Multi-Attribute Linguistic Tuning for Controlled Paraphrase Generation

Anonymous ACL submission

Abstract

We present a novel approach to paraphrase generation that enables precise control and fine-tuning of 40 linguistic attributes for English. Our model is an encoder-decoder architecture that takes as input a source sentence and desired linguistic attributes, and produces paraphrases of the source that satisfy the desired attributes. To guarantee high-quality outputs at inference time, our method is equipped with a quality control mechanism that gradually adjusts the embedding of linguistic attributes to find the nearest and most attainable configuration of desired attributes for paraphrase generation. We evaluate the effectiveness of our method by comparing it to recent controllable generation models. Experimental results demonstrate that the proposed model outperforms baselines in generating paraphrases that satisfy desired linguistic attributes.

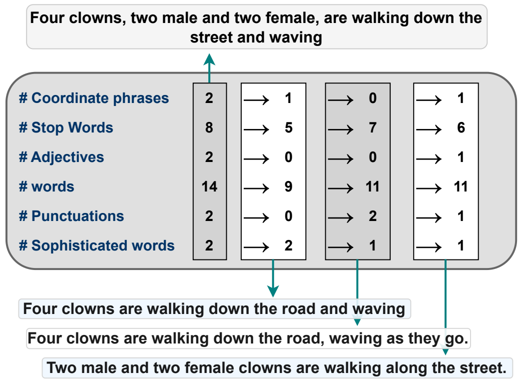


Figure 1: We aim to transform a given sentence into multiple paraphrases, each satisfying distinct linguistic attributes. Our model takes a source sentence and a set of target linguistic attributes and generates a paraphrase optimized to satisfy the target attributes. Here we show three paraphrases with different linguistic attributes generated for the source sentence.

1 Introduction

Controllable text generation (CTG) is the task of generating texts that satisfy desired attributes (Ficler and Goldberg, 2017; Jin et al., 2022). CTG has received significant attention recently following the improvements in text generation with large language models (LLMs) (Dathathri et al., 2020; Qin et al., 2022; Mireshghallah et al., 2022; Liu et al., 2023b; Zhang and Song, 2022a; Yang et al., 2023; Bandel et al., 2022).

Controlled paraphrase generation (CPG) is a sub-task of CTG that focuses on generating paraphrases of a source text that satisfy predetermined linguistic attributes. CPG allows users to shape given text to align with precise linguistic objectives, and is a more challenging task than unrestricted text generation (Sun et al., 2023). CPG has applications in text simplification (Lee and Lee, 2023b; Lee et al., 2021; Vajjala and Lučić, 2018; Zhang and Lapata, 2017; Xu et al., 2015), toxicity control (Zheng et al., 2023; Zhang and Song, 2022b; Liu et al., 2021), emotion and topic control (Yang et al., 2023), and personalized dialog generation (Huang et al., 2023b; Niu and Bansal, 2018).

CPG has the potential to generate data that challenges existing models from a linguistic perspective,[1] produce text with varying levels of linguistic complexity for language learners (Perkoff et al., 2023; Ashok Kumar et al., 2023; Wambsganss et al., 2022) or data augmentation (Iyyer et al., 2018a; Malandrakis et al., 2019), and make text accessible through language simplification (Lin et al., 2021). The main challenge in CPG is to generate text that preserves the meaning of the source and satisfies the desired linguistic attributes. While existing work has explored this balance, most work has focused on a limited set of attributes. Accommodating a wider array of linguistic attributes in CPG is crucial because it improves the flexibility

[1] Especially in the current era of NLP, where datasets often contain examples that lack enough linguistic complexity, leading to a plateau in model performance improvements.

and engagement for diverse audiences including language learners.

We introduce LINGCONV, a novel encoder-decoder CPG model that simultaneously controls multiple objectives (linguistic attributes) by adaptively integrating linguistic attributes into the decoding process of LLMs and implementing a robust quality-control mechanism for high-quality CPG. We will consider a set of 40 attributes, listed in Appendix A. LingConv represents the target attributes in a dense representation space using an embedding layer and controls the generation process by integrating the attribute embeddings with decoder inputs through element-wise addition, so that the rich attribute representation will be available as a strong signal and attended to by the transformer's self-attention. LINGCONV is trained in a supervised manner using triplets of source sentences, target attributes, and reference paraphrases. The objective is to generate paraphrases that satisfy the target linguistic attributes and preserve the original meaning of the source. To ensure high-quality outputs at inference time, LINGCONV implements a novel quality control (QC) mechanism for linguistic attributes. Since not all combinations of desired linguistic attributes are feasible for a given source, the QC component finds the closest set of attainable attributes. This is achieved through a linguistic attribute classifier, which fine-tunes the generation process based on feedback from its error signals (obtained from back-propagation). In addition, the QC component has a semantic consistency classifier to assess the semantic relevance of updated generations. It works based on an innovative and efficient line-search algorithm to determine the optimized magnitude of updates and iteratively refines the generations until no further improvement can be made. This mechanism ensures that LINGCONV generates paraphrases that closely align with the desired linguistic attributes.

To the best of our knowledge, LINGCONV is the first system designed to generate paraphrases with fine-grained linguistic attributes. The 40 linguistic attributes span lexical, syntactic, topical, discourse, and semantic aspects of language, extracted using tools developed by Lu (2010), Lu (2012), and Lee and Lee (2023a). The list of linguistic attributes and the rationale for choosing them are included in Appendix A. Figure 1 illustrates an example of linguistic control of a source sentence into three variations.

Extensive experiments show that our approach outperforms baselines by a substantial margin of 58% in generating text that satisfies desired linguistic attributes and preserves semantic consistency and fluency. The QC approach results in a further improvement of 9%. Furthermore, we show the application of our approach in data augmentation. The synthetic data generated by LINGCONV according to linguistic attributes of high/low complexity affect the downstream model differently. We find that the linguistic attributes of augmented data, and their relation to the attributes of the original data, directly affect the effectiveness of data augmentation. Then, we show how LingConv enables robust, successful augmentation through CPG.

Finally, we conduct further experiments, presented in Appendix 4.2, to understand which linguistic attributes are easy or hard to control for text generation, and why.

2 Related Work

We discuss developments in controllable text generation. Colin and Gardent (2018) show that the inclusion of a textual syntactic constraint to the paraphrase generation process produces syntactically diverse outputs. Kajiwara (2019) proposes a two-stage model for generating paraphrases. First, extract the keywords that should be modified. Second, generate a paraphrase with the condition of excluding those words. Qian et al. (2019) realize diverse paraphrase generation through training multiple paraphrase generators simultaneously that are guided by a discriminator network to enforce their outputs to be discriminable, and a paraphrase discriminator that ensures the output is semantically consistent. Chen et al. (2019) developed a dataset where given a source and an exemplar, the paraphrase should follow the syntax of the exemplar. FSET (Kazemnejad et al., 2020) performs paraphrasing in three steps. Given a source sentence s, it retrieves the most similar sentence p and its associated paraphrase q from a bank of paraphrase pairs. Then, it computes the edits required to change p into q. Then, it applies those edits onto s to generate a paraphrase for s. This process improves the quality and diversity of the generations. SCSVED (Chen et al., 2020) is a variational autoencoder that uses two encoder networks, making use of ground-truth targets to disentangle the semantic and syntactic. Diverse generations are realized by modifying the syntactic latent variable and keeping the semantic latent variable constant.

The SUP (Yang et al., 2021) framework uses a conditional VAE with the syntax structure to learn unsupervised SPG. GCPG (Yang et al., 2022) is a unified framework for CPG that works by concatenating the conditions to the input of an encoder-decoder model, supporting keyword constraints and syntactic conditions. Wahle et al. (2023) proposes splitting the task of paraphrasing into separate paraphrase types based on the linguistic variable being changed.

An alternative approach to CTG research focuses on energy-based models that sample from a latent space using ordinary differential equation solvers (Kumar et al., 2021; Wang et al., 2019b; Gu et al., 2023; Liu et al., 2023a).

A notable mention is the Plug and Play Language Model (PPLM) (Dathathri et al., 2020), which does not require training the language model and only trains an attribute classifier. At inference time, it computes the gradient of the classifier with respect to the hidden state, and simultaneously updates the hidden state towards maximizing the attribute probability and also towards maximizing the language model probability $p(x)$. This way, it ensures that the sentence remains fluent and is moved towards the target attribute. However, this approach is slow due to extensive computations and updates during each generation step. FUDGE (Yang and Klein, 2021) computes the probability of the next token conditioned on the desired attribute: $p(x_t|x_{<t}, c) \propto p(x_t|x_{<t})p(a|x_{\leq t})$.

QCPG (Bandel et al., 2022) controls for three attributes in paraphrase generation: semantic similarity, and syntactic and lexical variation with respect to the source. KCN (Zeng et al., 2019) and BOLT (Liu et al., 2023b) control the presence of specific keywords in the paraphrase. Methods of syntactically-controlled paraphrase generation (SPG) include ParaAMR (Huang et al., 2023a), which rotates the abstract meaning representation (AMR) tree; reordering of the segments in a sentence parse tree (Goyal and Durrett, 2020), and using templates of constituency parses (Iyyer et al., 2018b). SynPG (Huang and Chang, 2021) disentangles the semantics and syntax embeddings by adding the sentence parse tree as additional features and performs SPG. Similarly, AMRPG (Huang et al., 2022) adds the AMR tree as an added feature to allow for SPG. ParaMac (Liu et al., 2022) uses a language model along with word substitution, permutation, and lexical diversity ranking for paraphrase generation.

Previous works have mainly focused on the training phrase and a narrow set of linguistic attributes, and lack quality control mechanisms at inference time. LINGCONV addresses these limitations using 40 lexical, syntactic, semantic, and discourse linguistic attributes, along with a robust quality control mechanism that operates at inference time.

3 LingConv

3.1 Problem Formlation

Consider the dataset $\mathcal{D} = \{(s_i, t_i, l_i^t)\}_{i=1}^N$, where each triplet contains a source sentence (s), a target sentence (t), and the gold linguistic attributes of the target sentence ($l^t \in \mathbb{R}^k$, represented by real numbers). The source and target sentences in each triplet are paraphrases of one another. The task is to map from $(s, l^t) \to t$, such that the output t is a paraphrase of s and its linguistic attributes correspond precisely to the target (desired) l^t.[2]

3.2 LingConv Architecture

Overview LINGCONV is a seq2seq model consisting of three main components, illustrated in Figure 2: encoder-decoder (paraphrase generator), linguistic attribute predictor, and quality control components. The encoder-decoder component incorporates linguistic attributes in the generation process. The linguistic attribute predictor estimates attributes of generated text, allowing for backpropagation of linguistic attribute error. At inference, the quality control component iteratively adjusts inputs to guide outputs towards desired attributes. Given the source sentence and target attributes, the model is trained with a single objective function of conditional generation of paraphrases.

Encoder-Decoder is an extended T5 (Raffel et al., 2020) model. Specifically, in order to effectively guide the model toward generating desired outputs, we propose to embed the linguistic attributes l^t into a dense vector representation and integrate it with T5's *decoder* inputs.[3] To achieve this goal, we add the embedding of the target linguistic vector $\mathbf{l}^t$ to the first token of the decoder

[2]The linguistic attributes of the source sentence (l^s) can be considered as another input. However, we found that they are redundant, and do not result in increased performance.

[3]We also experimented with adding linguistic embeddings to all tokens of the decoder input, concatenating to the decoder inputs (equivalent to prompt tuning), concatenation/addition to encoder inputs, concatenating/adding to encoder outputs, and fusing to encoder outputs using a linear layer. In general, decoder injections were better than encoder injections. Decoder first-token-addition was the best-performing overall.

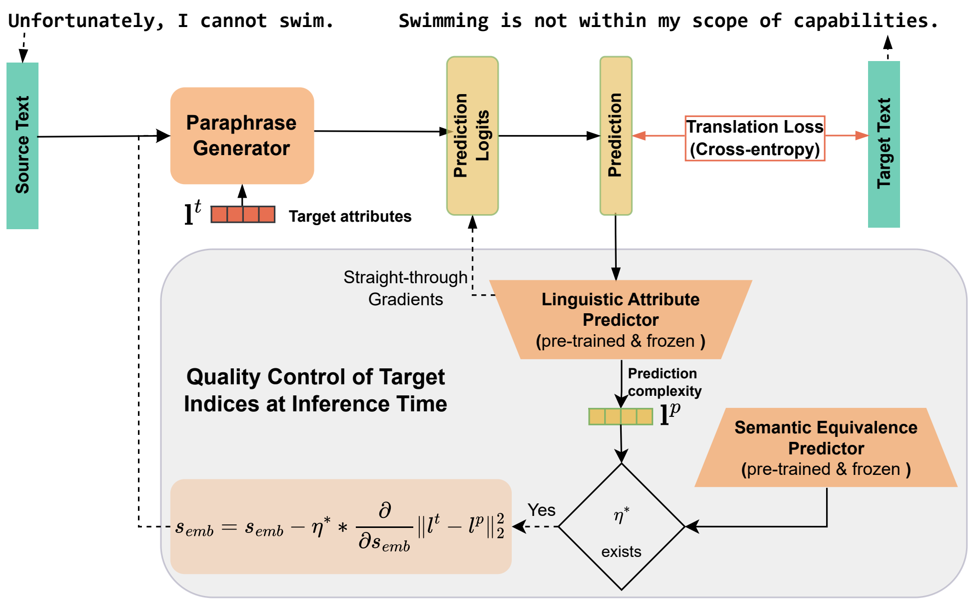


Figure 2: LINGCONV Architecture: The paraphrase generator extends the T5 model by incorporating linguistic attributes into the decoder inputs. Linguistic attributes of the source ($\mathbf{l}^s$) and target ($\mathbf{l}^t$) are embedded and fused with the generation using element-wise addition to the decoder inputs. In addition, the linguistic attribute predictor estimates attributes of the generated text, which facilitates backpropagation of the linguistic attribute error. During inference, the quality control mechanism iteratively adjusts inputs to guide outputs towards desired attributes. The model is trained with a dual objective of semantic equivalence and linguistic attribute adherence.

inputs, which corresponds to the beginning of sentence token <bos>:

$$Y'(l^t) = \begin{cases} Y_i \otimes \mathrm{LE}(l^t) & \text{if } i = 0 \\ Y_i & \text{otherwise,} \end{cases} \tag{1}$$

where Y is the decoder input embedding, LE is the linguistic attribute embedding layer, $\otimes$ indicates the element-wise addition operation, and Y' is the updated decoder inputs. LE is a fully connected layer from $\mathbb{R}^k$ to $\mathbb{R}^d$, where d is the dimension of text input embeddings.

Objective We train our model using cross entropy loss (2):

$$\ell_{CE}(s_i, t_i) = \sum_{j=0}^{len(y)-1} -\log p(y_i^{(j)}|x_i, y^{<j}), \tag{2}$$

where $p(y_i^{(j)}|x_i, y^{<j})$ is the probability of the model predicting the j-th token in the target sequence given the source sequence x_i and the previous tokens $y^{<j}$ in the target sequence; this loss translates the source sentence to a semantically equivalent sentence as induced by our choice of training data (only paraphrase examples). At test time, the model takes a source sentence, the linguistic attributes of the source sentence, and the desired linguistic attributes; and generates an output using auto-regressive greedy decoding.

Linguistic Attribute Predictor (LP) estimates the linguistic attributes of a given generation. This component is independently pre-trained and frozen. It allows for differentiable computation of linguistic attributes and thus backpropagation of the error. Moreover, it helps us avoid the computationally intensive task of calculating 40 linguistic attributes for each generated text within the training process. The component is pre-trained to provide a precise and efficient estimation of these attributes. We implement the linguistic predictor (LP) using a T5 encoder followed by a projection layer, and it is trained by minimizing the mean squared error of the predicted linguistic attributes of each text ($\mathrm{LP}(x) = l^p$ in Figure 2) from its gold attributes (l^x) as follows:

$$\ell_{disc}(x) = \|\mathrm{LP}(x) - l^x\|_2^2. \tag{3}$$

It is not possible to backpropagate the loss through a discrete prediction resulting from an *argmax* operation. Therefore, we apply Straight-through Gradient Estimation (Bengio et al., 2013) to the linguistic attribute predictor, so the gradient is propagated to the prediction logits through the multiplication of the prediction logits and the regressor's token embedding matrix, further described in Appendix B.1.

Semantic Equivalence Classifier (SE) quantifies semantic equivalence of a pair of sentences, and is used in the quality control algorithm. We implement SE using a T5 encoder followed by a projection layer, which is pre-trained by minimizing the following contrastive loss:

$$\ell_{sem}(s,t) = -\log \frac{\mathrm{SE}(s,t)}{\sum_{t' \in \mathcal{N}(s)} \mathrm{SE}(s,t')}, \qquad (4)$$

where $\mathcal{N}(s)$ is the set of negative paraphrases of s. The loss maximizes the probability of valid paraphrases (s,t) and minimizes the probability of invalid paraphrases (s,t'). For a mini-batch of size m, $m-1$ samples are used as negative paraphrases for the remaining sample.

Quality Control To ensure high-quality outputs, we propose a quality control mechanism to use at inference time. The idea is to iteratively adjust the input sentence embeddings to gradually steer the model's output toward the target attributes, l^t. For this purpose, we apply an iterative refinement process (Padmakumar et al., 2023), which updates the model's input with small, progressive changes to allow a smoother transition to significantly different target attributes by taking repeated steps of small conversions. Algorithm 1 shows the process. Initially, we freeze the parameters of the generation model and set input sentence embeddings as our parameter of interest. The model then generates an initial output $\hat{t}$ (line 4 in Algorithm 1). We use the linguistic attribute predictor component to predict the linguistic attributes of this generation. We compute the mean squared error, l_0, between the predicted attributes and the target attributes (line 5), and determine the gradient g of the loss relative to the source sentence embeddings (line 6). We find an effective step size to update the parameters in the negative direction of this gradient. For this purpose, we employ a modified line search algorithm (Armijo, 1966; Boyd and Vandenberghe, 2004) (lines 11–31). Specifically, we modify the line search algorithm to return the smallest viable step size and iteratively make edits to the input to get closer to the target attributes. The resulting generation should adhere to two conditions: (a): the predicted linguistic attribute error should be less than l_0, and (b): the "semantic equivalence" probability should be greater than a threshold τ. These conditions ensure linguistic accuracy (guaranteeing that the new generation has smaller linguistic errors than the original one) and semantic fidelity in the generation. The algorithm stops when no viable step size is found within the search space of the line search, indicating the generation has reached its optimal state.

3.3 Training Data Preparation

To ensure that the training algorithm converges more quickly, we discretize each linguistic attribute into several bins, using 20 bins in our experiments. In addition, we utilize the bidirectional equivalence inherent in paraphrase pairs to enrich our training set with augmented data. First, we augment the data by reversing the order of source and target sentences: $\{t_i, s_i, l_i^t, l_i^s\}$. Second, because a sentence is inherently a paraphrase of itself, we further augment the data with self-paraphrase pairs: $\{s_i, s_i, l_i^s, l_i^s\}$ and $\{t_i, t_i, l_i^t, l_i^t\}$. These strategies increase the diversity and volume of the training data, which potentially helps prevent overfitting and improves the model's ability to generalize to new, unseen examples. In addition, they strengthen the semantic consistency within and across paraphrase pairs, which potentially improves model's understanding and generation capabilities.

4 Experiments

Data We train models using a combination of the Microsoft Research paraphrase corpus (MRPC) (Dolan and Brockett, 2005), semantic textual similarity benchmark (STS-B) (Cer et al., 2017), and Quora question pairs.[4] We only use the *positive* samples in these datasets to constitute semantically equivalent text pairs. Appendix C provides more details.

Baselines We use the following baselines:

- **Copy**: the output is a copy of the input text.
- **Reference**: the output is the ground-truth target paraphrase from the dataset.
- **T5-FT**: a standard T5 model that lacks linguistic attribute control capabilities, fine-tuned on the dataset of paraphrase pairs.
- **FUDGE** (Yang and Klein, 2021): controlled text generation with future discriminators performs attribute control by weighting the token-

[4] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Model	BERTScore↑	MSE(l^t)↓	MSE(l^s)↑	Overall↑	Novel Target Challenge			
					BERTScore^F ↑	MSE(l^t)↓	MSE(l^s)↑	Overall↑
Ref	100.0	0.00	0.96	0.85	94.4	9.82	0.96	0.19
Copy	94.4	0.96	0.00	0.32	100.0	9.86	0.00	0.33
T5-FT	94.2	1.02	0.29	0.36	97.8	9.86	0.29	0.27
Llama	91.0	2.17	**1.80**	0.35	**92.8**	8.90	2.44	0.26
BOLT	90.6	1.11	1.06	0.36	90.4	7.47	1.83	0.21
Fudge	92.0	0.85	1.06	0.45	92.5	7.22	3.11	0.37
QCPG	**95.3**	0.58	0.78	**0.55**	91.4	5.61	3.25	0.41
Lingconv	95.2	0.58	0.73	0.54	92.0	3.69	4.39	0.59
+QC	95.2	**0.52**	0.72	**0.55**	91.5	**2.89**	**6.20**	**0.71**

Table 1: Controlled generation performance across evaluation metrics. Mean squared error (MSE) values reflect how close the linguistic attributes of the generated paraphrase are to the target (MSE(l^t)↓) or source (MSE(l^s)↑).

Model	Lexical	Syntactic	Discourse	Macro-MSE(l^t)
Ref	12.62	8.89	5.91	9.14
Copy	12.66	8.87	6.19	9.24
T5-FT	12.73	8.82	6.16	9.24
Llama	10.88	8.37	5.56	8.27
BOLT	9.36	7.23	3.21	6.60
Fudge	9.54	6.83	**2.34**	6.23
QCPG	7.64	4.30	5.46	5.80
Lingconv	4.25	3.08	4.70	4.01
+QC	**3.51**	**2.31**	3.62	**3.15**

Table 2: A detailed breakdown of model performance (MSE) across distinct groups of linguistic attributes. Each group represents specific linguistic attributes that contribute to the overall complexity and structure of the generated text.

Model	Lexical	Syntactic	Discourse	Macro-MSE(l^t)
Ling-disc	0.08	0.14	0.50	0.24

Table 3: Pre-training test loss of the linguistic discriminator.

prediction logits according to an attribute classifier of the potential continuations.

- **QCPG** (Bandel et al., 2022), quality controlled paraphrase generation is a state-of-the-art model for controlled generation. Target attributes are discretized into tokens, and added as a prefix to the encoder input.
- **BOLT** (Liu et al., 2023b): a decoding-time algorithm for controlled text generation. For each test sample, it learns a set of biases by minimizing the losses of an attribute discriminator model and an LM's perplexity.
- **LLama3 (70B)** (Dubey et al., 2024): we use an instruction fine-tuned LLM to evaluate the ability of generative models to perform the controlled conversion.

Experimental Setup For each source and target sentence in our dataset, we extract the linguistic attributes from existing linguistic toolkits (Lu, 2020, 2012; Lee and Lee, 2023a). The attributes include lexical, syntactic, semantic, and discourse attributes, which capture a comprehensive spectrum of linguistic structures. The backbone generation model in all approaches is $f$lan-t5-base. Greedy decoding is used to better reveal each approach's merits. Detailed hyper-parameter settings are provided in Appendix D.

Evaluation We employ several evaluation metrics to assess models in control paraphrase generation. **BERTScore** (Zhang et al., 2020) evaluates the quality of generated text by measuring the similarity between the generation and the reference sentences, quantifying semantic fedelity. We also use the average mean squared error, **MSE(l^t)**, to compute the discrepancy between the linguistic attributes of generated paraphrases and their corresponding target attributes, which quantifies how accurately the model satisfies the target attributes. In addition, **MSE(l^s)** measures the difference between the linguistic attributes of generated paraphrases and those of their corresponding sources, which helps determine if a paraphrase sufficiently diverges from its source. An effective CPG model should ideally have a high BERTScore and low MSE(l^t), while maintaining a high MSE(l^t).

The **Overall** score of the model is computed as the average of **BERTScore**, $\mathrm{norm}_{[0,1]}$ **MSE(l^s)**, and $(1 - \mathrm{norm}_{[0,1]}$ **MSE(l^t)**$)$.

In addition, we introduce a new evaluation setting termed *Novel Target Challenge*, which tests models on generating paraphrases that adhere to target linguistic attributes associated with an "irrelevant" sentence to the source. It evaluates the model's adaptability to novel linguistic attributes and can act as a more robust test for CPG models. In our datasets, the average Euclidean distance between the linguistic attributes of source and true target sentences is 1.17, while the distance to the linguistic attributes of irrelevant sentences is 3.91. The novel target challenge is therefore a harder task.

4.1 Main Results

Table 1 shows the results obtained by all models across evaluation metrics. Reference has access to gold targets and Copy, Reference, and Vanilla T5 are baselines that lack mechanisms for controlling linguistic attributes.

Our first observation is that LINGCONV generates paraphrases that align more precisely with the desired linguistic attributes, as demonstrated by its lower MSE(l^t) compared to other competing baselines. This result can be attributed to directly integrating linguistic attributes with the decoder input through element-wise addition and the linguistic attribute predictor which effectively guides the decoder to generate paraphrases that adhere to the target linguistic attributes. QCPG shows similar MSE(l^t) performance but it employs a more indirect method for incorporating target attributes–by prefixing the input sequence with special discrete tokens. While effective, this approach may not provide the same level of precision in guiding the generation process. The discrete token prefixes could potentially introduce ambiguity or weaken the direct influence of linguistic attributes on the generated text.

Second, we observe that LINGCONV performs well in balancing attribute control, and semantic similarity of output, as shown by the overall score. The balance between attribute control and paraphrase faithfulness is a crucial aspect of high-quality controlled paraphrase generation. Specifically, within the novel target case LINGCONV achieves a substantial 34% decrease in attribute error compared to the best-performing baseline while maintaining the same fluency and semantic consistency as the gold reference paraphrases. Furthermore, in the novel target challenge, our quality control approach provides a significant reduction in MSE($\boldsymbol{l^t}$) of the linguistic attributes with minimal reduction in BertScore, providing a 14% further decrease in attribute error.

Third, the novel target case shows LingConv scores a significant increase in MSE(l^s) compared to the baseline models, with a difference of 2.95 points. The low value of MSE(l^s) indicates that baseline CPG methods are biased by the linguistic structure of the source sentence, and do not deviate far from it, while LingConv can restructure the input sentence to achieve the desired control attributes.

In addition, we find that BOLT has a limited capacity on fine-grained attribute control. In the novel targets case, BOLT achieves a 24% drop in error compared to T5-FT, which indicates that it moves in the correct direction. However, it still has a high MSE compared to other CPG methods, indicating that it struggles to control many attributes at once. On the other hand, Fudge, with a high enough λ_{Fudge}, has a guarantee to reduce the attribute error compared to T5-FT, because it samples the next token with the joint maximum LLM likelihood and minimum attribute error. However, Fudge has difficulty performing linguistic controls because it relies on long-scale dependencies of the text, where the generation needs to be based on sentence-level decisions rather than token-level.

We observe that LLama, although able to generate semantically similar paraphrases, has difficulty following instructions for attribute controls. In the standard case, this is evident by the MSE(l^t) higher than T5-FT, and in the novel target case we see that LLama slightly follows the attribute controls, achieving a poor error comparable to that of T5-FT.

4.2 Analysis of Linguistic Attributes

We analyze the performance of models across different groups of linguistic attributes to understand their strengths and weaknesses, and the inherent difficulty in controlling different types of attributes. We group the linguistic attributes into several types according to the categorizations in (Lu, 2020, 2012; Lee and Lee, 2023a). The attribute types are lexical, syntactic, and discourse features. We analyze MSE values for each model across standard and novel target scenarios, revealing the following insights:

4.2.1 Controlling Discourse Proves Most Challenging

Table 2 shows the error rate of each approach in controlling different linguistic attribute groups. Despite having the lowest average error across models, discourse attributes show the smallest reduction in error by LINGCONV compared to T5-FT, at 41%. This suggests that discourse attributes are the most challenging to control. In contrast, lexical attributes have the highest average baseline error, and

Augmentation	CoLA (Matthew's Corr.) Limited Data	CoLA (Matthew's Corr.) Full Data	RTE (Acc.) Limited Data	RTE (Acc.) Full Data	SST-2 (Acc.) Limited Data	SST-2 (Acc.) Full Data
No Aug.	53.8 ± 0.4	60.6 ± 1.0	68.4% ± 1.5	74.2% ± 1.5	91.3% ± 0.1	92.4% ± 0.3
Ineffective Aug.	52.5 ± 0.8	58.4 ± 1.1	66.1% ± 2.8	71.7% ± 2.6	91.0% ± 0.3	91.7% ± 0.1
Effective Aug.	**54.8** ± 0.6	**60.8** ± 1.1	**71.2%** ± 1.3	**76.0%** ± 0.8	**92.2%** ± 0.3	**93.0%** ± 0.4

Table 4: Performance on GLUE tasks with No, Effective and Ineffective augmentation. Effective and ineffective augmentations differ in the set of target linguistic attributes used to generate them.

LINGCONV achieves the most significant reduction in this error, at 74%. Syntactic attributes appear to be the easiest to control, with the error rate dropping from 8.82 to 2.31, a 73% reduction, the lowest among all groups. We note that Fudge achieves the lowest error in discourse attributes. This is because many of these attributes are represented by the presence and density of particular named entities. The generation of Fudge is driven by the next word that minimizes the MSE. Therefore, it can generate the singular named entities that significantly reduce the error. However, this is not an optimal strategy for syntactic structures that require several iterations of planning and building, as evidenced by the high error rate of Fudge on syntactic attributes.

Quality Control Boosts Adherence across Linguistic Attributes The quality control algorithm reduces the error rates of LINGCONV across all types of attributes. The largest improvement of 25% is in syntactic attributes. The algorithm of iterative refinement of a source sentence is particularly suited to the task of iteratively adding and deleting selected entities, and matching the required target more closely. The second largest improvement is in lexical attributes at 23%, the algorithm can iteratively add and delete selected words, matching the desired lexicon and minimizing the error in lexical attributes. Finally, discourse features often require a complete restructuring of the sentence, which is the most difficult. However, quality control achieves a 17% reduction in error. To further verify, we apply the quality control mechanism to T5-FT, instead of LINGCONV. T5-FT plus quality control has a 0.90 MSE(l^t) in the standard case and 9.20 in novel target case. In both scenarios, the model improved over the vanilla T5. However, it is evident from this results that quality control alone is not sufficient for attribute control, and the architecture of LINGCONV is essential.

Linguistic Predictor Performance The final MSE loss of the pre-trained linguistic predictor (LP) is 0.16 on our test set, indicating that the model's results have been achieved despite using imperfect linguistic predictor. This could potentially compound errors in the refined outputs generated during inference time with quality control mechanism. We further report the error of the linguistic discriminator over different types of attributes in Table 3. We find that the error rates are lowest for lexical attributes, moderately higher for syntactic attributes, and highest for discourse attributes. This finding is consistent with the literature on linguistic attributes (Pallotti et al., 2019; Rafatbakhsh and Ahmadi, 2023).

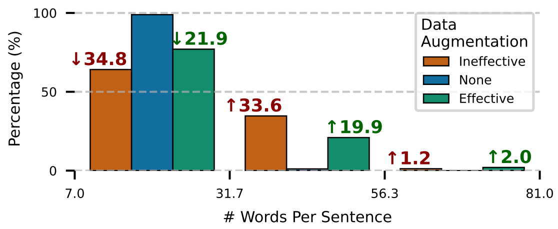


Figure 3: Attribute distributions for effective vs. ineffective augmentation on the RTE (Limited) dataset. Effective augmentation has a greater percentage of shorter sentences.

4.3 Paraphrase Generation for Augmentation

We study the use of LINGCONV in generating paraphrases for data augmentation, showing that controlling linguistic attributes is crucial.

We focus on three tasks from the GLEU benchmark (Wang et al., 2019a): Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019), Stanford Sentiment Treebank(SST-2) (Socher et al., 2013), and Recognizing Textual Entailment (RTE) (Dagan et al., 2005) with 8.5k and 1k, 67k and 1.8k, and 2.5k and 3k training and test samples respectively. Data augmentation is generally more effective for smaller datasets (Okimura et al., 2022; Louvan and Magnini, 2020). Therefore, we use Full and Limited versions of each dataset, with Limited containing reduced training data (10% for CoLA and SST-2, and 40% of RTE due to its smaller size). We use LINGCONV to generate paraphrases of the training samples, which are added back to the training set with labels matching the original samples. We create two sets of target attribute vectors by non-uniform sampling from the original data's linguistic attribute vectors ($\mathcal{T}$). Biased sampling aims to produce increased or decreased prevalence of particular attributes in the generated paraphrases for augmentation, compared to the original data. This approach allows us to identify which attribute values result in "Effective" vs. "Ineffective" augmentation based on task performance post-augmentation, compared to no augmentation. For example, we may sample data such that $p(l^t : l^t \in \mathcal{T}) = 0.9$ if $l^t_{[\text{TTR}]} > 0.8$ and $p(l^t : l^t \in \mathcal{T}) = 0.1$ otherwise, which results in substantial prevalence of high TTR values in the augmented samples.

We run experiments with DeBERTa$_{\text{base}}$ (He et al., 2021), using the same parameters as their GLUE benchmark experiments. Each experiment is run with six random seeds, and we report the mean and standard error. We identify "Effective" and "Ineffective" sets by first evaluating 20 randomly sampled sets. From these, we select two sets: one that shows a statistically significant performance increase and one that shows a significant decrease compared to no augmentation. We then compare the attribute distributions of these two sets to identify which attributes differ significantly. Results in Table 4 confirms that the distribution of the target attributes influence the effectiveness of data augmentation, see supplementary materials for our data.

Figure 3 visualizes attribute distributions that lead to effective and ineffective augmentation for the RTE (Limited) dataset. For effective augmentation, target attributes should have a significantly higher prevalence of shorter sentences, while ineffective augmentation produces more medium-length sentences. The Mann–Whitney U test confirms significant differences with p-value < 0.05 in the attribute distributions between effective and ineffective sets across all our six datasets. Details are provided in Appendix F.

5 Conclusion

We present a model for controllable text generation, offering control over 40 linguistic attributes and an effective mechanism for quality control at inference time, yielding a 12% improvement in output quality. We introduce the "Novel Target Challenge", where models generate paraphrases based on attributes from an "irrelevant" sentence. The setting effectively evaluates models' adaptability to novel linguistic attributes and can act as a more robust test for controlled paraphrase generation models.

We evaluate the model on the downstream application of generating synthetic data for data augmentation. Our model generates viable paraphrases that boost performance and produce data with targeted complexity levels, addressing biases in the original datasets.

References

Larry Armijo. 1966. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3.

Nischal Ashok Kumar, Nigel Fernandez, Zichao Wang, and Andrew Lan. 2023. Improving reading comprehension question generation with data augmentation and overgenerate-and-rank. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 247–259, Toronto, Canada. Association for Computational Linguistics.

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 596–609, Dublin, Ireland. Association for Computational Linguistics.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Stephen P Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984, Florence, Italy. Association for Computational Linguistics.

Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. 2020. A semantically consistent and syntactically variational encoder-decoder framework for paraphrase generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1186–1198, Barcelona, Spain (Online).

International Committee on Computational Linguistics.

Emilie Colin and Claire Gardent. 2018. Generating syntactic paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 937–943, Brussels, Belgium. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252, Online. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12590–12616, Toronto, Canada. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1022–1033, Online. Association for Computational Linguistics.

Kuan-Hao Huang, Varun Iyer, I-Hung Hsu, Anoop Kumar, Kai-Wei Chang, and Aram Galstyan. 2023a. ParaAMR: A large-scale syntactically diverse paraphrase dataset by AMR back-translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8047–8061, Toronto, Canada. Association for Computational Linguistics.

Kuan-Hao Huang, Varun Iyer, Anoop Kumar, Sriram Venkatapathy, Kai-Wei Chang, and Aram Galstyan. 2022. Unsupervised syntactically controlled paraphrase generation with Abstract Meaning Representations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1547–1554, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qiushi Huang, Yu Zhang, Tom Ko, Xubo Liu, Bo Wu, Wenwu Wang, and H Tang. 2023b. Personalized dialogue generation with persona-adaptive attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12916–12923.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018a. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018b. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy. Association for Computational Linguistics.

Amirhossein Kazemnejad, Mohammadreza Salehi, and Mahdieh Soleymani Baghshah. 2020. Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6010–6021, Online. Association for Computational Linguistics.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 14542–14554.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bruce W. Lee and Jason Lee. 2023a. LFTK: Handcrafted features in computational linguistics. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 1–19, Toronto, Canada. Association for Computational Linguistics.

Bruce W. Lee and Jason Lee. 2023b. Prompt-based learning for text readability assessment. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1819–1824, Dubrovnik, Croatia. Association for Computational Linguistics.

Zhe Lin, Yitao Cai, and Xiaojun Wan. 2021. Towards document-level paraphrase generation with sentence rewriting and reordering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1033–1044, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2023a. Composable text controls in latent space with ODEs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16543–16570, Singapore. Association for Computational Linguistics.

Jinxin Liu, Jiaxin Shi, Ji Qi, Lei Hou, Juanzi Li, and Qi Tian. 2022. ParaMac: A general unsupervised paraphrase generation framework leveraging semantic constraints and diversifying mechanisms. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6193–6206, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xin Liu, Muhammad Khalifa, and Lu Wang. 2023b. BOLT: Fast energy-based controlled text generation with tunable biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 186–200, Toronto, Canada. Association for Computational Linguistics.

Samuel Louvan and Bernardo Magnini. 2020. Simple is better! lightweight data augmentation for low resource slot filling and intent classification. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 167–177, Hanoi, Vietnam. Association for Computational Linguistics.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners' oral narratives. *The Modern Language Journal*, 96(2):190–208.

Xiaofei Lu. 2020. Automatic analysis of syntactic complexity in second language writing. *ArXiv preprint*, abs/2005.02013.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 90–98, Hong Kong. Association for Computational Linguistics.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. 2022. Mix and match: Learning-free controllable text generationusing energy language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 401–415, Dublin, Ireland. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2018. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:373–389.

Itsuki Okimura, Machel Reid, Makoto Kawano, and Yutaka Matsuo. 2022. On the impact of data augmentation on downstream performance in natural language processing. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 88–93, Dublin, Ireland. Association for Computational Linguistics.

Vishakh Padmakumar, Richard Yuanzhe Pang, He He, and Ankur P Parikh. 2023. Extrapolative controlled sequence generation via iterative refinement. In *International Conference on Machine Learning (ICML)*.

Gabriele Pallotti et al. 2019. An approach to assessing the linguistic difficulty of tasks. *Journal of the European Second Language Association*, 3(1):58–70.

E. Margaret Perkoff, Abhidip Bhattacharyya, Jon Cai, and Jie Cao. 2023. Comparing neural question generation architectures for reading comprehension. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 556–566, Toronto, Canada. Association for Computational Linguistics.

Lihua Qian, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Exploring diverse expressions for paraphrase generation. 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 3173–3182, Hong Kong, China. Association for Computational Linguistics.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551.

Elaheh Rafatbakhsh and Alireza Ahmadi. 2023. Predicting the difficulty of efl reading comprehension tests based on linguistic indices. *Asian-Pacific Journal of Second and Foreign Language Education*, 8(1):41.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating large language models on controlled generation tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Jan Philip Wahle, Bela Gipp, and Terry Ruas. 2023. Paraphrase types for generation and detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12148–12164, Singapore. Association for Computational Linguistics.

Thiemo Wambsganss, Andrew Caines, and Paula Buttery. 2022. ALEN app: Argumentative writing support to foster English language learning. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 134–140, Seattle, Washington. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019b. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11034–11044.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2021. Syntactically-informed unsupervised paraphrasing with non-parallel data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023. Tailor: A soft-prompt-based approach to attribute-based controlled text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–427, Toronto, Canada. Association for Computational Linguistics.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Haibo Zhang, Xue Zhao, Wenqing Yao, and Boxing Chen. 2022. GCPG: A general framework for controllable paraphrase generation. In *Findings of the Association for Computational Linguistics: ACL*

2022, pages 4035–4047, Dublin, Ireland. Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, Lingyun Xiang, Jin Wang, and Guoliang Ji. 2019. User-oriented paraphrase generation with keywords controlled network. *IEEE Access*, 7:80542–80551.

Hanqing Zhang and Dawei Song. 2022a. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hanqing Zhang and Dawei Song. 2022b. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Carolina Zheng, Claudia Shi, Keyon Vafa, Amir Feder, and David Blei. 2023. An invariant learning characterization of controlled text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3186–3206, Toronto, Canada. Association for Computational Linguistics.

A List of Linguistic Attributes

We use expert-crafted linguistic indices as the control attributes for CPG. Table 5 lists all the indices that we use. We select 40 out of 276 total indices in the three libraries. We select indices such that there are no duplicates, there is a representative index from each family, there is at least one index from each domain, the index is not too granular as to not be useful, and the selected included indices have utility in text style control.

For the full descriptions please refer to Lu (2020), Lu (2012), and Lee and Lee (2023a). The following is a brief description of a few indices as an example: **Automated Readability Index** is the grade level required for a reader to comprehend the text, from preschool to professor level. **Lexical words** are nouns, verbs, adjectives, and adverbs. **Sophisticated words** are the unconventional words. We consider the 2000 least frequent words in the American National Corpus as sophisticated. **Gpe Entity** is a geopolitical entity. **Norp entity** is nationalities or religious or political groups. **Age of acquisition** is the typical age at which a person learns and begins to use a particular word.

B Algorithm Background

This section describes further details on the STE and line search algorithms.

B.1 Straight-through Gradients

STE (Bengio et al., 2013) is a technique used to propagate gradients through non-differentiable equations in the computational graph, through an estimation of the derivative. In our case, the decoder produces token logits, which are then transformed into probabilities through softmax. Then, we transform the probabilities into an output sequence using argmax. LP takes as an input the sequence of tokens and not the sequence of logits. However, if we want to propagate the gradient of the loss generated by LP to the main model, we must pass the gradient through the output logits. Thus, we use the following trick to create a pathway in the computational graph from LP's inputs to the logits. First, the output sequence is represented in one-hot encoding rather than a sequence of tokens. Second, we add the logits to the one-hot encoding and subtract a detached (constant) variable equal to the logits. The end result would be equal to the one-hot encoding, but the computational graph now has a path from the logits to LP through the multiplication of the one-hot encoding with LP's text embedding. This means that the gradient propagated to each token of the logits is scaled according to the weights of the text embedding matrix.

B.2 Line Search

Line search (Armijo, 1966) is a standard numerical optimization algorithm, where at every update step, the step size is chosen dynamically. There are different methods of finding the best step size. They often include trying out many different step sizes, evaluating the resulting parameters, and choosing the step size that results in the lowest loss value.

Our algorithm is based on backtracking line search, which starts with a large candidate step size, and if it doesn't result in a lower loss than the current, reduce it by a factor of γ (often $= 0.5$) and try again. The intuition is that we would like to take the largest step possible that results in an improvement to descend toward the global minimum and potentially avoid local minima. However, we would like the opposite; we would like to take the smallest possible step that results in an improvement to not deviate away from the original sentence semantics. Therefore, our algorithm starts from a small step size and grows it by a factor of γ at each line search step.

C Datasets

We combine The Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), The Semantic Textual Similarity Benchmark (STS-B) (Cer et al., 2017), and The Quora Question Pairs. The three datasets are created for the task of classifying whether the pair of texts are semantically equivalent. Therefore, we only select the positive samples for our model's training and discard the remaining samples. The data distribution is shown in Table 6.

The dataset is randomly split into training, validation, and testing sets according to the ratio 80:10:10. The same data is used for training all versions of our approach and baselines. The semantic equivalence and linguistic predictor models are both pre-trained using the same data and splits.

D Experimental Settings

We train our model using a single A100 GPU with a batch size of 40, and a learning rate of $1e-3$ Adam optimizer. We optimize the hyper-parameters of FUDGE and QCPG. In QCPG, optimized batch size $= 8$, learning rate $= 1e-4$, and we train for a large number of epochs $= 20$ to ensure high performance. In FUDGE, we optimize the update factor and the multiplicative factor $\lambda_{FUDGE} = 0.7$. We use the linguistic predictor described in § 3.2 as an attribute classifier for FUDGE, and weigh the logits according to the inverse of the mean squared error of the prediction's linguistic attributes and the target linguistic attributes. Although FUDGE benefits from not having to train or fine-tune the language model, it is extremely slow at inference time due to the demand of evaluating numerous candidates at each generation step. The parameters for the Algorithm 1 are: $\eta_0 = 10^3, \gamma = 2.25, \tau = 0.95, k = 4$ All models are run with 1 seed. The random seed used for all data processing and models is 0. When $k > 1$ random seeds are used, such as in section 4.3, seeds are from 0 to $k - 1$.

The three augmentation settings are trained for 2 epochs, and the best checkpoint is used. We use a learning rate of $1e - 3$, batch size of 40, and linear learning rate scheduling.

Linguistic attributes are quantized using the KBinsDiscretizer[5] with the "kmeans" clustering strategy.

E Attribute-specific Performance

Table 7 shows the error rate of each approach with respect to individual attributes. The errors are reported in mean absolute error (MAE).

LingConv achieves the least error in 5 out of 6 of the listed indices. LLama shows the worst performance compared to CPG methods. Compared to the T5-FT baseline, BOLT and Fudge only slightly improve the error. QCPG is the best-performing baseline after LingConv. Notably, QCPG shows the smallest error in controlling the number of nouns in a sentence. Moreover, QCPG controls the readability index of the generation with an MAE of 3 and the ratio of unique words in a sentence with an error of 6%. For both of these indices, LingConv still achieves the smallest error.

LingConv controls the number of words up to an error of 3 words, which is the best among all baselines. LingConv also significantly improves upon the control of word sophistication in the sentence, with an MAE of 2 words. Finally, LingConv can control the reading level of a sentence from Kindergarten (1) to Professor (14) level with an MAE of 3, which is non-trivial given that non-control baselines have an MAE of 6 levels, and LLama has an MAE of 8 levels.

F Distributions of Augmentation Attributes

Figures 4-8 show the distributions of the biased attributes in the strong and weak sets of target linguistic variables.

Figure 4 shows that for the CoLA (Limited) dataset, effective augmentation is correlated with an increased percentage of sentences where the ratio of unique verbs exceeds 0.7. This suggests

[5] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html

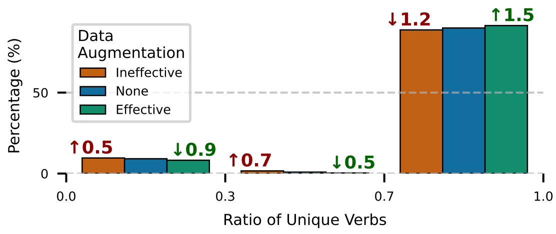
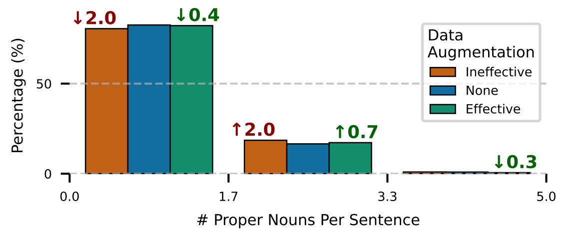
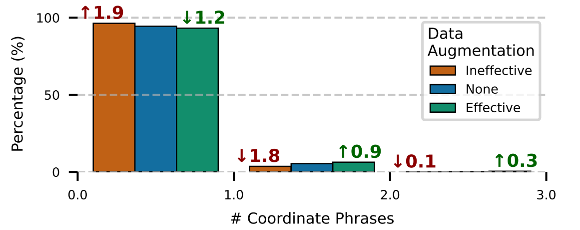


Figure 4: For CoLA (Limited), effective augmentation is associated with increased percentage of sentences with ratio of unique verbs > 0.7.



(a) Increase the percentage of sentences with a smaller number of proper nouns.



(b) Increase the number of sentences with more than 1 corodinate phrase.

Figure 5: For CoLA (Full), we bias two attributes.

that sentences with a higher diversity of verbs contribute to more effective augmentation, likely by enhancing the semantic richness of the generated data.

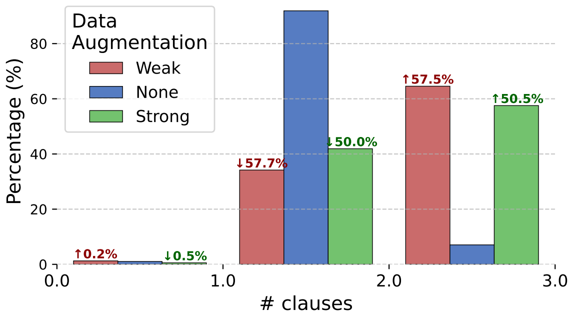
Figure 5 presents results for the CoLA (Full) dataset with two distinct attribute biases. On the left, we see that increasing the percentage of sentences with fewer proper nouns is associated with effective augmentation. This indicates that simpler sentences with fewer proper nouns may improve performance. On the right, the data shows that increasing the number of sentences containing more than one coordinate phrase also leads to effective augmentation. This suggests that complex sentence structures with multiple coordinate phrases contribute positively to augmentation effectiveness.

Figure 6 details biases applied to the RTE (Full) dataset. The left subplot indicates that effective augmentation is linked to a higher percentage of sentences with more than three clauses. This suggests that sentences with more complex structures are beneficial for augmentation. Conversely, the right subplot shows that decreasing the percentage of sentences with a Type-Token Ratio (TTR) greater than 0.8 is associated with effective augmentation. This implies that sentences with a lower TTR, reflecting less lexical variety, can also enhance augmentation effectiveness.

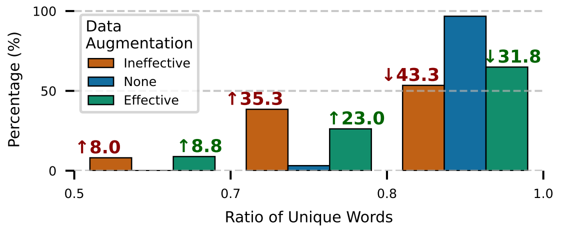
Figure 7 demonstrates the impact of reducing the ratio of sophisticated words in the SST-2 (Limited) dataset. Effective augmentation is associated with a decrease in sophisticated words, suggesting that simpler vocabulary contributes to better augmentation outcomes in this dataset.

Figure 8 provides a detailed view of biased attributes for the SST-2 (Full) dataset. The top-left subplot shows that increasing the number of unique lexical words leads to effective augmentation. The top-right subplot reveals that increasing the average sentence length is also beneficial. Additionally, the bottom subplot indicates that a higher number of sentences with more than nine lexical words contributes to effective augmentation. These results suggest that a richer vocabulary and longer sentences improve augmentation effectiveness.

These figures collectively illustrate how manipulating various linguistic attributes influences the effectiveness of data augmentation, highlighting specific features that can be optimized to enhance performance across different datasets.



(a) Increase the percentage of sentences with more than 3 clauses.



(b) Decrease the percentage of sentences with TTR > 0.8.

Figure 6: For RTE (Full), we bias two attributes.

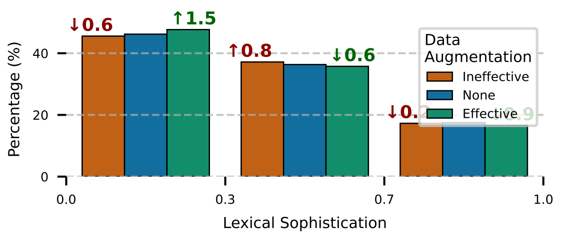


Figure 7: For SST-2 (Limited), decrease the ratio of sophisticated words.

Algorithm 1 Quality Control

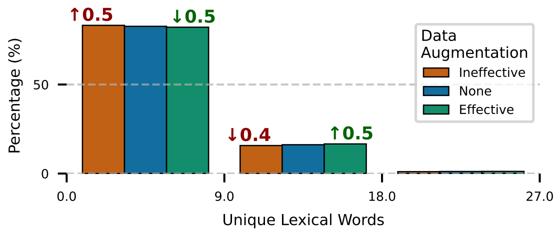
This algorithm optimizes the alignment of generated text with target linguistic attributes while preserving semantic equivalence to the source. The quality control loop adjusts the text embeddings iteratively using a gradient-based method combined with a line search to minimize attribute errors. The process continues until a satisfactory generation is found or the algorithm exhausts its search.

Require: model M, linguistic predictor LP, semantic classifier SE, input s, target attributes l^t, base step size η_0, step size scaling factor γ, semantic equivalence threshold τ, patience k

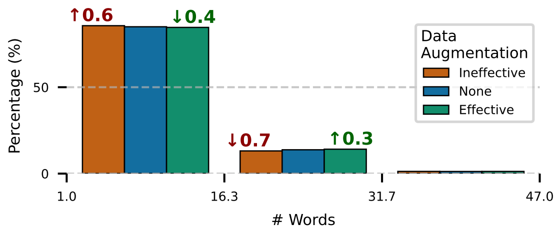
procedure QUALITY_CONTROL(s, l^t)
  $\Theta \leftarrow Emb(s)$ ▷ Initialize embeddings from the source text
  while True **do**
    $\hat{t} \leftarrow M(\Theta, l^t)$ ▷ Generate text with current embeddings
    $l_{\text{current}} \leftarrow \|LP(\hat{t}) - l^t\|_2^2$ ▷ Compute attribute error
    $g \leftarrow \nabla_\Theta l_0$ ▷ Compute gradient w.r.t. embeddings
    $\Theta \leftarrow$ ADAPTIVE_STEP_SEARCH(Θ, l_0)
    if $\Theta = null$ **then**
      break ▷ Terminate if no improvement is found
  return $\hat{t}$
procedure ADAPTIVE_STEP_SEARCH(Θ, l_0)
  $\eta \leftarrow \eta_0$ ▷ Initialize step size
  patience $\leftarrow k$ ▷ Initialize patience counter
  while patience > 0 **do**
    $\sigma_{\text{sem}} \leftarrow SE(s, \hat{t}')$ ▷ Check semantic equivalence
    if $l' < l_0$ and $\sigma_{\text{sem}} \geq \tau$ **then**
      return Θ' ▷ Accept and return the new embeddings
    else
      $\eta \leftarrow \eta * \gamma$ ▷ Reduce step size
      patience $\leftarrow$ patience $- 1$ ▷ Decrease patience
  while patience > 0 **do**
    $\Theta' \leftarrow \Theta - \eta * g$ ▷ Update embeddings
    $\hat{t}' \leftarrow M(\Theta', l^t)$ ▷ Generate text
    $l' \leftarrow \|LP(\hat{t}') - l^t\|_2^2$ ▷ Compute new attribute error
    $\sigma_{\text{sem}} \leftarrow SE(s, \hat{t}')$ ▷ Check semantic equivalence
    if $l' < l_0$ and $\sigma_{\text{sem}} \geq \tau$ **then**
      return Θ' ▷ Accept and return the new embeddings
    else
      $\eta \leftarrow \eta * \gamma$ ▷ Reduce step size
      patience $\leftarrow$ patience $- 1$ ▷ Decrease patience
  return $null$ ▷ Return null if no improvement

Unique sophisticated words
Unique lexical words
Unique sophisticated lexical words
Total words
Total sophisticated words
Lexical sophistication (unique)
Verb sophistication
Ratio of unique words
Ratio of unique verbs
Ratio of unique adjectives
Ratio of unique adverbs
Dependent clauses
Clauses
T-units
Complex T-units
Complex nominals
Stop Words
Sentences
Characters
Average Words Per Sentence
Average Characters Per Sentence
Average Characters Per Word
Average Syllables Per Sentence
Total Age Of Acquistion Of Words
Named Entities Norp
Named Entities Gpe
Named Entities Law
Named Entities Money
Named Entities Ordinal
Coordinating Conjunctions
Nouns
Numerals
Proper Nouns
Subordinating Conjunctions
Automated Readability Index
Reading Time For Average Readers

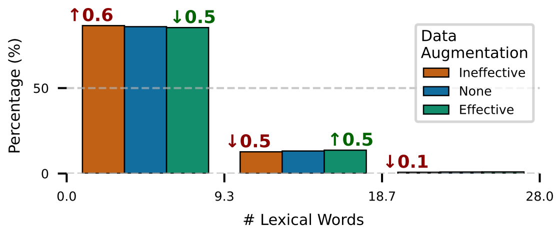
Table 5: Linguistic indices used in this paper.



(a) Increase number of unique lexical words.



(b) Increased average sentence length.



(c) Increase sentences with # Lexical Words > 9

Figure 8: For SST-2 (Full), we bias the number of lexical words, total words, and unique lexical words.

Dataset	Full Dataset	Positive Samples
QQP	363,846	134,378
MRPC	3,668	2,474
STS-B	5,749	2,994
Total	373,263	139,846

Table 6: QQP, MRPC, and STS-B contain samples that are either semantically equivalent or not equivalent. We select from the three datasets samples with the *equivalent* label for training and evaluating our model.

	# words	# sophisticated words	# lexical words	ratio of unique words	# nouns	readability index
ref	12.97	4.29	7.60	9.13%	2.16	6.62
copy	12.98	4.29	7.61	9.25%	2.14	6.65
t5-ft	12.83	4.22	7.49	9.18%	2.10	6.69
llama	12.04	4.55	7.25	8.29%	2.36	8.01
bolt	10.85	3.36	6.11	8.51%	1.83	5.47
fudge	11.10	3.36	6.29	7.95%	2.00	5.09
qcpg	5.34	2.83	3.62	5.93%	**1.16**	3.04
lingconv	4.37	2.38	3.04	**5.92**%	1.27	3.36
lingconv+qc	**3.21**	**1.97**	**2.36**	6.38%	1.23	**3.01**

Table 7: a detailed breakdown of model performance across a selected set of linguistic attributes. performance is reported in mean absolute error (mae). the results are based on novel targets of linguistic attributes.