

For all datasets, besides XSum[18], *performance saturation* is observed. This is in contrast to Hu et al. [2021], who report *performance degradation* for PREFIX-TUNING. The authors report more than a 7% drop in validation accuracy after a certain p for GPT-3 on the WikiSQL dataset. However the implementation has differences[19] and is PREFIX-TUNING for a decoder-only architecture. Therefore, many of the benefits discussed in Sections 3.2.3 and 4 do not apply.

6.5.2 PREFIX-TUNING + Control Tokens

We also describe results found in Appendix E for the architecture 'PREFIX-TUNING + control tokens'. We use the same discrete guidance signal for 'PREFIX-TUNING + control tokens' as our top-performing CONTROL PREFIXES model on DART, E2E Clean, W&I and both WebNLG datasets. For WebNLG, an identical zero-shot approach, as described in Section 7.2 is executed

[18]The experiments with XSum are limited, as mentioned in Section 6.3, the training time is considerable for $\rho > 100$. Therefore we are limited to reporting Li and Liang [2021] results for XSum.
[19]Distinct trainable keys and values are not trained, but a single activation in each layer.

for the *unseen* categories. Across the Data-to-Text datasets, PREFIX-TUNING + control tokens attains lower performance than CONTROL PREFIXES.

These supplementary experiments bolster the claim that for many tasks, 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. As discussed in Section 4.1.3, CONTROL PREFIXES possesses more degrees of freedom, as well as coordinating the modulation with the general task prefix, P_θ, of the fixed-LM in each attention mechanism subspace. This is not the case for the Simplification test sets, despite being where the guidance signal we use, ostensibly (through % change of desired automatic metric), improves performance the most.

We argue there are two factors at play, i) the strength of the underlying guidance signal and ii) how difficult it is to leverage this guidance signal. For Text Simplification, the gains are due to the validation and test sets being drawn from the same distribution, as opposed to the WikiLarge training distribution. Therefore, during optimization on the validation set, the model is able to calibrate qualities for the respective test set references; namely, the length compression ratio. However, we argue controlling such attributes as length ratio does not require complex conditioning, even if the LM is fixed; therefore, the performance of CONTROL PREFIXES and PREFIX-TUNING + control tokens are similar.

Selecting Control Prefix Length

We fixed the ρ from inspecting the results shown in Fig 6.2, and trained multiple CONTROL PREFIXES models, increasing the control prefix length for a given attribute, $\rho_{c,a}$, where $a \in \mathcal{A}$ until the desired metric on the validation-set stopped increasing. There are limitations in this study; with modest resources one cannot do a wide parameter search across multiple dimensions. We leave for future work further exploration of the interplay of ρ and control prefix lengths. A $\rho_{c,a} = 1$ governs many degrees of freedom ($6dL$ parameters), when considering that with vanilla PREFIX-TUNING $\rho = 1$ ($6dL$ parameters) is enough to reach the 99% level in Fig. 6.2 for W&I Corpus and WebNLG 2017. It may be that if the conditional attribute information is more complicated to integrate, a larger $\rho_{c,a}$ is successfully able to utilize this added complexity than a smaller $\rho_{c,a}$.

6.6 Section Conclusion

Through learned and non-learned automatic metrics, we have demonstrated how CONTROL PREFIXES performs on a variety of NLG tasks when leveraging an input-level guidance signal. This section shows CONTROL PREFIXES can be used to leverage disparate datasets to the dataset of interest; we argue it can do this through capturing inductive biases not relevant for the downstream dataset in those respective control prefixes, enabling better final performance. CONTROL PREFIXES outperforms PREFIX-TUNING on all datasets. We believe the performance improvement over vanilla PREFIX-TUNING is contingent on the underlying task, and the guidance signal G.

Section 7: Control Prefix Interpretability

The objectives of this section are to first demonstrate different control prefixes corresponding to similar attribute classes (i.e. in Simplification, varying length ratios for the length attribute) share properties. The second objective is to build on this insight and illustrate that zero-shot learning is able to be successfully deployed to samples possessing attribute classes unseen during training. We acknowledge that there has to be some prior on the properties of an unknown attribute class.

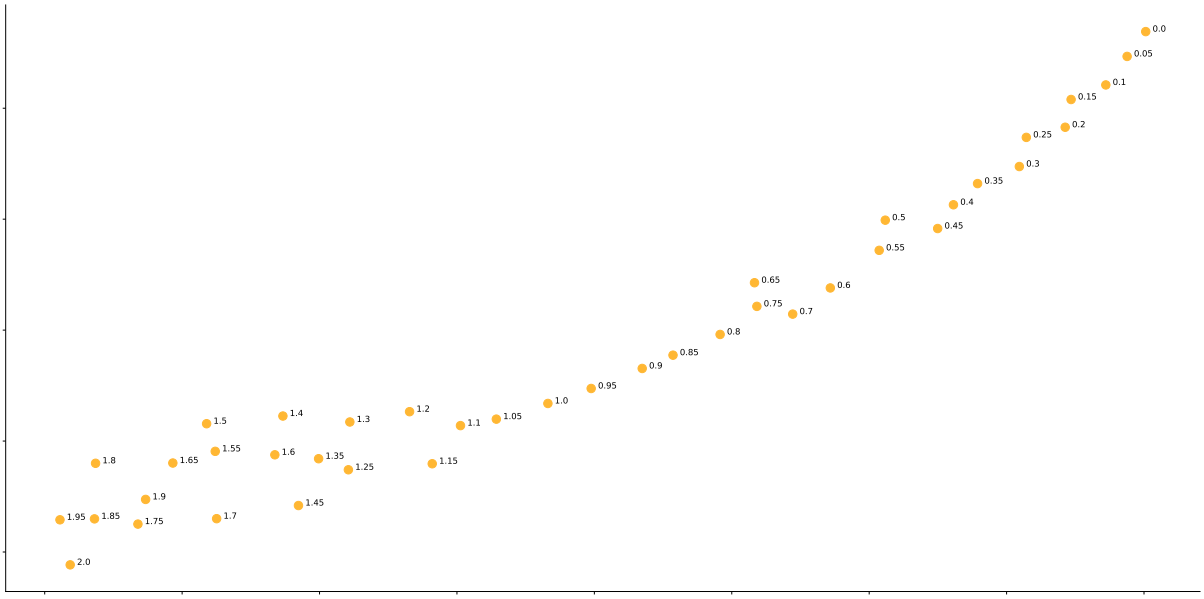
7.1 Target *controlled generation*

Fig. 7.1 (over the page) displays t-SNE (Maaten and Hinton [2008]) visualizations[1] of the Length Compression control prefixes learnt as part of the Simplification model CONTROL PREFIXES (96;1,1,1,1). These visualizations pertain to the discussion in Section 4.1.1, where we argue there is a coordination of the control prefixes and general prefix for each attention mechanism, enforced by the shared MLPs: η^{E},η^{Dc},η^{Dm}. One control prefix represents three components, the decoder masked attention (Fig. 7.1 **a**), encoder attention (Fig. 7.1 **b**), and decoder cross-attention (Appendix D.2) mechanisms.

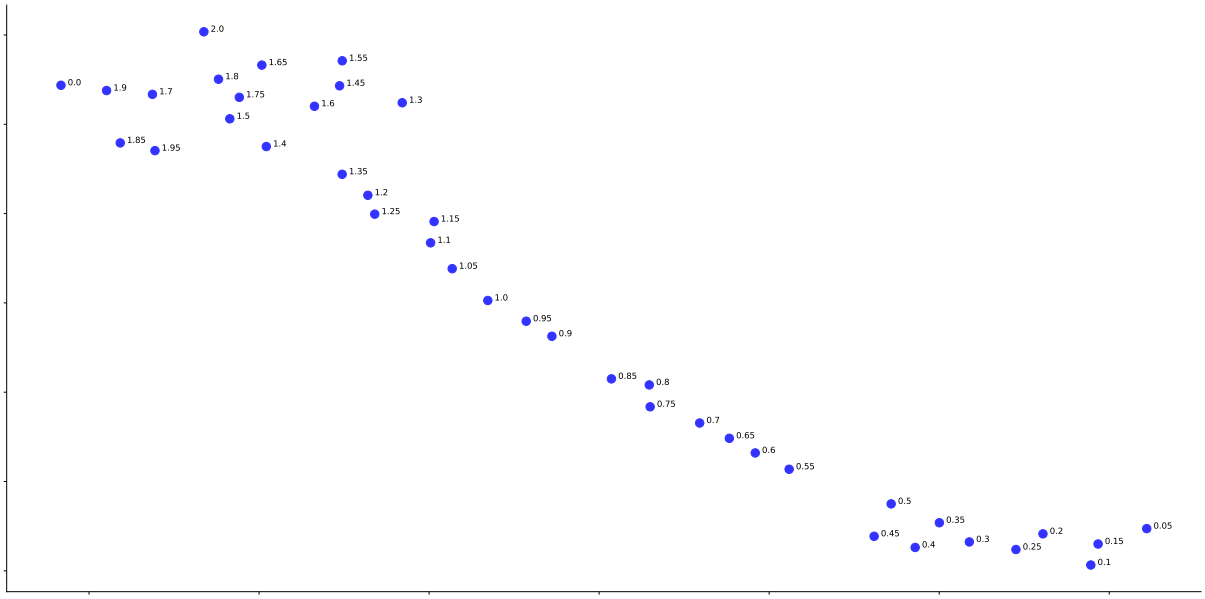
There are known limitations in forming conclusions using t-SNE representations (Linderman and Steinerberger [2017]); however, if we take Fig. 7.1 **(a)**—the relationship here is very manifest. An ordered line of 41 representations would be difficult to contrive if it were not indeed present. As each control prefix constituent is high dimensional and non-linear, t-SNE is a suitable technique to investigate if there is an underlying relationship (Linderman and Steinerberger [2017]). Each control prefix is formed from a shared non-linear MLP, and comprises learnt keys-value pairs at each layer.

Fig. 6.4 depicts the WikiLarge training set distribution of length compression ratios. Note the near uniform distribution of ratios from 0 to 1.0, which explains how meaningful control prefixes can be learnt for each of these classes. The spread in Fig. 6.4 becomes very sparse after 1.2, which explains why the representations are not as interpretable for values greater than 1.2 in Fig. 7.1 **(a)**. There are far fewer training samples where the simplified output is much longer than the complex, original input. As expected, the control is exerted in the masked-attention of the decoder, for the length compression ratio directly concerns the target.

[1] A perplexity of 5 is used for all plots.



(a) $C^{Dm}_{\theta,l}$, the Decoder, **masked** attention constituents of the length control prefixes.



(b) $C^{E}_{\theta,l}$, the **Encoder** attention constituents of the length control prefixes.

Figure 7.1: t-SNE visualizations for constituents of the length Compression (l) Control Prefixes, i.e. $C_{\theta,l}$, where $l \in \mathcal{A}$. 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: **(a)** D_m **(b)** E. (A graphic for D_c can be found in Appendix D.2).

Interestingly, the decoder cross-attention constituents (which we place in the Appendix D.2) are not as interpretable. For encoder-decoder LMs, the cross-attention key-value pair is identical in each layer (not the case for the prefix key-value pairs in PREFIX-TUNING and CONTROL PREFIXES). The lack of an evident relationship for such a simple attribute may support the idea of research into parameter efficiency for PREFIX-TUNING and CONTROL PREFIXES. This may be indicative that the cross-attention prefix constituent requires fewer parameters.

The control prefixes for the other attributes of CONTROL PREFIXES (96;1,1,1,1) are visualized in D.1—where there are some conspicuous relationships. We note Martin et al. [2019] found it was necessary to fix the length ratio to 1.0 for control of other attributes to manifest themselves.

7.2 Zero-shot Learning

(a) WebNLG 2017 **(b)** WebNLG+ 2020

Figure 7.2: t-SNE visualizations for the WebNLG category ($c \in \mathcal{A}$) control prefixes. Each diagram depicts only the encoder constituent, $C^E_{\theta,c}$, of the respective CONTROL PREFIXES (WebNLG) (48;2) model. Each circle represents a category seen during training. WebNLG 2017 has 10 seen categories and 5 unseen categories. All 15 categories are seen categories in WebNLG+ 2020, along with the category *Company*. WebNLG+ 2020 has three additional unseen categories to those shown.

How to approach *Unseen* WebNLG Categories?

Fig. 7.2 depicts t-SNE visualizations of the *encoder* constituent of the control prefixes relating to WebNLG Categories seen during training. When training CONTROL PREFIXES with WebNLG category attributes, a choice emerges: what does one do for unseen samples with an accompanying textual label?

Fig 7.1 indicates that similar classes of an attribute share properties. Previous work has discussed the notion of task similarity for *prompt learning* methods (Achille et al. [2019]); however, we argue prefixes concerning different classes of one attribute are more likely to overlap in terms of learnable properties than different tasks or whole datasets. In our experiments the control prefix lengths are generally much smaller than the general prefix length, so we posit this is a better testbed for these ideas.

We map each category's textual label, including for the unseen categories, to a Glove embedding[2]. Then for each unseen category, we map to the seen category with the highest cosine similarity in embedding space, and use that control prefix at inference for the corresponding unseen sample. Table 7.2 displays example model output for WebNLG+ 2020, and WebNLG 2017, along with the zero-shot procedure [3].

Table 7.1 shows a comparison of training an out-of-vocabulary (OOV) style control prefix on a random 2% of the data for each accumulated batch [4], and the zero-shot transfer method. These results indicate that zero-shot transfer is more promising than a learned OOV representation. Admittedly, the result fundamentally depends on the WebNLG categories, and if similar textual labels pertain to similar triplesets that CONTROL PREFIXES can utilize. A limitation of this study is only having results for two models where in each case the unseen categories are those set by the organizers. In future work we hope to explore this in more detail, for example on the WebNLG training datasets by training multiple models and performing leave-one-category-out-cross-validation—i.e. each test fold made up solely of a category not present in the train fold.

	Unseen		
	BLEU	METEOR	TER ↓
WebNLG 2017			
OOV Representation	56.35	43.15	38.82
Zero-shot	56.41	43.18	38.78
WebNLG+ 2020			
OOV Representation	50.02	40.8	40.3
Zero-shot	50.39	40.9	40.0

Table 7.1: A comparison of the performance on *Unseen* (Every sample from the test set with an unseen category) for both WebNLG datasets, with i) a single OOV Control Prefix used for all samples from unseen categories, or ii) the zero-shot transfer approach outlined, utilizing the textual labels available. Each CONTROL PREFIXES (DART, WebNLG) (48;2,2) model was trained with an OOV Control Prefix, and then both zero-shot and the OOV control prefixes run at inference to facilitate direct comparison.

Note: For our models in Section 6 that have a *WebNLG category* attribute control prefix, we use the zero-shot procedure outlined in this section.

[2]Glove Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors).
[3]Further examples of model generations including zero-shot transfer generations for PREFIX-TUNING, CONTROL PREFIXES, and the accompanying gold references, can be found in C.0.2.
[4]See Appendix B Hyper-parameters, for details on effective batch size.

WebNLG 2017	
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 (50)	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 (48;2,2)	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 (50)	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 (48;2,2)	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 7.2: WebNLG 2017 Example Generations: Sources are shown in their linearized form, as fed to the T5-large based models, with PREFIX-TUNING and one of the Gold References shown for comparison with CONTROL PREFIXES. Triplesets are from WebNLG 2017 unseen categories and the zero-shot procedure using the textual category labels is depicted. 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.

Section 8: Human Evaluation

A key objective of this thesis is to systematically evaluate the performance of CONTROL PREFIXES across NLG tasks. Here we present results of human evaluation of CONTROL PREFIXES from both external and internal human annotators. This provides a complementary angle of evaluation to the learned and non-learned automatic metrics in Section 6 and Appendix A.

8.1 External Human Evaluation: GENIE Platform

We report human-assessment of CONTROL PREFIXES's abstractive summaries for XSum, via the external evaluation platform *GENIE* (Khashabi et al. [2021])[1] run by independent researchers. The annotation template used is similar to that of Chaganty et al. [2018]—where different aspects of quality (redundancy, fluency, conciseness, etc.) are assessed which have traditionally been of interest in the field (McKeown and Radev [1995]). Table 8.1 is transcribed from the GENIE public leaderboard, where the human evaluated scores by crowd workers are depicted, along with 95% confidence intervals which are computed with bootstrap re-sampling. The current submissions include the fully fine-tuned models $\text{BART}_{\text{LARGE}}$ and $\text{PEGASUS}_{\text{LARGE}}$, which as we illustrated in Table 6.6, outperform PREFIX-TUNING of $\text{BART}_{\text{LARGE}}$ in terms of ROUGE. Evaluation takes place on 300 instances costing $90 per submission.

CONTROL PREFIXES (113;2,1) is the lowest performing system, as assessed by automatic learned and non-learned metrics. Nevertheless, it is the top-performing system in terms of human-assessed qualities. The confidence intervals indicate that this result is not necessarily definitive—but it at least highlights the problems with evaluation for XSum, and that the quality of CONTROL PREFIXES' generations in this domain are not captured fully with ROUGE. A sample size of 300 is typically much larger than that where authors construct their own evaluation (e.g. Narayan et al. [2018] use 50, Dou et al. [2020] use 100). Moreover, recent work by Fabbri et al. [2020b] demonstrate that for Summarization the learned automated metric BERTScore (Zhang et al. [2020b]) does not improve upon the correlation with human judgements of ROUGE.

Support for External Human Evaluation

For most NLG tasks, there is little consensus on how human evaluation should be conducted. Papers often leave out important details on how they were conducted, such as who the evaluators were and how many people evaluated the text. Clear reporting of human evaluation is extremely important; currently there is considerable inconsistency, which inhibits comparison of results across papers (van der Lee et al. [2021]).

[1] For full details of the approach and justification we refer the reader to Khashabi et al. [2021]. Important factors are taken into account: the quality of workers, the efficacy of Likert scales and mean aggregation; as well as empirical evaluation of whether results remain the same across time, or whether results remain the same across annotator populations.

XSUM (Summarization)								
Systems	**Human overall**	**Human conciseness**	**Human fluency**	**Human no-hallucination**	**Human informativeness**	BERTScore	ROUGE	BLEURT
$\text{BART}_{\text{LARGE}}$	$0.49^{+0.03}_{-0.04}$	$0.50^{+0.03}_{-0.03}$	$0.50^{+0.03}_{-0.03}$	$0.52^{+0.03}_{-0.03}$	$0.49^{+0.03}_{-0.03}$	0.92	0.37	−0.19
Pegasus	$0.49^{+0.03}_{-0.03}$	$0.52^{+0.02}_{-0.03}$	$0.49^{+0.03}_{-0.02}$	$0.49^{+0.03}_{-0.03}$	$0.49^{+0.03}_{-0.03}$	0.92	**0.39**	−0.17
T5 (11B)	$0.47^{+0.03}_{-0.03}$	$0.49^{+0.02}_{-0.02}$	$0.50^{+0.03}_{-0.03}$	$0.49^{+0.03}_{-0.03}$	$0.48^{+0.03}_{-0.03}$	0.92	0.38	**−0.14**
Our Proposed Methods								
$\text{BART}_{\text{LARGE}}$ CONTROL PREFIXES (113;2,1)	$\mathbf{0.51^{+0.03}_{-0.03}}$	$\mathbf{0.53^{+0.02}_{-0.02}}$	$\mathbf{0.51^{+0.03}_{-0.03}}$	$\mathbf{0.53^{+0.03}_{-0.03}}$	$0.49^{+0.03}_{-0.03}$	0.91	0.36	−0.21

Table 8.1: GENIE results: results of several fine-tuned models alongside our $\text{BART}_{\text{LARGE}}$ -based CONTROL PREFIXES (fixed-LM method) model on the XSum dataset. This is the public leaderboard, reproduced verbatim and available at https://leaderboard.allenai.org/genie-xsum/submissions/public. We also present qualitative model outputs (Table C.4) of T5-large fine-tuned and our CONTROL PREFIXES (113;2,1) model.[2]

Even for automatic evaluation, external benchmarks with hidden test sets assist with result reporting consistency (Celikyilmaz et al. [2020]). We fully advocate more benchmarks like GENIE, which have the express aim of having unbiased attestations of performance. Human evaluation is only the gold standard provided considerations are taken into account, such as annotator training and inter-annotator agreement.

8.2 Internal Human Evaluation: WebNLG+ 2020

This section further validates the quality of models, specifically comparing vanilla PREFIX-TUNING to CONTROL PREFIXES. We conduct a human evaluation for WebNLG+ 2020, for the models PREFIX-TUNING (50) and CONTROL PREFIXES (WebNLG) (48;2)[3].

This dataset was chosen due to T5-large based CONTROL PREFIXES excelling in Data-to-Text. In addition, conscious of the difficulties with orchestrating human evaluation, we noted the authors of the task (Castro Ferreira et al. [2020]) ran a systematic human evaluation study comparing the output of 15 systems and the gold references. Thus, there was a well-reasoned framework to echo—we follow much of this approach, with the same intrinsic evaluation criteria. We similarly stratify selected samples according to tripleset size, as well as the *Seen*, *Unseen Entities*, *Unseen* dataset splits, in order to provide additional axes of performance analysis.

The notable distinctions in our study is use of a Ratings Scale of 1-10, rather than a scale of 0-100; we evaluate 70 tripleset inputs, opposed to 178; and we use three internal annotators, as opposed to recruited crowd-sourced workers. The principal gold reference, PREFIX-TUNING (50), and CONTROL PREFIXES (WebNLG) (48;2) are annotated for the same 70 IDs, with a 19/19/32 split for *Seen*, *Unseen Entities*, *Unseen*. We use a (1-10) Ratings scale, considering

[2]It would have been useful to have a GENIE submission for PREFIX-TUNING, so as to compare human-assessed performance of PREFIX-TUNING vs. CONTROL PREFIXES. However, participants are encouraged to keep to one submission due to the goal of keeping the expenses of running the leaderboard low.

[3]We use CONTROL PREFIXES (WebNLG) (48;2) to maintain consistent training data: only WebNLG+ 2020.

our sample size and we had concerns of differing annotator interpretations of the scale if it was too nuanced (Briakou et al. [2021]). Using internal annotators does offer some benefits; in that with crowd-sourcing, numerous difficulties arise in ensuring the quality in annotators' judgements (Howcroft et al. [2020]).

There are clear caveats: our sample size is small and each example is only annotated once. To mitigate the differences between scoring strategies of our internal human raters, we normalized scores of each participant by computing their z-scores.

The following human-assessed criteria are[4]:

1. **Data Coverage:** *Does the text include descriptions of all predicates presented in the data?*
2. **Relevance:** *Does the text describe only such predicates (with related subjects and objects), which are found in the data?*
3. **Correctness:** *When describing predicates which are found in the data, does the text adequately mention the objects and introduce the subject for this specific predicate?*
4. **Text Structure:** *Is the text grammatical, well-structured and written in acceptable English?*
5. **Fluency:** *Is it possible to say that the text progresses naturally, forms a coherent whole and is easy to understand?*

Human Evaluation Analysis

Table 8.2 reveals the results of our internal human evaluation study. CONTROL PREFIXES is assessed as marginally better than PREFIX-TUNING in each criteria for the Unseen Categories. The only statistically significant difference $(\mathrm{p} < 0.05)$ for an assessed criteria, according to a two-tailed t-test, was the *Relevance* between vanilla PREFIX-TUNING and the Gold References. Qualitative examples of model output for these systems can be found in Table C.4, where for the examples shown both PREFIX-TUNING and CONTROL PREFIXES were assessed to have produced higher quality textual representations of the input than the gold reference.

Using the Wilcoxon Rank-Sum Test we observe no statistically significant difference ($p > 0.05$) between the Gold Reference, PREFIX-TUNING and CONTROL PREFIXES. This is not unsurprising, as the top-performing models in Castro Ferreira et al. [2020] that were based on fine-tuned T5, such as AmazonAI (Guo et al. [2020], used as baseline in Table 6.3), were found to share the same ranking cluster with the gold references.

Fig. 8.2 presents sentence-level Pearson correlations of the human-assessed dimensions and automatic metrics for the 140 human-assessed samples corresponding to model output. BLEURT correlates most highly with our human-assessed ratings compared to character-overlapping metrics, such as BLEU, chrF++ and TER. This mirrors the findings in Castro Ferreira et al. [2020]—who caution concluding that the task is "solved", but recognize the dataset's limitations: a relatively restricted vocabulary, and a template-based structure where properties are lexicalised in a similar manner across texts.

[4]We refer the reader to the instructions in the Appendix of Castro Ferreira et al. [2020] for further details on the annotation instructions used.

	Data Coverage		Relevance		Correctness		Text Structure		Fluency	
	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw
Overall										
Gold Reference	0.17	9.81	0.19	9.81	0.19	9.80	0.08	9.61	0.01	9.57
PREFIX-TUNING † (50)	−0.14	9.60	−0.17	9.54	−0.10	9.57	−0.02	9.54	0.05	9.60
CONTROL PREFIXES (DART, WebNLG) (48;2,2)	−0.03	9.67	−0.03	9.64	−0.09	9.59	−0.05	9.53	−0.06	9.54

(a): Overall Human Evaluation Results.

	Data Coverage		Relevance		Correctness		Text Structure		Fluency	
	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw	Avg. Z	Avg. Raw
Seen										
Gold Reference	0.24	9.84	0.08	9.74	0.17	9.79	−0.05	9.53	−0.09	9.53
PREFIX-TUNING † (50)	0.07	9.74	0.02	9.68	0.12	9.74	0.16	9.68	0.34	9.79
CONTROL PREFIXES (WebNLG) (48;2)	0.39	9.95	0.39	9.95	0.33	9.89	0.35	9.84	0.21	9.74
Unseen Entities										
Gold Reference	0.31	9.89	0.38	9.95	0.39	9.95	0.07	9.63	0.05	9.58
PREFIX-TUNING † (50)	0.22	9.84	0.02	9.68	0.15	9.79	0.16	9.68	0.21	9.68
CONTROL PREFIXES (WebNLG) (48;2)	0.05	9.79	0.05	9.74	−0.18	9.58	−0.18	9.47	−0.23	9.47
Unseen										
Gold Reference	0.05	9.75	0.15	9.78	0.09	9.72	0.16	9.66	0.04	9.59
PREFIX-TUNING † (50)	−0.48	9.38	−0.38	9.38	−0.38	9.34	−0.25	9.38	−0.21	9.44
CONTROL PREFIXES (WebNLG) (48;2)	−0.33	9.44	−0.32	9.41	−0.29	9.41	−0.21	9.38	−0.12	9.47

(b): Granular Human Evaluation Results: *Seen*, *Unseen Entities*, *Unseen Categories*.

Table 8.2: Human evaluation results **(a)** Overall, and **(b)** per the original dataset splits. Average z-scores (Avg. Z) alongside the average raw scores (Avg. Raw) are shown for each of the five criteria.

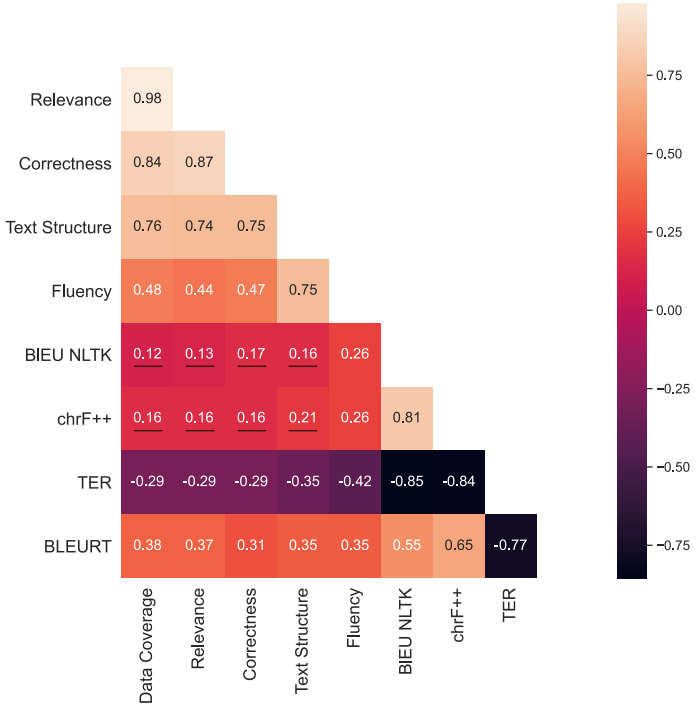


Figure 8.1: The sentence-level Pearson correlations of the evaluation metrics for the 140 *generated* samples in our human evaluation study for WebNLG+ 2020, between the five human-assessed criteria and automatic metrics. The results underlined are not statistically significant ($p > 0.05$).

Section 9: Conclusion & Future Work

9.1 Contributions

Demonstrate PREFIX-TUNING for the encoder-decoder is a promising *fixed-LM* technique for various NLG tasks other than Summarization

After supplying a more explicit description of PREFIX-TUNING for the transformer encoder-decoder architecture than Li and Liang [2021], we reveal PREFIX-TUNING alone can act as a strong baseline. PREFIX-TUNING can outperform incumbent Data-to-Text SOTA methods when implemented for T5-large. This thesis argues that the distinct re-parameterizations involved in producing the *prefix* allow for the modulation of the input and target to be delegated effectively which results in more effective control of the base language model (LM). This observation is partly what enables CONTROL PREFIXES to work effectively for controllable attributes concerning either the input or output sentence.

We observe increasing prefix length results in *performance saturation* rather than *performance degradation*. This is in contrast with Hu et al. [2021], who use a form of PREFIX-TUNING for the decoder-only models GPT-3, and GPT-2. We measure the capacity required to capture a specific task by varying prefix length, akin to experiments conducted by Aghajanyan et al. [2020]. We believe this gives a proxy indicator for *dataset complexity*.

Present CONTROL PREFIXES as an effective *fixed-LM Controlled Generation* Technique

We introduce CONTROL PREFIXES, which can successfully leverage various types of discrete guidance signal, whilst only adding 1-2% additional parameters to a fixed LM. The signal can concern the *input*, such as the WebNLG tripleset category, the news article domain for Summarization, or the CEFR level for GEC. The signal can also concern *target* attributes such as length output/input ratio for Text Simplification. CONTROL PREFIXES can even be used as a data augmentation technique—we demonstrate that CONTROL PREFIXES offers a means of more effectively leveraging multiple heterogeneous Data-to-Text datasets for a downstream dataset. CONTROL PREFIXES transforms PREFIX-TUNING into a *dynamic* prompting method and as a result is state-of-the-art according to several GEM datasets and WebNLG 2017.

Compare CONTROL PREFIXES with a less expressive architecture

We compare CONTROL PREFIXES with PREFIX-TUNING + control tokens, another *fixed-LM* method introduced in this work, with <2% additional parameters to the fixed LM. We show that CONTROL PREFIXES outperforms PREFIX-TUNING + control tokens on all datasets except the Simplification datasets.

Demonstrate how attribute-level similarity can be exploited with CONTROL PREFIXES

This thesis illustrates how zero-shot learning can be successful with CONTROL PREFIXES. We establish that the control prefixes corresponding to similar classes of a single attribute (i.e. different length ratios for the length attribute) share properties. We believe that even for more complicated attributes if the attribute classes are similar, the respective control prefixes will similarly guide both the general, task-specific prefix parameters and the frozen LM parameters. This idea is in a similar vein to the task similarity measurement discussed in Achille et al. [2019].

With this insight, we show zero-shot learning is able to be successfully deployed to samples with unseen attribute information. In the case of CEFR level for GEC, this is done by applying a CEFR C prefix to native samples rather than A or B. In the case of WebNLG, where although no examples of the unseen category are present during training, a natural language representation of the category exists. This gives us some prior on the properties of the unseen category, which we show is enough to successfully zero-shot transfer with control prefixes.

9.2 Conclusion

In this thesis, we have proposed CONTROL PREFIXES, a novel method which builds on PREFIX-TUNING (a lightweight *fixed-LM* method). CONTROL PREFIXES is able to utilise discrete guidance signal at the input-level. We discover that CONTROL PREFIXES offers a powerful way to integrate conditional input-level information and outperforms vanilla PREFIX-TUNING on an array of natural language generation tasks. CONTROL PREFIXES attains state-of-the art results on the WebNLG+ 2020, DART and WebNLG 2017 Data-to-Text datasets[1]. This is despite learning <2% additional parameters to the fixed LM parameters. CONTROL PREFIXES has comparable performance with the state-of-the-art on the Simplification test dataset ASSET and currently holds the highest human evaluation ranking on the external platform *GENIE*, for the Abstractive Summarization dataset XSum.

9.3 Future Work

Parameter Efficiency and Reducing Inference Latency

PREFIX-TUNING and CONTROL PREFIXES do add additional inference latency compared to parameter efficient *low-rank* methods like Hu et al. [2021]. This is dependent on the prefix length, the control prefix lengths and the beam width used during decoding. As discussed in Section 2.2.1, the prefixes add to the space and time complexity of multi-head attention computations. Future work is also necessary to systematically evaluate the run-time and maximum memory GPU usage during inference for each technique compared to the fixed LM baseline.

[1] All results are calculated with the official evaluation scripts provided by the dataset curators.

In this work, we did not consider parameter efficiency to be a chief aim of the thesis, as long as the additional parameters were less than 3%. We showed that the cross-attention CONTROL PREFIXES constituent was not as interpretable for length control specification. This may indicate fewer parameters are needed for this constituent. A future area of research would be looking at what parameters can be excised and what is the minimal number of parameters to achieve strong performance.

Prompt + LM Tuning

In this work, we discussed *fixed-LM* methods. CONTROL PREFIXES doesn't have to be constrained to *fixed-LM* methods, which was a criterion for this thesis. As discussed in Section 3.3, GSum (Dou et al. [2020]) employ an effective prompt + fine-tuning strategy, where both the prompt and pre-trained LM's parameters are tuned. CONTROL PREFIXES may be an effective way to integrate conditional information, where the rest of the LM's parameters could adapt to the modulation of the control prefixes. As we discussed in 4.1.3, CONTROL PREFIXES is a more expressive way to integrate a guidance signal than control tokens, control tokens having been traditionally trained alongside the underlying LM.

Hu et al. [2021] is the second paper to use PREFIX-TUNING, and they cite PREFIX-TUNING can be used in conjunction with their technique *LoRA*, which injects low-rank weight updates. CONTROL PREFIXES may offer additional benefits to *LoRA*, when applied in conjunction.

CONTROL PREFIXES Similarity Metric

Measuring the similarity of the CONTROL PREFIXES attribute classes, similar to the ideas in Achille et al. [2019], may offer an efficient means of working out which types of guidance signal complement one another. Lester et al. [2021] proposed to use prompt-embedding tuning and design a task similarity metric to provide an efficient way to search existing datasets and identify which tasks could benefit each other. We argue control prefixes concerning different classes of one attribute are more likely to overlap in terms of learnable properties than different tasks or whole datasets.

Increasing Power at Scale

Results by Lester et al. [2021] indicate that the *prompt learning* method prompt-embedding tuning is more competitive with the size of the base LM model. A future research direction would be to see if this held with CONTROL PREFIXES, and that if larger base language models are more capable of being steered with control prefixes in light of a guidance signal. This thesis was limited with resources, therefore only base LMs of $<$1B parameters were considered.

9.4 Ethical, Legal and Environmental Considerations

Ethics Checklist We have included the *Ethics Checklist* as provided by the Imperial College Department of Computing. Only Section 2 and Section 4 are relevant for this study—we do use human participants, collecting data for our internal evaluation. No personal data is collected, only annotation data with written consent and the data is not publicly distributed. Therefore this is in compliance with GDPR[2]. The use of the publicly available XSum dataset contains articles published on individuals including celebrities and politicians. As we are conducting scientific research and we assess that this "processing is necessary for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes", we are also in compliance with the GDPR.

Bias in machine learning Looking further than the *Ethics Checklist*, we acknowledge that the biases machine learning models learn during training pose a substantial problem in both the machine learning and natural language processing community[3]. 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 many kinds of biases. We also acknowledge that it is also possible to have biases in the data records from DBpedia in WebNLG (Castro Ferreira et al. [2020]) and in the model outputs that are presented to our human annotators.

Environmental footprint The leading models in NLP have become increasingly resource-intensive since the arrival of the Transformer in 2017. Although our research is conducted under the purview of parameter efficient NLP methods, we still require at least one large pre-trained language model. In addition, researching these techniques is still resource-intensive. We have trained models using up to 6 V100-SXM2-16GB GPUs[4], amassing 1000s of hours of compute. There is a responsibility for the considerable CO2 emissions in the NLP community and for developing more resource-efficient training and inference methods.

Openness and reproducibility These are central tenets of the scientific ethos. We abide by these principles by releasing our code.

[2] https://gdpr-info.eu/art-4-gdpr/.
[3] Chang et al. [2020] and Floridi and Chiriatti [2020] give a good account of the increasing concerns surrounding this issue.
[4] Due to GPU provider restrictions it was not possible to distribute training.

Appendices

Appendix A: Supporting Results

A.0.1 GEM Metrics

Dataset	Model	Metrics (Lexical Similarity and Semantic Equivalence)						
		METEOR	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	BLEURT
Structure-to-text (T5-large)								
DART	PREFIX-TUNING (50)	0.405	76.7	53.0	61.7	50.2	0.95	0.32
	Control Tokens (DART) (50)	0.408	77.2	53.5	62.3	50.8	0.95	0.33
	CONTROL PREFIXES (DART) (48;2)	0.410	77.3	53.7	62.4	51.1	0.96	0.33
E2E Clean	PREFIX-TUNING (50)	0.385	74.5	48.3	55.8	43.7	0.95	0.23
	Control Tokens (DART) (50	0.385	74.1	47.8	55.8	43.6	0.95	0.22
	CONTROL PREFIXES (DART) (48;2)	0.387	74.4	48.4	55.9	44.1	0.95	0.23
WebNLG 2017	PREFIX-TUNING (50)	0.443	81.1	59.6	67.8	60.3	0.96	0.43
	PREFIX-TUNING (50) (dart_human_annotated)	0.443	81.2	59.8	67.8	60.4	0.96	0.43
	CONTROL PREFIXES (WebNLG) (48;2)	0.443	81.3	59.9	67.9	60.5	0.96	0.43
	CONTROL PREFIXES (DART) (48;2)	0.443	81.3	59.8	68.1	60.5	0.96	0.43
	Control Tokens (DART, WebNLG) (50)	0.443	81.2	59.7	68.0	60.5	0.96	0.43
	CONTROL PREFIXES (DART, WebNLG) (48;2,2)	0.444	81.4	60.0	68.0	60.8	0.96	0.43
WebNLG+ 2020	PREFIX-TUNING (50)	0.417	79.6	56.2	64.8	56.2	0.96	0.32
	CONTROL PREFIXES (WebNLG) (48;2)	0.417	79.5	56.3	65.1	56.3	0.96	0.32
	CONTROL PREFIXES (DART) (48;2)	0.418	79.6	56.5	65.3	56.4	0.96	0.33
	Control Tokens (DART, WebNLG) (50)	0.417	79.5	56.2	64.8	56.2	0.96	0.32
	CONTROL PREFIXES (DART, WebNLG) (48;2,2)	0.419	80.0	56.9	65.4	56.8	0.96	0.34
Summarization ($BART_{LARGE}$)								
XSum	PREFIX-TUNING (165)	0.197	43.4	20.5	35.5	14.1	0.91	-0.22
	CONTROL PREFIXES (113;2,1)	0.199	43.8	20.9	35.8	14.4	0.91	-0.21

Table A.1: The set of additional lexical similarity and semantic equivalence results on the official test sets. These metrics are proposed by Gehrmann et al. [2021] [1]

Dataset	Model	Metrics (Lexical Similarity and Semantic Equivalence)						
		METEOR	ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	BLEURT
Simplification ($BART_{LARGE}$)								
ASSET	Control Tokens (100)	$0.447_{\pm 0.016}$	$79.1_{\pm 2.2}$	$62.6_{\pm 3.1}$	$73.0_{\pm 4.0}$	$69.6_{\pm 4.4}$	$0.96_{\pm 0.00}$	$0.18_{\pm 0.05}$
	CONTROL PREFIXES (96;1,1,1,1)	$0.464_{\pm 0.008}$	$81.7_{\pm 2.0}$	$66.1_{\pm 2.9}$	$76.9_{\pm 2.9}$	$73.8_{\pm 4.1}$	$0.96_{\pm 0.00}$	$0.22_{\pm 0.03}$
TurkCorpus	Control Tokens (100)	$0.508_{\pm 0.013}$	$85.9_{\pm 1.0}$	$75.6_{\pm 1.3}$	$84.6_{\pm 1.2}$	$80.5_{\pm 1.1}$	$0.98_{\pm 0.00}$	$0.38_{\pm 0.02}$
	CONTROL PREFIXES (96;1,1,1,1)	$0.514_{\pm 0.012}$	$86.2_{\pm 1.2}$	$75.1_{\pm 1.6}$	$85.0_{\pm 1.1}$	$80.7_{\pm 2.2}$	$0.98_{\pm 0.00}$	$0.38_{\pm 0.02}$

Table A.2: The set of additional lexical similarity and semantic equivalence results, as proposed by Gehrmann et al. [2021], for ASSET and TurkCorpus on the test sets averaged over 5 random seeds with 95% confidence intervals.

[1] The evaluation scripts for calculating Tables (A.1,A.2,A.3,A.4) can be found here: https://github.com/GEM-benchmark/GEM-metrics. 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	$\lvert\mathcal{V}\rvert$	Output Len.
Structure-to-text (T5-large)										
DART	PREFIX-TUNING (50)	0.45	0.04	0.13	8.1	10.97	1.5k	5.2k	4.8k	21.2
	Control Tokens (DART) (50)	0.45	0.04	0.13	8.12	11.01	1.5k	5.4k	4.8k	21.4
	CONTROL PREFIXES (DART) (48;2)	0.45	0.04	0.13	8.11	10.98	1.5k	5.3k	4.8k	21.5
E2E Clean	PREFIX-TUNING (50)	0.32	0.003	0.01	5.70	7.28	6	57	130	24.8
	Control Tokens (DART) (50)	0.33	0.003	0.01	5.72	7.30	20	94	153	25.3
	CONTROL PREFIXES (DART) (48;2)	0.32	0.003	0.01	5.71	7.29	8	73	140	25.3
WebNLG 2017	PREFIX-TUNING (50)	0.52	0.09	0.26	8.57	11.88	973	4.6k	3.4k	21.1
	PREFIX-TUNING (50) (dart_human_annotated)	0.52	0.09	0.26	8.57	11.87	968	4.6k	3.4k	21.1
	CONTROL PREFIXES (WebNLG) (48;2)	0.52	0.09	0.26	8.57	11.89	997	4.7k	3.4k	21.2
	CONTROL PREFIXES (DART) (48;2)	0.52	0.09	0.26	8.57	11.88	965	4.6k	3.4k	21.1
	Control Tokens (DART, WebNLG) (50)	0.52	0.09	0.26	8.56	11.87	968	4.6k	3.4k	21.2
	CONTROL PREFIXES (DART, WebNLG) (48;2,2)	0.52	0.08	0.25	8.52	11.81	962	4.4k	3.4k	21.3
WebNLG+ 2020	PREFIX-TUNING (50)	0.66	0.04	0.13	8.05	10.94	327	1.8k	1.6k	23.0
	CONTROL PREFIXES (WebNLG) (48;2)	0.66	0.04	0.13	8.05	10.92	326	1.8k	1.6k	23.0
	CONTROL PREFIXES (DART) (48;2)	0.66	0.04	0.13	8.04	10.92	326	1.8k	1.6k	23.1
	Control Tokens (DART, WebNLG) (50)	0.66	0.04	0.13	8.06	10.89	322	1.8k	1.6k	23.2
	CONTROL PREFIXES (DART, WebNLG) (48;2,2)	0.66	0.04	0.13	8.05	10.9	300	1.7k	1.5k	23.0
Summarization ($\text{BART}_{\text{LARGE}}$)										
XSum	PREFIX-TUNING (165)	0.76	0.08	0.38	9.77	14.57	8.4k	59.3k	17.0k	20.0
	CONTROL PREFIXES (113;2,1)	0.76	0.07	0.38	9.76	14.56	8.5k	59.5k	17.2k	20.2

Table A.3: The set of additional diversity and system characterization results. These metrics are proposed by Gehrmann et al. [2021]. 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 ($\text{Distinct}_{1,2}$), and the count of n-grams that only appear once across the entire test output ($\text{Unique}_{1,2}$, Li et al. [2016]), as well as the vocabulary size over the output ($\lvert\mathcal{V}\rvert$) and the mean output length of a system (Sun et al. [2019]).

Dataset	Model	Metrics (Diversity and System Characterization)								
		MSTTR	Distinct_1	Distinct_2	H_1	H_2	Unique_1	Unique_2	$\lvert\mathcal{V}\rvert$	Output Len.
Simplification ($\text{BART}_{\text{LARGE}}$)										
ASSET	Control Tokens (100)	$0.70_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.80_{\pm 0.01}$	$8.96_{\pm 0.05}$	$12.10_{\pm 0.07}$	$1.8\text{k}_{\pm 0.0}$	$5.1\text{k}_{\pm 0.2}$	$2.5\text{k}_{\pm 0.1}$	$20.5_{\pm 1.1}$
	CONTROL PREFIXES (96;1,1,1,1)	$0.71_{\pm 0.00}$	$0.35_{\pm 0.01}$	$0.81_{\pm 0.01}$	$8.99_{\pm 0.04}$	$12.11_{\pm 0.03}$	$1.8\text{k}_{\pm 0.1}$	$5.1\text{k}_{\pm 0.1}$	$2.5\text{k}_{\pm 0.1}$	$20.2_{\pm 0.4}$
TurkCorpus	Control Tokens (100)	$0.72_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.82_{\pm 0.04}$	$9.03_{\pm 0.03}$	$12.20_{\pm 0.05}$	$1.8\text{k}_{\pm 0.0}$	$5.4\text{k}_{\pm 0.2}$	$2.6\text{k}_{\pm 0.0}$	$21.0_{\pm 0.6}$
	CONTROL PREFIXES (96;1,1,1,1)	$0.72_{\pm 0.01}$	$0.34_{\pm 0.01}$	$0.82_{\pm 0.02}$	$9.04_{\pm 0.06}$	$12.22_{\pm 0.03}$	$1.9\text{k}_{\pm 0.1}$	$5.5\text{k}_{\pm 0.1}$	$2.6\text{k}_{\pm 0.1}$	$21.4_{\pm 0.9}$

Table A.4: The set of additional diversity and system characterization results, as proposed by Gehrmann et al. [2021], for ASSET, and TurkCorpus, on the test sets averaged over 5 random seeds with 95% confidence intervals.

Appendix B: Hyper-parameters

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 50	-	7e-5	Ada	2000	30	6	96	5	1	0	384	No
Control Tokens (DART) (50)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
CONTROL PREFIXES (DART) (48;2)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
E2E Clean (T5-large)												
PREFIX-TUNING 50	-	8e-5	Ada	2000	50	6	96	5	1	0	384	No
Control Tokens (DART) (50)	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
CONTROL PREFIXES (DART) (48;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 2017 (T5-large)												
PREFIX-TUNING (50)	-	7e-5	Ada	2000	30	6	96	5	1	0	384	No
PREFIX-TUNING (50) (dart_human_annotated)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
CONTROL PREFIXES (WebNLG) (48;2)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
CONTROL PREFIXES (DART) (48;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 Tokens (DART, WebNLG) (50)	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 (DART, WebNLG) (48;2,2)	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
WebNLG+ 2020 (T5-large)												
PREFIX-TUNING (50)	-	7e-5	Ada	2000	30	6	96	5	1	0	384	No
CONTROL PREFIXES (48;2)	-	7e-5	Ada	2000	40	6	96	5	1	0	384	No
CONTROL PREFIXES (DART) (48;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 Tokens (DART, WebNLG) (50)	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 (DART, WebNLG) (48;2,2)	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
W&I Corpus (BART$_{LARGE}$)												
PREFIX-TUNING (Lang-8, FCE, W&I) (100)	1	1e-5	AdamW	2000	50	8	64	6	1	0	100	No
	2	5e-5	AdamW	2000	50	8	64	6	1	0	100	No
PREFIX-TUNING (15)	-	5e-5	AdamW	2000	50	8	64	6	1	0	100	No
PREFIX-TUNING Ensemble (15)	-	5e-5	AdamW	2000	50	8	64	6	1	0	100	No
Control Tokens (CEFR) (50)	-	5e-5	Ada	2000	50	8	64	6	1	0	100	No
CONTROL PREFIXES (CEFR) (10,5)	-	5e-5	Ada	2000	50	8	64	6	1	0	100	No
XSum												
PEGASUS$_{LARGE}$: PREFIX-TUNING (116)†	-	1e-4	Ada	2000	30	4	140	8	0.8	10	60	✓
BART$_{LARGE}$: PREFIX-TUNING (165)	-	5e-5	AdamW	0	30	8	40	6	1	10	60	✓
BART$_{LARGE}$: CONTROL PREFIXES (113;2,1)	-	7e-5	Ada	2000	40	8	128	6	1	10	60	✓
ASSET & TurkCorpus (BART$_{LARGE}$)												
PREFIX-TUNING (100)	-	5e-5	AdamW	2000	30	8	64	6	0.8	3	100	✓
PREFIX-TUNING (100)	-	5e-5	AdamW	2000	30	8	64	6	0.8	3	100	✓
Control Tokens (100)	-	4e-5	Ada	5000	30	8	64	6	1	3	100	✓
CONTROL PREFIXES (96;1,1,1,1)	-	4e-5	Ada	5000	30	8	64	6	1	3	100	✓

Table B.1: Hyper-parameters of the salient experiments in the report. 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. †Additionally, label smoothing of 0.1 is performed as was the case for the original fine-tuning (Zhang et al. [2019]) of XSum.

Appendix C: Qualitative Examples

C.0.1 Summarization

	XSum
news world	Kamal C Chavara was detained by the police in Kerala state on Sunday after the youth wing of the Hindu nationalist BJP lodged a complaint against him. Last month, the Supreme Court ruled that the anthem must be played in every cinema before a film is screened. Some 20 people have been held in Kerala and Tamil Nadu since then for remaining seated during the anthem. Also, India's colonial-era sedition law has been often used against students, journalists, writers and social activists and those critical of the government. Reports said that the BJP's youth wing lodged a complaint against a Facebook post by Mr Chavara which allegedly insulted the anthem. The post was apparently an excerpt from one of his books. Senior police official Sateesh Bino told the NDTV news channel that the writer-activist "is being questioned for his controversial post on the national anthem on Facebook" and had been charged with sedition. Earlier this month, 12 people were arrested at a cinema in Kerala, after they remained seated while the national anthem played. The cinemagoers, who were attending an international film festival, were later freed but they face charges of "failure to obey an order issued by a public servant, thereby causing obstruction or annoyance to others". And at a cinema in Chennai, eight people who did not stand for the anthem were assaulted and abused, police said. The eight were later charged with showing disrespect to the anthem.
Gold	A writer in India has been charged with sedition for allegedly showing disrespect to the national anthem.
T5-large (fine-tuned) (**70.97/48.28/70.97**)	A prominent Indian writer has been charged with sedition for defying the National Anthem.
CONTROL PREFIXES (**59.46/34.29/54.05**)	An Indian writer-activist has been charged with sedition over a post on Facebook which allegedly insulted the national anthem.
sport horse-racing	The 33-1 shot, ridden by David Mullins and trained by Mouse Morris, triumphed at Aintree in April to become the first novice to win the race since 1958. The nine-year-old, owned by the Gigginstown House Stud, has twice recovered from a cracked pelvis. "We didn't want to send him back to Aintree with a big weight, that wouldn't be fair," said Gigginstown's racing manager Eddie O'Leary. "He provided us with our first Grand National and we'll never forget him." BBC horse racing correspondent Cornelius Lysaght: "As the first Grand National winner for owner Michael O'Leary's burgeoning Gigginstown House Stud as well as the first novice chaser to win the race in nearly 60 years, Rule The World has his place in history. "Though he ran highly respectably at Punchestown after Aintree, O'Leary had already hinted that, having defied serious injury to reach one of the great pinnacles, he had perhaps done his bit. "What a season for Gigginstown, with success at Aintree, in the Irish National and Cheltenham Gold Cup, but at a price. Rule the World has been retired and there are doubts whether Gold Cup winner Don Cossack will race again."
Gold	This year's Grand National winner Rule The World has been retired.
T5-large (fine-tuned) (**57.14/46.15/57.14**)	A Grand National-winning novice ridden by the brilliant rider Rule The World has been retired.
CONTROL PREFIXES (**55.17/44.44/55.17**)	Winning Grand National hurdler Rule the World has been retired from racing at the age of nine.

Table C.1: XSum generated summaries for T5-large fine-tuned and $\text{BART}_{\text{LARGE}}$ CONTROL PREFIXES, presented alongside the source document and the sole Gold Reference. Source documents are truncated to 300 words if necessary. **R-1/R-2/R-L** are reported in bold. The news/sport Control Prefix, and the related sub-directory control prefix are shown. The Test IDs were randomly selected from the test data.

	XSum
news uk	Vithiya Alphons, 24, has acute myeloid leukaemia but her Sri Lankan background makes the search hard as not many South Asian people are on donor registers. She launched a social media campaign to help find a match. Charity Anthony Nolan said 5,600 have signed up in the last week. "The impact of Vithiya's appeal has been nothing short of incredible," said Ann O'Leary, head of register development, at the charity. "But Vithiya still needs to find her stem cell match and there are people just like her all across the world who are still waiting. "We need to continue to diversify the register so we can find a match for all." Miss Alphons was diagnosed with the aggressive form of blood cancer after falling ill just days after returning for her final year as an optometry student at Cardiff University. She started feeling unwell with severe sickness and a fever, while she had a pain in her leg, so she went to her doctor for tests. She underwent chemotherapy in Cardiff before being well enough to be transferred to a hospital in London, where she is from. After her third course of chemotherapy, Miss Alphons felt better and thought she had beaten her illness. But further tests showed the leukaemia was still in her blood and doctors told her the best option was a stem cell transplant from a donor, which is needed in the next two months. Speaking to BBC's Asian Network Miss Alphons said she is grateful for all the support. "I would just like to say thank you so much to everyone who has registered so far and please, please do carry on registering because you can save my life and you can also save so many other's lives. ...
Gold	Thousands of people have come forward following a worldwide appeal to find a stem cell donor for a Cardiff University student who needs a match in the next two months.
T5-large (fine-tuned) (**39.22/24.49/27.45**)	A student who is struggling to find a stem cell donor has been urged to register as many people as possible.
CONTROL PREFIXES (**50.00/22.22/35.71**)	More than 5,000 people have signed up to the stem cell donor register in a bid to save the life of a Cardiff University student.

Table C.2: XSum generated summaries for T5-large fine-tuned and $\text{BART}_{\text{LARGE}}$ CONTROL PREFIXES, presented alongside the source document and the sole Gold Reference. Source documents are truncated to 300 words if necessary. **R-1/R-2/R-L** are reported in bold. The news/sport Control Prefix, and the related sub-directory control prefix are shown. The Test IDs were randomly selected from the test data.

C.0.2 Data-to-Text

WebNLG+ 2020 Unseen

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 (50) **36.48**	The Pontiac Rageous is a coupe manufactured by Pontiac. It is assembled in Detroit, Michigan and began production in 1997.
CONTROL PREFIXES (48;2,2) **37.51**	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 (50) **49.41**	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 (48;2,2) **47.12**	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 C.3: WebNLG+ 2020 generations: sources are shown in their linearized form as fed to the T5-large based models. The **BLEU** score is reported in bold. Examples are chosen from the *unseen entities* & *unseen* splits. The Source control prefix is highlighted, along with the final Category Control Prefix for the CONTROL PREFIXES (48;2,2) model. The Zero-shot procedure is depicted, e.g. for the Unseen Category *MusicalWork*, the closest embedding belonging to a Seen category in embedding space is Artist.

Higher Human Annotated Score Vs. Automatic Metric

WebNLG+ 2020	
Politician (Seen)	**Source:** <H> Abraham A. Ribicoff <R> spouse <T> Ruth Ribicoff <H> Abraham A. Ribicoff <R> birth Place <T> United States <H> Abraham A. Ribicoff <R> office <T> United States Secretary of Health Education and Welfare <H> United States <R> ethnic Group <T> African Americans
Gold	Abraham A. Ribicoff's birthplace is the USA where he worked for the United States Secretary of Health, Education, and Welfare office, in a country where African Americans are a national ethnic group. His spouse is Ruth Ribicoff.
PREFIX-TUNING (50) **63.61**	Abraham A. Ribicoff, who was married to Ruth Ribicoff, was born in the United States and worked as the United States Secretary of Health, Education, and Welfare. African Americans are an ethnic group in the United States.
CONTROL PREFIXES (48;2) **81.10**	Abraham A. Ribicoff was born in the United States, where African Americans are an ethnic group. He worked as the United States Secretary of Health, Education and Welfare and was married to Ruth Ribicoff.
Building (Seen)	**Source:** <H> Alan B. Miller Hall <R> current Tenants <T> Mason School of Business <H> Alan B. Miller Hall <R> location <T> Williamsburg Virginia <H> Alan B. Miller Hall <R> owner <T> College of William & Mary
Gold	The College of William and Mary, located in Williamsburg Virginia is the location of the Mason School of Business located inside Alan B Miller Hall.
PREFIX-TUNING (50) **79.53**	The Mason School of Business are the current tenants of Alan B Miller Hall, Williamsburg, Virginia, which is owned by the College of William and Mary.
CONTROL PREFIXES (48;2) **76.16**	The Mason School of Business are the current tenants of Alan B Miller Hall in Williamsburg, Virginia. The Hall is owned by the College of William and Mary.

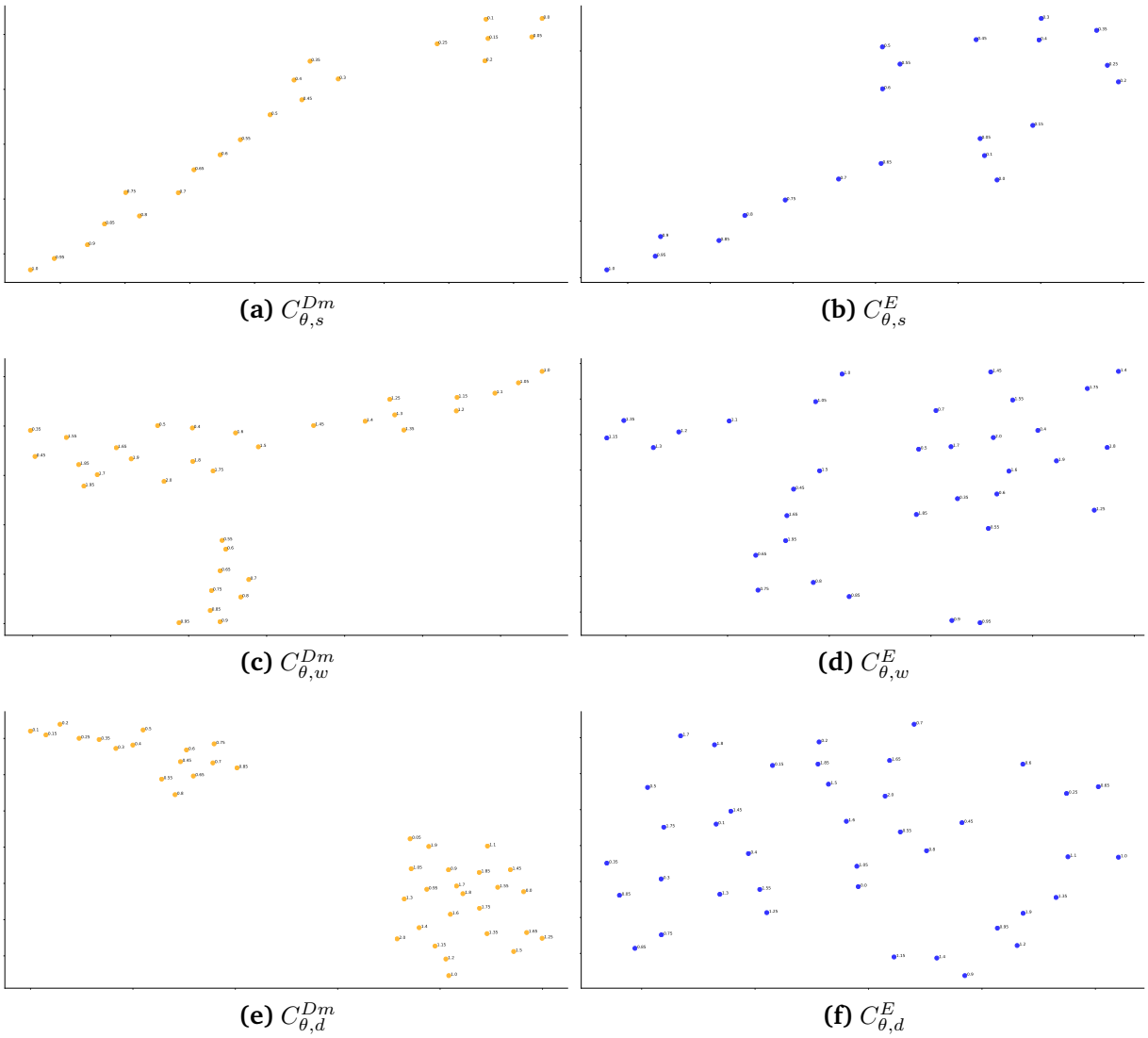
Table C.4: WebNLG+ 2020 generations: sources are shown in their linearized form as fed to the T5-large based models, with the Category Control Prefix signposted. The individual **BLEU** score is shown. These generations were chosen from our internal Human Evaluation where the Gold was annotated with a lower overall assessed score than both CONTROL PREFIXES and PREFIX-TUNING, despite the disparity in BLEU for the Politician example.

C.0.3 Simplification

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 (96;1,1,1,1)	It is thought that the Great Dark Spot is a hole in Neptune's methane cloud deck.
$BART_{LARGE}$ with ACCESS (Martin et al. [2020])	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 (96;1,1,1,1)	Fives is a British sport. It is believed to have its origins in racquet sports.
$BART_{LARGE}$ with ACCESS (Martin et al. [2020])	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 (96;1,1,1,1)	Tagore emulated many different styles of art, including Haida carvings from the west coast of Canada (British Columbia), and woodcuts by Max Pechstein.
$BART_{LARGE}$ with ACCESS (Martin et al. [2020])	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 C.5: *Fixed-LM* vs fine-tuned *controlled* Text Simplification. Generated simplifications from CONTROL PREFIXES (96;1,1,1,1) and $BART_{LARGE}$ with ACCESS (Martin et al. [2020] 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 (96;1,1,1,1) and BART with ACCESS are also randomly selected from one of the five model outputs.

Appendix D: Supplementary Graphics



(a) $C^{Dm}_{\theta,s}$ **(b)** $C^{E}_{\theta,s}$

(c) $C^{Dm}_{\theta,w}$ **(d)** $C^{E}_{\theta,w}$

(e) $C^{Dm}_{\theta,d}$ **(f)** $C^{E}_{\theta,d}$

Figure D.1: t-SNE Visualizations for the Levenshtein similarity (s), WordRank (w), DepTreeDepth (d) control prefixes learnt as part of the simplification model: CONTROL PREFIXES (96;1,1,1,1), where $s, w, d \in \mathcal{A}$. The cross-attention constituents of the control prefixes are ommitted as the representations were not as manifest.

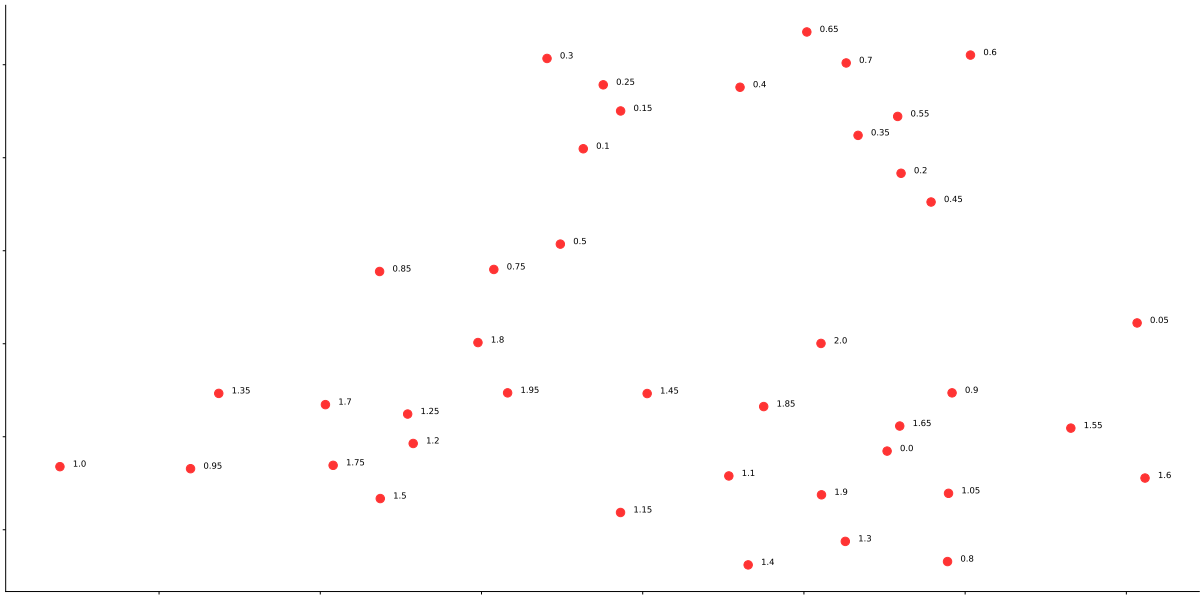


Figure D.2: t-SNE visualizations for Decoder cross-attention constituent of the Length Compression (l) i.e. $C^{Dc}_{\theta,l}$, where $l \in \mathcal{A}$.

Appendix E: Supplementary Results

E.0.1 PREFIX-TUNING + control tokens results

	ϕ%	BLEU	METEOR	TER ↓	BERTScore(F1)
Selected Result (T5-large)					
CONTROL PREFIXES (DART) (48;2)	1.1	51.95	41.07	42.75	0.95
Supplementary Result (T5-large)					
Control Tokens (DART) (50)	1.0	51.72	40.94	42.81	0.95

Table E.1: Supplementary model results on the test set of DART (Radev et al. [2020]) for PREFIX-TUNING + control tokens. Selected result is our top-performing CONTROL PREFIXES model from Table 6.1.

.

	ϕ%	BLEU	NIST	METEOR	R-L	CIDEr
Selected Result (T5-large)						
+Data: DART						
CONTROL PREFIXES (DART) (48;2)	1.0	**44.15**	**6.51**	**0.392**	57.3	**2.04**
Supplementary Result (T5-large)						
+Data: DART						
Control Tokens (DART) (50)	1.0	43.57	6.45	0.389	57.2	2.00

Table E.2: Supplementary results on the E2E Clean test set for PREFIX-TUNING + control tokens. Selected result is our top-performing model from Table 6.4.

	ϕ%	Dev $F_{0.5}$	Test $F_{0.5}$				
			CEFR: A	CEFR: B	CEFR: C	CEFR: N	Overall
Selected Result ($BART_{LARGE}$)							
CONTROL PREFIXES (CEFR) (10;5)	0.5	51.69	63.53	66.70	68.75	63.98	65.21
Supplementary Result ($BART_{LARGE}$)							
Control Tokens (CEFR) (15)	0.3	51.25	63.33	65.76	69.01	62.75	64.72

Table E.3: Supplementary ERRANT $F_{0.5}$ results on the BEA-dev and BEA-test sets for PREFIX-TUNING + control tokens with the CONTROL PREFIXES result from Table 6.5. The Overall $F_{0.5}$ is calculated from the Codalab competition using the inference procedure outlined in Section 6.2.2.

	ϕ%	BLEU S	BLEU U	BLEU **A**	METEOR S	METEOR U	METEOR **A**	TER ↓ S	TER ↓ U	TER ↓ **A**
Selected Result (T5-large)										
+Data: DART										
CONTROL PREFIXES (DART, WebNLG) (48;2,2)	1.4	67.15	**56.41**	**62.27**	46.64	43.18	45.03	31.08	38.78	34.61
Supplementary Result (T5-large)										
+Data: DART										
Control Tokens (DART, WebNLG) (50)	1.0	67.09	55.58	61.89	46.69	42.85	44.91	31.19	38.99	34.77

Table E.4: Supplementary model results on the WebNLG 2017 test set for PREFIX-TUNING + control tokens. Selected result is the top-performing model from Table 6.2.

	ϕ%	BLEU	METEOR	chrF++	TER ↓	BLEURT
Selected Result (T5-large)						
+Data: DART						
CONTROL PREFIXES (DART, WebNLG) (48;2,2)‡	1.0	**55.41**	**0.419**	**0.698**	**0.392**	**0.63**
Supplementary Result (T5-large)						
Control Tokens (DART, WebNLG) (50)	1.0	54.73	0.417	0.693	0.400	0.62

(a) Overall automatic learned and non-learned results reported on the WebNLG+ 2020 test set.

	BLEU S	BLEU U	BLEU UE	METEOR S	METEOR U	METEOR UE	TER ↓ S	TER ↓ U	TER ↓ UE
Selected Result (T5-large)									
+Data: DART									
CONTROL PREFIXES (DART, WebNLG) (48;2,2)‡	60.57	50.39	56.41	0.428	0.409	0.427	0.402	0.400	0.360
Supplementary Result (T5-large)									
Control Tokens (DART, WebNLG) (50)	59.78	49.59	55.75	0.427	0.405	0.424	0.414	0.409	0.362

(b): Granular Results: *Seen* (S), *Unseen Categories* (U) and *Unseen Entities* (UE).

Table E.5: Supplementary model results on the WebNLG+ 2020 test set for PREFIX-TUNING + control tokens. The overall WebNLG+ 2020 test set results **(a)** and results breakdown across categories **(b)**. Selected result is the top-performing model from Table 6.3.

Appendix F: Ethics Checklist

	Yes	No
Section 1: HUMAN EMBRYOS/FOETUSES		
Does your project involve Human Embryonic Stem Cells?		x
Does your project involve the use of human embryos?		x
Does your project involve the use of human foetal tissues / cells?		x
Section 2: HUMANS		x
Does your project involve human participants?	x	
Section 3: HUMAN CELLS / TISSUES		
Does your project involve human cells or tissues? (Other than from "Human Embryos/Foetuses" i.e. Section 1)?		x
Section 4: PROTECTION OF PERSONAL DATA		
Does your project involve personal data collection and/or processing?	x	
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?	x	
Does it involve processing of genetic information?		
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.		x
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?		x
Section 5: ANIMALS		
Does your project involve animals?		x
Section 6: DEVELOPING COUNTRIES		
Does your project involve developing countries?		x
If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?		x
Could the situation in the country put the individuals taking part in the project at risk?		x

Table F.1: Ethics Checklist, part 1. Adapted from the official Imperial College Project Checklist.

	Yes	No
Section 7: ENVIRONMENTAL PROTECTION AND SAFETY		
Does your project involve the use of elements that may cause harm to the environment, animals or plants?		x
Does your project deal with endangered fauna and/or flora /protected areas?		x
Does your project involve the use of elements that may cause harm to humans, including project staff?		x
Does your project involve other harmful materials or equipment, e.g. high-powered laser systems?		x
Section 8: DUAL USE		
Does your project have the potential for military applications?		x
Does your project have an exclusive civilian application focus?		x
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?		x
Does your project affect current standards in military ethics ? e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?		x
Section 9: MISUSE		x
Does your project have the potential for malevolent/criminal/terrorist abuse?		x
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?		x
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?		x
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?		x
SECTION 10: LEGAL ISSUES		
Will your project use or produce software for which there are copyright licensing implications?		x
Will your project use or produce goods or information for which there are data protection, or other legal implications?		x
SECTION 11: OTHER ETHICS ISSUES		
Are there any other ethics issues that should be taken into consideration?		x

Table F.2: Ethics Checklist, part 2. Adapted from the official Imperial College Project Checklist.

Bibliography

[1] Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C., Soatto, S., and Perona, P. (2019). Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6429–6438. pages 55, 62, 63

[2] Aghajanyan, A., Zettlemoyer, L., and Gupta, S. (2020). Intrinsic dimensionality explains the effectiveness of language model fine-tuning. pages 15, 18, 22, 49, 61

[3] Alva-Manchego, F., Martin, L., Scarton, C., and Specia, L. (2019). Easse: Easier automatic sentence simplification evaluation. *arXiv preprint arXiv:1908.04567*. pages 35

[4] Alva-Manchego, F. E., Martin, L., Bordes, A., Scarton, C., Sagot, B., and Specia, L. (2020). ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *CoRR*, abs/2005.00481. pages 35

[5] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors, *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg. pages 31

[6] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *CoRR*, abs/2004.05150. pages 7

[7] Briakou, E., Agrawal, S., Zhang, K., Tetreault, J. R., and Carpuat, M. (2021). A review of human evaluation for style transfer. *CoRR*, abs/2106.04747. pages 59

[8] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *NeurIPS*. pages 16, 17

[9] Bryant, C., Felice, M., Andersen, Ø. E., and Briscoe, T. (2019). The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics. pages 26, 33, 34, 43

[10] Bryant, C., Felice, M., and Briscoe, T. (2017). Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *ACL*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics. pages 33, 34

[11] Buhai, R.-D., Halpern, Y., Kim, Y., Risteski, A., and Sontag, D. (2020). Empirical study of the benefits of overparameterization in learning latent variable models. pages 21

[12] Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., and Charlin, L. (2018). Language gans falling short. *CoRR*, abs/1811.02549. pages 12

[13] Cachola, I., Lo, K., Cohan, A., and Weld, D. S. (2020). TLDR: extreme summarization of scientific documents. *CoRR*, abs/2004.15011. pages 24

[14] Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., and Shimorina, A. (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. pages 12, 32, 33, 36, 58, 59, 64

[15] Celikyilmaz, A., Clark, E., and Gao, J. (2020). Evaluation of text generation: A survey. *CoRR*, abs/2006.14799. pages 11, 12, 58

[16] Chaganty, A., Mussmann, S., and Liang, P. (2018). The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics. pages 57

[17] Chang, H., Nguyen, T. D., Murakonda, S. K., Kazemi, E., and Shokri, R. (2020). On adversarial bias and the robustness of fair machine learning. pages 64

[18] Cohan, A., Ammar, W., van Zuylen, M., and Cady, F. (2019). Structural scaffolds for citation intent classification in scientific publications. *CoRR*, abs/1904.01608. pages 24

[19] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*. pages 3

[20] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. A. (2020). Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305. pages 15

[21] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929. pages 5

[22] Dou, Z., Liu, P., Hayashi, H., Jiang, Z., and Neubig, G. (2020). Gsum: A general framework for guided neural abstractive summarization. *CoRR*, abs/2010.08014. pages 23, 57, 63

[23] Dušek, O., Howcroft, D. M., and Rieser, V. (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. pages 31

[24] Fabbri, A., Ng, P., Wang, Z., Nallapati, R., and Xiang, B. (2020a). Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4508–4513, Online. Association for Computational Linguistics. pages 13

[25] Fabbri, A. R., Kryscinski, W., McCann, B., Xiong, C., Socher, R., and Radev, D. R. (2020b). Summeval: Re-evaluating summarization evaluation. *CoRR*, abs/2007.12626. pages 57

[26] Fan, A., Gardent, C., Braud, C., and Bordes, A. (2019). Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4186–4196, Hong Kong, China. Association for Computational Linguistics. pages 30

[27] Floridi, L. and Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:1–14. pages 64

[28] Freitag, M., Grangier, D., and Caswell, I. (2020). BLEU might be guilty but references are not innocent. *CoRR*, abs/2004.06063. pages 12

[29] Gao, J., Galley, M., and Li, L. (2019). Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298. pages 12

[30] Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (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. pages 30, 31, 33

[31] Gehrmann, S., Adewumi, T. P., Aggarwal, K., Ammanamanchi, P. S., Anuoluwapo, A., Bosselut, A., Chandu, K. R., Clinciu, M., Das, D., Dhole, K. D., Du, W., Durmus, E., Dusek, O., Emezue, C., Gangal, V., Garbacea, C., Hashimoto, T., Hou, Y., Jernite, Y., Jhamtani, H., Ji, Y., Jolly, S., Kumar, D., Ladhak, F., Madaan, A., Maddela, M., Mahajan, K., Mahamood, S., Majumder, B. P., Martins, P. H., McMillan-Major, A., Mille, S., van Miltenburg, E., Nadeem, M., Narayan, S., Nikolaev, V., Niyongabo, R. A., Osei, S., Parikh, A. P., Perez-Beltrachini, L., Rao, N. R., Raunak, V., Rodriguez, J. D., Santhanam, S., Sedoc, J., Sellam, T., Shaikh, S., Shimorina, A., Cabezudo, M. A. S., Strobelt, H., Subramani, N., Xu, W., Yang, D., Yerukola, A., and Zhou, J. (2021). The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672. pages 14, 29, 42, 66, 67

[32] Grangier, D. and Auli, M. (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. pages 23

[33] Guo, H., Tan, B., Liu, Z., Xing, E. P., and Hu, Z. (2021). Text generation with efficient (soft) q-learning. *CoRR*, abs/2106.07704. pages 4

[34] Guo, Q., Jin, Z., Dai, N., Qiu, X., Xue, X., Wipf, D., and Zhang, Z. (2020). $\sqrt{}^2$: A plan-and-pretrain approach for knowledge graph-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 100–106, Dublin, Ireland (Virtual). Association for Computational Linguistics. pages 41, 42, 59

[35] Han, A. L. and Wong, D. F. (2016). Machine translation evaluation: A survey. *CoRR*, abs/1605.04515. pages 11, 12

[36] Harkous, H., Groves, I., and Saffari, A. (2020). Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. *CoRR*, abs/2004.06577. pages 42

[37] Hendrycks, D. and Gimpel, K. (2016). Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415. pages 8

[38] Holtzman, A., Buys, J., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *CoRR*, abs/1904.09751. pages 10

[39] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R., editors, *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. pages 16

[40] Howcroft, D. M., Belz, A., Clinciu, M.-A., Gkatzia, D., Hasan, S. A., Mahamood, S., Mille, S., van Miltenburg, E., Santhanam, S., and Rieser, V. (2020). Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics. pages 59

[41] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685. pages 15, 19, 22, 50, 61, 62, 63

[42] Jiang, C., Maddela, M., Lan, W., Zhong, Y., and Xu, W. (2020). Neural CRF model for sentence alignment in text simplification. *CoRR*, abs/2005.02324. pages 35, 47

[43] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558. pages 23

[44] Johnson, W. (1944). Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15. pages 67

[45] Kale, M. (2020). Text-to-text pre-training for data-to-text tasks. *CoRR*, abs/2005.10433. pages 9, 30, 37, 40, 41

[46] Katsumata, S. and Komachi, M. (2020). Stronger baselines for grammatical error correction using pretrained encoder-decoder model. *CoRR*, abs/2005.11849. pages 43

[47] Keskar, N., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858. pages 23

[48] Khashabi, D., Stanovsky, G., Bragg, J., Lourie, N., Kasai, J., Choi, Y., Smith, N. A., and Weld, D. S. (2021). GENIE: A leaderboard for human-in-the-loop evaluation of text generation. *CoRR*, abs/2101.06561. pages 1, 11, 29, 57

[49] Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. (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. pages 1, 23

[50] Kilickaya, M., Erdem, A., Ikizler-Cinbis, N., and Erdem, E. (2017). Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, Valencia, Spain. Association for Computational Linguistics. pages 12

[51] Kincaid, J. P., Fishburne Jr., R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. pages 35

[52] Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). Opennmt: Open-source toolkit for neural machine translation. *CoRR*, abs/1701.02810. pages 11

[53] Konstas, I. and Lapata, M. (2012). Unsupervised concept-to-text generation with hypergraphs. pages 752–761. pages 30

[54] Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S. R., Socher, R., and Rajani, N. F. (2020). Gedi: Generative discriminator guided sequence generation. *CoRR*, abs/2009.06367. pages 23

[55] Läubli, S., Castilho, S., Neubig, G., Sennrich, R., Shen, Q., and Toral, A. (2020). A set of recommendations for assessing human-machine parity in language translation. *CoRR*, abs/2004.01694. pages 12

[56] Lavie, A. and Agarwal, A. (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. pages 12

[57] Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691. pages 17, 18, 21, 22, 23, 41, 49, 50, 63

[58] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (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. pages 3, 9, 45

[59] Li, C., Farkhoor, H., Liu, R., and Yosinski, J. (2018). Measuring the intrinsic dimension of objective landscapes. *CoRR*, abs/1804.08838. pages 15

[60] Li, J., Galley, M., Brockett, C., Gao, J., and Dolan, B. (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. pages 67

[61] Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. pages 1, 7, 9, 15, 19, 21, 22, 23, 36, 44, 45, 50, 61

[62] Lichtarge, J., Alberti, C., Kumar, S., Shazeer, N., Parmar, N., and Tong, S. (2019). Corpora generation for grammatical error correction. 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 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics. pages 43

[63] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. pages 12

[64] Lin, Z., Madotto, A., and Fung, P. (2020). Exploring versatile generative language model via parameter-efficient transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 441–459, Online. Association for Computational Linguistics. pages 16

[65] Linderman, G. C. and Steinerberger, S. (2017). Clustering with t-sne, provably. *CoRR*, abs/1706.02582. pages 52

[66] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. pages 1, 16, 17, 23

[67] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2021b). GPT understands, too. *CoRR*, abs/2103.10385. pages 17

[68] Liu, Y. and Lin, Z. (2019). Unsupervised pre-training for natural language generation: A literature review. *CoRR*, abs/1911.06171. pages 5

[69] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692. pages 9

[70] Logeswaran, L., Lee, A., Ott, M., Lee, H., Ranzato, M., and Szlam, A. (2020). Few-shot sequence learning with transformers. *CoRR*, abs/2012.09543. pages 17

[71] Loshchilov, I. and Hutter, F. (2017). Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101. pages 37

[72] Maaten, L. V. D. and Hinton, G. E. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605. pages 52

[73] Mahabadi, R. K., Henderson, J., and Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. *CoRR*, abs/2106.04647. pages 15, 16, 19

[74] Martin, L., Fan, A., de la Clergerie, É., Bordes, A., and Sagot, B. (2020). Multilingual unsupervised sentence simplification. *CoRR*, abs/2005.00352. pages 36, 46, 47, 73

[75] Martin, L., Sagot, B., de la Clergerie, É., and Bordes, A. (2019). Controllable sentence simplification. *CoRR*, abs/1910.02677. pages 35, 36, 47, 54

[76] Mathur, N., Baldwin, T., and Cohn, T. (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. pages 12

[77] McKeown, K. and Radev, D. R. (1995). Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, page 74–82, New York, NY, USA. Association for Computing Machinery. pages 57

[78] Moryossef, A., Dagan, I., and Goldberg, Y. (2019). Improving quality and efficiency in plan-based neural data-to-text generation. pages 32

[79] Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *CoRR*, abs/1808.08745. pages 34, 57

[80] Novikova, J., Dusek, O., and Rieser, V. (2017). The E2E dataset: New challenges for end-to-end generation. *CoRR*, abs/1706.09254. pages 31

[81] Paetzold, G. and Specia, L. (2016). SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics. pages 35

[82] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics. pages 12

[83] Pasricha, N., Arcan, M., and Buitelaar, P. (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. pages 41, 42

[84] Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304. pages 11

[85] Peters, M. E., Ruder, S., and Smith, N. A. (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. pages 15

[86] Rabinovich, E., Mirkin, S., Patel, R. N., Specia, L., and Wintner, S. (2016). Personalized machine translation: Preserving original author traits. *CoRR*, abs/1610.05461. pages 24

[87] Radev, D. R., Zhang, R., Rau, A., Sivaprasad, A., Hsieh, C., Rajani, N. F., Tang, X., Vyas, A., Verma, N., Krishna, P., Liu, Y., Irwanto, N., Pan, J., Rahman, F., Zaidi, A., Mutuma, M., Tarabar, Y., Gupta, A., Yu, T., Tan, Y. C., Lin, X. V., Xiong, C., and Socher, R. (2020). DART: open-domain structured data record to text generation. *CoRR*, abs/2007.02871. pages 30, 31, 32, 39, 40, 43, 76

[88] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. pages 3, 10

[89] Radiya-Dixit, E. and Wang, X. (2020). How fine can fine-tuning be? learning efficient language models. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2435–2443, Online. PMLR. pages 15

[90] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67. pages 3, 8, 37

[91] Rapin, J. and Teytaud, O. (2018). Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad. pages 36

[92] Rebuffi, S.-A., Bilen, H., and Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 506–516. Curran Associates, Inc. pages 16

[93] Rello, L., Baeza-Yates, R., Bott, S., and Saggion, H. (2013). *Simplify or Help? Text Simplification Strategies for People with Dyslexia*. Association for Computing Machinery, New York, NY, USA. pages 35

[94] Ribeiro, L. F. R., Schmitt, M., Schütze, H., and Gurevych, I. (2020). Investigating pretrained language models for graph-to-text generation. *arXiv*. pages 32, 36, 37, 40, 41

[95] Schick, T. and Schütze, H. (2020). It's not just size that matters: Small language models are also few-shot learners. *CoRR*, abs/2009.07118. pages 16

[96] Schmitt, M., Sharifzadeh, S., Tresp, V., and Schütze, H. (2020). An unsupervised joint system for text generation from knowledge graphs and semantic parsing. pages 7117–7130. pages 32

[97] Scialom, T., Martin, L., Staiano, J., de la Clergerie, É. V., and Sagot, B. (2021). Rethinking automatic evaluation in sentence simplification. *CoRR*, abs/2104.07560. pages 13, 14, 35, 48

[98] Sellam, T., Das, D., and Parikh, A. (2020). BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics. pages 13

[99] Semeniuta, S., Severyn, A., and Gelly, S. (2019). On accurate evaluation of GANs for language generation. pages 12

[100] Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations. *CoRR*, abs/1803.02155. pages 8

[101] Shazeer, N. and Stern, M. (2018). Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235. pages 37

[102] Shin, T., Razeghi, Y., au2, R. L. L. I., Wallace, E., and Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. pages 17

[103] Shleifer, S. and Rush, A. M. (2020). Pre-trained summarization distillation. *CoRR*, abs/2010.13002. pages 45

[104] Stahlberg, F. and Kumar, S. (2021). Synthetic data generation for grammatical error correction with tagged corruption models. *CoRR*, abs/2105.13318. pages 26, 33

[105] Subramani, N., Bowman, S. R., and Cho, K. (2020). Can unconditional language models recover arbitrary sentences? pages 17

[106] Sun, S., Shapira, O., Dagan, I., and Nenkova, A. (2019). How to compare summarizers without target length? pitfalls, solutions and re-examination of the neural summarization literature. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 21–29, Minneapolis, Minnesota. Association for Computational Linguistics. pages 48, 67

[107] Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., Zhao, Y., Lu, Y., Liu, W., Wu, Z., Gong, W., Liang, J., Shang, Z., Sun, P., Liu, W., Ouyang, X., Yu, D., Tian, H., Wu, H., and Wang, H. (2021). ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137. pages 17

[108] Swayamdipta, S., Thomson, S., Lee, K., Zettlemoyer, L., Dyer, C., and Smith, N. A. (2018). Syntactic scaffolds for semantic structures. In *EMNLP*. pages 24

[109] Takahashi, Y., Katsumata, S., and Komachi, M. (2020). Grammatical error correction using pseudo learner corpus considering learner's error tendency. pages 27–32. pages 26, 33

[110] Taylor, W. L. (1953). "cloze procedure": a new tool for measuring readability. *Journalism Mass Communication Quarterly Natural*, 30:415–433. pages 9

[111] Tsimpoukelli, M., Menick, J., Cabi, S., Eslami, S. M. A., Vinyals, O., and Hill, F. (2021). Multimodal few-shot learning with frozen language models. *CoRR*, abs/2106.13884. pages 23

[112] van der Lee, C., Gatt, A., Miltenburg, E., and Krahmer, E. (2021). Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech Language*, 67:101151. pages 57

[113] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*. pages 3, 5, 6

[114] Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. pages 5797–5808. pages 6

[115] Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. (2019). Neural text generation with unlikelihood training. *CoRR*, abs/1908.04319. pages 12

[116] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280. pages 4

[117] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (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. pages 36

[118] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144. pages 11

[119] Xu, W., Napoles, C., Pavlick, E., Chen, Q., and Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415. pages 35

[120] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777. pages 3, 10, 45, 68

[121] Zhang, J. O., Sax, A., Zamir, A., Guibas, L., and Malik, J. (2020a). Side-tuning: A baseline for network adaptation via additive side networks. pages 16

[122] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). BERTScore: Evaluating text generation with bert. In *International Conference on Learning Representations*. pages 12, 13, 57

[123] Zhang, X. and Lapata, M. (2017). Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*. pages 35, 46

[124] Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., and Dolan, B. (2020c). 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, Online. Association for Computational Linguistics. pages 3