Linguistically-Controlled Paraphrase Generation

Mohamed Elgaar and Hadi Amiri
University of Massachusetts Lowell
{melgaar,hadi}@cs.uml.edu

Abstract

Controlled paraphrase generation produces paraphrases that preserve meaning while allowing precise control over linguistic attributes of the output. We introduce LingConv, an encoder-decoder framework that enables fine-grained control over 40 linguistic attributes in English. To improve reliability, we introduce a novel inference-time quality control mechanism that iteratively refines attribute embeddings to generate paraphrases that closely match target attributes without sacrificing semantic fidelity. LingConv reduces attribute error by up to 34% over existing models, with the quality control mechanism contributing an additional 14% improvement.[1]

1 Introduction

Controllable text generation (CTG) aims to produce text with specified linguistic attributes (Ficler and Goldberg, 2017; Jin et al., 2022). A sub-task, controlled paraphrase generation (CPG), aims to generate paraphrases that satisfy desired attributes while preserving meaning. CPG has applications in text simplification (Lee and Lee, 2023b; Zhang and Lapata, 2017), toxicity control (Zheng et al., 2023), data augmentation (Iyyer et al., 2018a), and creating linguistically challenging data (Perkoff et al., 2023). The key challenge is to balance attribute adherence with semantic fidelity.

Prior work in CPG typically controls a small number of attributes, often less than three (Bandel et al., 2022; Liu et al., 2023b; Yang et al., 2023). Large language models (LLMs) such as Llama (Dubey et al., 2024), while powerful, struggle with precise and simultaneous control over many attributes via prompting (Dekoninck et al., 2024). In addition, decoding-time methods that use attribute classifiers can be slow and less effective in high-dimensional attribute spaces (Yang and Klein, 2021; Liu et al., 2023b), and inference-time quality control is rarely addressed.

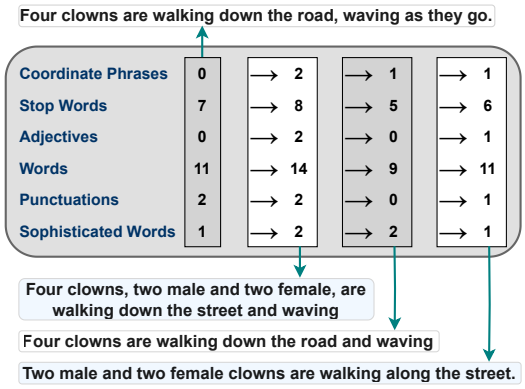


Figure 1: We aim to transform a given sentence into multiple paraphrases, each satisfying distinct linguistic attributes. Our model takes a source sentence and a set of target linguistic attributes and generates a paraphrase optimized to satisfy the target attributes. Here we show three paraphrases with different linguistic attributes generated for the source sentence. Linguistic features identified using the spaCy "en_core_web_sm", with stop-word list from Explosion AI (2025).

CPG can generate linguistically challenging data[2] (Perkoff et al., 2023; Ashok Kumar et al., 2023; Wambsganss et al., 2022), augment datasets (Iyyer et al., 2018a; Malandrakis et al., 2019), and support language simplification (Lin et al., 2021). The main challenge is to generate text that preserves meaning and satisfies target attributes. Most prior work focuses on a limited set of attributes. However, broader attribute control increases flexibility for diverse audiences.

We introduce LingConv, a novel encoder-decoder CPG model that offers fine-grained control over 40 linguistic attributes spanning lexical, syntactic, discourse, and semantic aspects (see Ap-

[1] Our code and an interactive demo (Elgaar and Amiri, 2025) are available at https://github.com/CLU-UML/LingConv.

[2] 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.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 20842–20864
November 4-9, 2025

pendix B). LINGCONV integrates attribute embeddings directly into the decoder and employs a robust inference-time quality control (QC) mechanism. This QC mechanism iteratively refines outputs using linguistic attribute and semantic consistency classifiers, guided by a line-search algorithm to ensure close alignment with target attributes without sacrificing meaning. Figure 1 shows an example of our model's capability.

Our contributions are:

- the first system, to our knowledge, for CPG with simultaneous control over 40 fine-grained linguistic attributes;
- a novel inference-time quality control mechanism that significantly improves attribute adherence; and
- application to data augmentation, generating attribute-controlled synthetic data to improve downstream task performance.

We demonstrate through extensive experiments that LINGCONV outperforms strong baselines by up to 34% in attribute control, with the QC mechanism providing a further 14% improvement.

CPG has the potential to generate data that challenges existing models from a linguistic perspective, produce text with varying levels of linguistic complexity for language learners (Okano et al., 2023; Perkoff et al., 2023; Uto 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, as discussed below. Accommodating a wider array of linguistic attributes in CPG is crucial because it improves the flexibility and engagement for diverse audiences including language learners.

LINGCONV is an encoder-decoder CPG model that offers fine-grained control over 40 linguistic attributes spanning lexical, syntactic, discourse, and semantic aspects (see Appendix B). It integrates attribute embeddings directly into the decoder and employs a novel inference-time quality control (QC) mechanism that iteratively refines outputs using linguistic attribute and semantic consistency classifiers to ensure close alignment with target attributes without sacrificing meaning.

Extensive experiments demonstrate that LINGCONV outperforms strong baselines by up to 34% in attribute control, with the QC mechanism providing a further 14% improvement. We also show the utility of our approach in data augmentation, where attribute-controlled synthetic data can be tailored to improve downstream task performance. Analysis reveals which linguistic attributes are easier or harder to control and the factors influencing controllability.

2 Related Work

Controllable text generation (CTG) and controlled paraphrase generation (CPG) have seen significant advances, with early works focusing on controlling a small set of attributes such as formality (Ficler and Goldberg, 2017; Dathathri et al., 2020; Yang and Klein, 2021). Most prior CPG approaches are limited to manipulating up to three attributes simultaneously (Bandel et al., 2022; Liu et al., 2023b; Yang et al., 2023), often relying on discrete control tokens or prompt-based strategies, which can be imprecise and lack fine-grained control. Decoding-time control methods using attribute classifiers (Yang and Klein, 2021; Liu et al., 2023b) are typically slow and struggle with high-dimensional attribute spaces, and quality control at inference time is rarely addressed. LLMs such as Llama (Dubey et al., 2024) and T5 (Raffel et al., 2020) demonstrate strong general-purpose generation, but prompt-based control remains coarse and unreliable for fine-grained attribute manipulation.

Colin and Gardent (2018) show that including a textual syntactic constraint in paraphrase generation produces syntactically diverse outputs. Other approaches have explored keyword exclusion (Kajiwara, 2019), using discriminator networks to enforce diversity (Qian et al., 2019), and following exemplar syntax (Chen et al., 2019). FSET (Kazemnejad et al., 2020) improves quality and diversity by retrieving similar paraphrase pairs and applying their edits to the source sentence. variational autoencoders (VAEs) were used to disentangle semantic and syntactic representations to generate diverse paraphrases (Chen et al., 2020; Yang et al., 2021). GCPG (Yang et al., 2022) concatenates conditions to the input to control keywords and syntax. Shi and Wu (2024) introduced action-controlled paraphrasing using action tokens, though this does not directly control specific linguistic attributes.

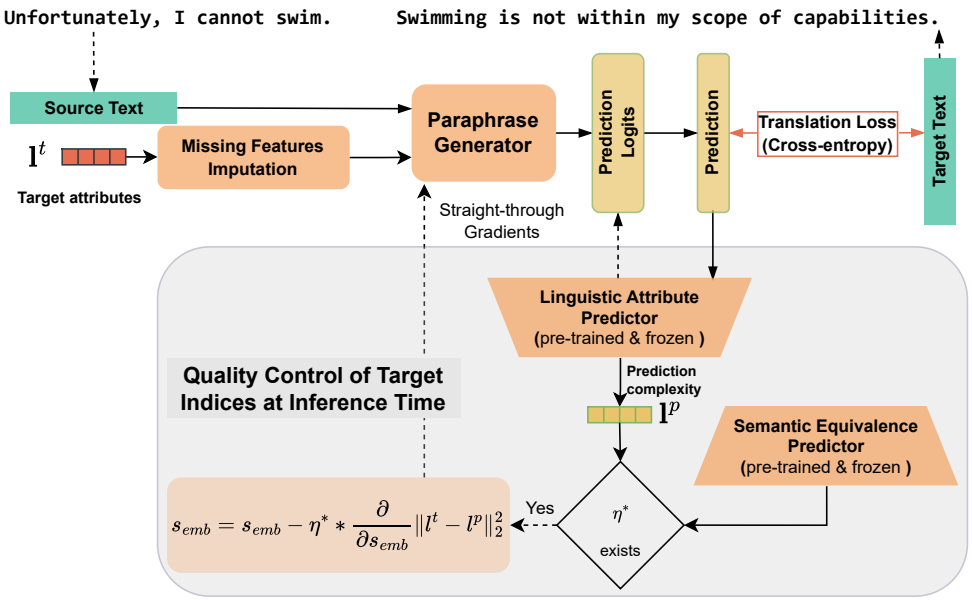


Figure 2: LINGCONV Architecture: The paraphrase generator extends the T5 model by incorporating linguistic attributes into the decoder inputs. Linguistic attributes of the source (l^s) and target (l^t) are embedded and fused with the generation using element-wise addition to the decoder inputs. In addition, 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 Semantic Equivalence Predictor (SE) receives as input the source sentence and the candidate generation $\hat{t}$, as in Algorithm 1 (line 25), to assess semantic similarity. The model is trained with a dual objective of semantic equivalence and linguistic attribute adherence.

Alternative approaches to CTG include energy-based models that sample from a latent space (Kumar et al., 2021; Wang et al., 2019; Gu et al., 2023; Liu et al., 2023a). Gradient-based methods like PPLM (Dathathri et al., 2020) steer generation using an attribute classifier at inference time but are often slow. FUDGE (Yang and Klein, 2021) improves efficiency by re-weighting the next-token probability based on the desired attribute.

Existing work has focused on specific types of control. QCPG (Bandel et al., 2022) controls for three abstract attributes (semantic similarity, syntactic variation, lexical variation), while others focus on keyword presence (Zeng et al., 2019; Liu et al., 2023b). Syntactically-controlled paraphrase generation has been explored by manipulating abstract meaning representation (AMR) trees (Huang et al., 2023), reordering parse tree segments (Goyal and Durrett, 2020), or using constituency parse templates (Iyyer et al., 2018b). Other methods disentangle semantics and syntax by adding sentence parse trees or AMR trees as features (Huang and Chang, 2021; Huang et al., 2022).

In summary, previous works have primarily focused on a narrow set of linguistic attributes and often lack robust quality control mechanisms at inference time. In addition, LLMs are powerful general-purpose generation, but achieving fine-grained, multi-attribute control is still a major challenge. Our work addresses these gaps by introducing a model capable of controlling a large, diverse set of linguistic attributes simultaneously, complemented by a novel inference-time quality control mechanism to ensure both attribute adherence and semantic fidelity.

3 LingConv

3.1 Problem Formulation

Consider a 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 target's linguistic attributes ($l^t \in \mathbb{R}^k$). The task is to generate t, given s and l^t.

3.2 LingConv Architecture

Overview LINGCONV is a seq2seq model with three main components (Fig. 2): an encoder-decoder paraphrase generator, a linguistic attribute predictor, and a quality control (QC) module. The attribute predictor and a semantic equivalence classifier are pre-trained and used only during inference

for QC. Input attribute vector may specify any subset of attributes; missing values are allowed and they are imputed via MICE (Azur et al., 2011). The encoder-decoder integrates attribute embeddings into the generation process, and the QC module iteratively refines outputs to match target attributes. The model is trained to generate paraphrases conditioned on source and target attributes.

Encoder-Decoder The encoder-decoder extends T5 (Raffel et al., 2020), embedding the target attribute vector l^t and adding it to the first decoder input token. Decoder-side injection provides direct, precise control over generation (see Appendix C). This approach balances semantic preservation and attribute control, and allows users to specify only selected attributes, with the remainder imputed from training data patterns.

Specifically, to guide the model toward generating desired outputs, we embed the linguistic attributes l^t into a dense vector representation and integrate them with T5's decoder inputs. While incorporating attributes via input modifications is common in controlled generation, our architecture achieves a balanced trade-off between semantic preservation and attribute control by performing decoder-side injection.

Our architecture effectively balances semantic preservation and attribute control by injecting attributes at the decoder side, allowing direct influence on token generation through self-attention while maintaining access to the full source context through cross-attention. First-token injection strikes an optimal balance between providing a strong control signal and minimizing disruption to the pre-trained model's capabilities.

To address practical concerns regarding the specification of all linguistic attributes, our approach can utilize variable imputation. This allows users to specify only the variables of interest, while the model fills in the rest based on learned patterns from the training data.

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. To achieve this goal, we add the embedding of the target linguistic vector $\mathbf{l}^t$ to the first token of the decoder inputs, which corresponds to the beginning of sentence token <bos>:

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

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 k is the number of linguistic attributes and d is the dimension of text input embeddings. The input attributes are standardized to a mean of 0 and a variance of 1 prior to embedding.

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

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

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. While existing linguistic tools can extract attributes, they are not differentiable and would require reinforcement learning approaches for optimization. Moreover, it helps us avoid the computationally intensive task of calculating 40 linguistic attributes for each generated text within the training process. We implement the linguistic predictor (LP) using a T5 encoder followed by a projection layer, and it is pre-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:

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

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 E.1. Additionally, the LP enables baseline methods like BOLT (Liu et al., 2023b) and FUDGE (Yang and Klein, 2021) that require differentiable attribute scoring for decoding-time control.

During inference, this pre-trained LP is used within the QC mechanism to evaluate how well the generated text $\hat{t}$ matches the target attribute vector l^t. Specifically, LP computes the attribute error $\|\text{LP}(\hat{t}) - l^t\|_2^2$, which is then used to guide the iterative refinement of the generated output, as detailed in Algorithm 1.

Imputation of Missing Values Manually specifying all 40 linguistic attributes for a target paraphrase is impractical and prone to error, as users may not know desirable values for every attribute or may specify inconsistent combinations. To address this real-world challenge, we employ the Multiple Imputation by Chained Equations (MICE) algorithm (Azur et al., 2011). This allows users to provide values for only a small subset of attributes they wish to control. The model then imputes the remaining values by leveraging statistical relationships learned from the training data. This feature significantly improves usability and reduces the risk of misconfiguration. See Appendix L for details.

Semantic Equivalence Classifier (SE) quantifies semantic equivalence of a pair of sentences, and is used in the quality control algorithm. The SE module receives as input the source sentence and the candidate generation $\hat{t}$ to compute semantic equivalence. We implement SE using a T5 encoder followed by a projection layer. This design ensures architectural compatibility and efficient integration with our T5-based LingConv generation model, and allows us to pre-train SE using a contrastive loss function specifically tailored to our paraphrase data. Notably, the contrastive loss described below was used exclusively for pre-training SE; it was not explored during the main LingConv model training. During inference, SE serves solely as a fixed, pre-trained module for semantic equivalence assessment, without further updates or integration of the contrastive loss into the primary model's training objective. SE is pre-trained by minimizing the following contrastive loss:

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

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. Achieving precise control over multiple linguistic attributes while maintaining text quality presents significant challenges in controlled text generation. Our approach employs an adaptive, gradient-based iterative refinement process (Padmakumar et al., 2023) that dynamically adjusts the model's input embeddings to steer outputs toward the target attributes. To ensure both attribute control and semantic fidelity, we use a line search algorithm (Armijo, 1966; Boyd and Vandenberghe, 2004) to select the optimal update strength at each step. This mechanism enables robust, fine-grained control over linguistic properties during inference. For a detailed description of the algorithm, see Appendix D.

4 Experiments

We evaluate LingConv on MRPC, STS-B, and QQP, using BERTScore and mean squared error (MSE) of attribute adherence (see Appendix F).

4.1 Experimental Setup

For each source and target sentence in our dataset, we extract the 40 linguistic attributes (listed in Appendix B) from existing linguistic toolkits (Lu, 2020, 2012; Lee and Lee, 2023a; Elgaar and Amiri, 2023). The attributes include lexical, syntactic, semantic, and discourse attributes, which capture a comprehensive spectrum of linguistic structures. We use `flan-t5-base` (Chung et al., 2024) as a base model, and re-implement all baselines to use the same base model for fairness. We use greedy decoding for all models.

Furthermore, we compare against: Copy (input as output), Reference (gold paraphrase), T5-FT (fine-tuned T5), FUDGE (Yang and Klein, 2021),

Model	BERTS↑	MSE(l^t)↓	MSE(l^s)↑	Overall↑	Novel Target Challenge BERTS^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	94.24	0.96 ± 0.03	0.51 ± 0.04	0.48 ± 0.01	96.65	9.00 ± 0.78	0.68 ± 0.02	0.32 ± 0.03
Llama	91.03	2.24 ± 0.08	**1.86** ± 0.07	0.38 ± 0.01	**92.77**	8.71 ± 0.49	2.47 ± 0.26	0.27 ± 0.03
BOLT	90.64	1.12 ± 0.03	1.10 ± 0.05	0.44 ± 0.01	90.38	7.34 ± 0.66	1.84 ± 0.06	0.29 ± 0.04
FUDGE	92.01	0.85 ± 0.01	1.05 ± 0.05	0.48 ± 0.01	92.53	6.94 ± 0.78	2.92 ± 0.57	0.32 ± 0.05
QCPG	**95.36**	0.63 ± 0.02	0.82 ± 0.05	0.54 ± 0.01	91.36	5.54 ± 0.55	3.14 ± 0.06	0.36 ± 0.04
Lingconv	95.15	0.62 ± 0.04	0.80 ± 0.06	0.55 ± 0.01	92.04	3.92 ± 0.32	4.20 ± 0.30	0.41 ± 0.03
+QC	95.17	**0.56** ± 0.04	0.77 ± 0.05	**0.56** ± 0.01	91.54	**3.07** ± 0.29	**5.92** ± 0.37	**0.44** ± 0.04

Table 1: 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)↑). Lower MSE(l^t) indicates better attribute control; higher MSE(l^s) indicates greater deviation from the source. Results are averaged over three seeds; standard error is shown for all metrics except BERTScore, where it is always less than 0.01.

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

Table 2: Pre-training test loss of the linguistic discriminator.

QCPG (Bandel et al., 2022), BOLT (Liu et al., 2023b), and Llama (Dubey et al., 2024). See Appendix H for a detailed description of each baseline.

4.2 Evaluation

Our evaluation is designed to assess both semantic fidelity and fine-grained linguistic control in our controlled paraphrase generation system. In the standard evaluation setting, where target reference paraphrases are available, we adopt **BERTScore** (Zhang et al., 2020) to measure semantic similarity between the generated paraphrase and its corresponding reference. BERTScore leverages contextualized embeddings to capture deep semantic correspondences that go beyond surface-level n-gram overlap, making it particularly effective in scenarios with substantial linguistic reformulation.

To quantify the model's ability to adhere to target linguistic attributes, we measure the mean squared error between the generated paraphrase's linguistic attributes and the target attributes, denoted as **MSE(l^t)**. Lower values of MSE(l^t) indicate that the generated paraphrase closely follows the desired attribute controls. Furthermore, we compute **MSE(l^s)** to assess the divergence of the paraphrase from its source text, ensuring that the output not only preserves the intended semantic content but also exhibits the required linguistic modifications.

A lower MSE(l^t) indicates better attribute control, a higher MSE(l^s) is actually desirable in many cases, as it reflects the model's ability to produce significant linguistic transformations from the source. This is particularly important in the Novel Target Challenge, where successful models must demonstrate the capacity to significantly restructure inputs according to target attributes that differ substantially from the source text's attributes. A model that simply copies the source (or makes minimal changes) would have a low MSE(l^s), indicating insufficient attribute transformation.

To provide a concise summary of performance across these dimensions, we define an **Overall** score computed as the average of three normalized metrics, each scaled to lie between 0 and 1: BERTScore, the normalized MSE(l^s), and $(1 - \text{normalized MSE}(l^t))$. This Overall score captures our dual objectives of preserving semantic fidelity and effective attribute control.

In addition to standard evaluation, we introduce the *Novel Target Challenge*, a more demanding setting in which models generate paraphrases based on target linguistic attributes derived from an "irrelevant" sentence relative to the source. An "irrelevant" sentence is one randomly sampled from the test set, with no guaranteed semantic or topic relation to the source. This creates a robust test of a model's ability to generate diverse paraphrases, independent of the source's linguistic structure. Since no gold reference is available in this scenario, we employ a reference-free variant of BERTScore (Shen et al., 2022), denoted by **BERTScore**^{F}. Reference-free BERTScore computes semantic similarity directly with respect to the source text instead of a gold standard reference, thereby providing a robust assessment when the target attributes are decoupled from conventional reference paraphrases. This is crucial for testing model adaptability in real-world

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

applications where specified target attributes may be entirely novel relative to the source.

Alternative metrics such as iBLEU (Liu et al., 2020; Niu et al., 2021) have been proposed for paraphrase evaluation to balance semantic similarity with lexical diversity by penalizing excessive overlap with the source. However, these metrics focus largely on surface-level comparisons. In contrast, our evaluation framework, which combines BERTScore (or BERTScore^F in the novel target setting) with MSE metrics for target and source linguistic attributes, directly quantifies both semantic preservation and the degree to which controlled linguistic attributes are followed, regardless of whether they are similar or different from those of the source. This approach is more aligned with linguistically controlled paraphrase generation.

Detailed analysis of linguistic attribute control and a full description of our paraphrase generation for data augmentation are presented in Appendix J and Appendix 5.5.

5 Results

Table 1 shows the results obtained by all models across evaluation metrics.

5.1 Attribute Control vs. Semantic Fidelity

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 comparable 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.

5.2 Trade-offs in the Novel Target Challenge

In the Novel Target Challenge, LINGCONV sometimes achieves a slightly lower BERTScore compared to baselines. This can be explained as a trade-off inherent to the task: LINGCONV is designed to prioritize adherence to the specified (and often difficult or even conflicting) novel attribute targets, as evidenced by its much lower MSE(l^t). Achieving these attribute targets may require the model to make more substantial changes to the source sentence, which can result in greater semantic deviation from the original text. In contrast, models that are less effective at precise attribute control (and thus have higher MSE(l^t)) tend to produce outputs that remain closer to the source, thereby achieving a higher BERTScore, at the cost of failing to achieve the requested attribute modifications.

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. Detailed per-dataset results are available in Appendix Table 8, showing that our approach consistently outperforms baselines across all three datasets.

5.3 Analysis of Baseline Methods

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.

5.4 Comparison with Large Language Models

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.

Our model achieves a 34% error reduction over LLama in attribute control. While large models like LLama-70B excel at general-purpose generation, our results show they struggle with precise attribute control (MSE(l^t) of 8.90 vs LingConv's 3.69 in novel target scenarios). This highlights a fundamental limitation of prompt-based approaches—even with detailed instructions, LLMs lack the specialized architecture and optimization procedures needed for fine-grained attribute matching. The quality control mechanism provides an additional 14% error reduction, demonstrating the value of having full access to model gradients and intermediate states, which is not possible with black-box LLM APIs. These results suggest that specialized fine-tuned models remain state-of-the-art for narrow, well-defined tasks requiring precise control and guarantees. For a qualitative comparison of model outputs with attribute-level analysis, see Appendix A.

5.5 Application in Data Augmentation

We demonstrate the utility of LINGCONV for data augmentation on three GLUE tasks: CoLA, SST-2, and RTE. By generating paraphrases of training samples with specific linguistic properties, we show that the effectiveness of augmentation is highly dependent on the attributes of the synthetic data. We created "Effective" and "Ineffective" sets of augmented data by sampling target attributes to either increase or decrease the prevalence of certain linguistic features. Our results (Table 3) show that "Effective" augmentation yields statistically significant performance improvements on downstream tasks, while "Ineffective" augmentation can harm performance. This highlights the importance of controlled generation for creating high-quality, targeted training data.

The number of training and test samples for CoLA, SST-2, and RTE are 8.5k and 1k, 67k and 1.8k, and 2.5k and 3k, 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

Augmentation	CoLA (Matthew's Corr.)		RTE (Acc.)		SST-2 (Acc.)	
	Limited Data	Full Data	Limited Data	Full Data	Limited Data	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 3: 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.

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 3 confirms that the distribution of the target attributes influence the effectiveness of data augmentation.

We find that on RTE (Limited), 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 M.

6 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 evaluates models' adaptability to novel attributes and acts as a robust test for controlled paraphrase generation models. In addition, we evaluate the model on the downstream application of generating synthetic data for augmentation. Our model generates paraphrases that boost performance and can be used to mitigate dataset biases. Future work can investigate mechanisms to handle contradictory or noisy attribute specifications to enable the model to resolve conflicts and prioritize constraints, and extend LINGCONV beyond English to multilingual and low-resource settings, where new attribute extractors and cross-lingual transfer will be needed.

Limitations

Our approach requires the availability of linguistic attributes, which, although available for the English language, may not be available for all languages. Certain linguistic attributes may require more sophisticated control mechanisms. The direct injection of embedded linguistic attributes into the decoder input in LINGCONV, although effective, has weaknesses. Specifically, we find it to be sensitive to outlier linguistic targets. If the linguistic target contains extreme values, we find that the model degenerates into non-grammatical and repetitive text.

In addition, while `flan-t5-base` (Chung et al., 2024) was likely exposed to the training sets of our evaluation tasks during its pre-training, we conduct all evaluations strictly on held-out test sets. We believe any potential data contamination is mitigated because our core challenge is fine-grained attribute control, and our evaluation uses novel at-

tribute combinations and reference-free metrics in the challenging scenarios.

An estimate of human performance serves is a useful baseline. However, the task of generating paraphrases that adhere to an extensive set of 40 linguistic attributes is beyond the capabilities of even expert linguists, making human evaluation impractical and potentially unreliable. Fortunately, we have direct access to the same software tools that precisely compute these linguistic attributes for both the source texts and generated outputs, enabling straightforward and highly accurate automatic evaluation. These deterministic tools calculate exact values for each attribute with consistent reliability, eliminating the subjectivity and variability inherent in human judgments. Our evaluation framework combines BERTScore, which has shown strong correlation with human judgments of semantic similarity in prior work (Zhang et al., 2020), with precise attribute measurements that objectively quantify how closely the generated text adheres to target linguistic specifications.

Regarding the performance of our linguistic discriminator, we evaluated its accuracy on predicting 40 different linguistic attributes. The model achieves an average mean square error of 0.52 on the test set, which is significantly lower than the naive baseline of predicting the mean value for each attribute (MSE=1.0). The accuracy varies by attribute category, with some syntactic features (e.g., clause count, T-unit count) being more accurately predicted (average MSE=0.35) than lexical features like sophisticated word usage (average MSE=0.61). This performance gap reflects the inherent complexity of modeling certain linguistic phenomena and represents a limitation of LINGCONV.

Ethical Statement

Controlled text generation needs ethical considerations. There is a fine line between controlled generation and manipulation. Malicious actors may use such a model for the propagation of biased, misleading, or harmful information. We must ensure that the technology is disseminated responsibly, with safeguards in place to prevent malicious usage and unintended consequences. Furthermore, these models allow for generating paraphrases with great diversity that may be undetectable in cases of plagiarism. More sophisticated safeguards around plagiarism, cheating, and theft must be put in place to address this issue.

Broader Impacts

The implications of our complexity-controlling paradigm are wide-ranging and significant. By generating more accessible text, this technology extends its reach to individuals with limited literacy proficiency, cognitive impairments, learning disabilities, aphasia, or dementia. Moreover, it allows for personalized communication, functions as a valuable tool for linguistic researchers and natural language processing (NLP) experts, and enhances the pedagogical landscape of second language acquisition by dynamically adapting text complexity to match the learner's skill level. In addition, our approach addresses the conversion of text complexity through fine-grained control of linguistic attributes. Text complexity plays a crucial role in text readability, comprehension, and propriety for different readers. For example, an educational platform that dynamically adjusts the complexity of its content to match the reader's proficiency level can enable better comprehension and engagement. Such personalized learning experiences can potentially revolutionize education by adjusting complexity with respect to the learner's capabilities and accommodating a wider range of learners. Our model can also help content creators to tailor their messaging to their target audience.

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.

Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. 2011. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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.

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.

Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. 2024. Controlled text generation via language model arithmetic. In *The Twelfth International Conference on Learning Representations*.

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.

Mohamed Elgaar and Hadi Amiri. 2023. Ling-CL: Understanding NLP models through linguistic curricula. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13526–13542, Singapore. Association for Computational Linguistics.

Mohamed Elgaar and Hadi Amiri. 2025. Lingconv: An interactive toolkit for controlled paraphrase generation with linguistic attribute control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Suzhou, China. Association for Computational Linguistics.

Explosion AI. 2025. spaCy English stop words list. `https://github.com/explosion/spaCy/blob/master/spacy/lang/en/stop_words.py`. Accessed: 2025-05-19.

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.

Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. 1999. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21(1):185–194.

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.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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. 2023. 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.

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 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.

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.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. 2020. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, Online. 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. 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.

Tong Niu, Semih Yavuz, Yingbo Zhou, Nitish Shirish Keskar, Huan Wang, and Caiming Xiong. 2021. Unsupervised paraphrasing with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5136–5150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuki Okano, Kotaro Funakoshi, Ryo Nagata, and Manabu Okumura. 2023. Generating dialog responses with specified grammatical items for second language learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 184–194, Toronto, Canada. Association for Computational Linguistics.

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.

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.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3178–3190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ning Shi and Zijun Wu. 2024. Action controlled paraphrasing. *arXiv preprint arXiv:2405.11277*.

Masaki Uto, Yuto Tomikawa, and Ayaka Suzuki. 2023. Difficulty-controllable neural question generation for reading comprehension using item response theory. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 119–129, Toronto, Canada. 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.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. 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.

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.

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.

Linguistic Attributes	
Characters per Sentence	Word Count
Words per Sentence	Character Count
Syllables per Sentence	Sentence Count
Characters per Word	Clause Count
NORP Entities	T-unit Count
GPE Entities	Noun Count
Law Entities	Numeral Count
Money Entities	Stop Words
Ordinal Entities	Proper Nouns
Sophisticated Words	Complex T-units
Sophisticated Word Count	Complex Nominals
Sophisticated Lexical Words	Dependent Clauses
Lexical Sophistication	Verb Sophistication
Unique Word Ratio	Total Words
Unique Verb Ratio	Unique Lexical Words
Coordinating Conjunctions	Unique Adverb Ratio
Subordinating Conjunctions	Unique Adjective Ratio
Age of Acquisition Score	Readability Level

Table 4: Linguistic indices used in this paper.

A Qualitative Analysis

Figure 3 presents a qualitative analysis of outputs from different models, highlighting the varying capabilities in linguistic attribute control. This analysis provides insight into the strengths and limitations of our approach compared to existing methods.

B List of Linguistic Attributes

We use expert-crafted linguistic indices as the control attributes for CPG. Table 4 lists all the indices that we use. We select 40 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.

The linguistic indices employed in our work are derived from off-the-shelf tools that implement linguist-defined rules grounded in psycholinguistics literature. Lexical and surface-level indices, for example, are computed by simply counting word occurrences, ensuring robust and reliable measurements. Syntactic and discourse indices are extracted using part-of-speech (POS) tagging, dependency parsing, and named entity recognition (NER), for which we employ the `en_core_web_sm` model from spaCy (Honnibal et al., 2020). This model has been reported to achieve 97% accuracy on POS tagging, 92% on parsing, and a 0.84 F1-score on NER, verifying the reliability of these derived attributes. Moreover, the deterministic nature of these algorithms guarantees consistent results across experiments. All models and baselines in our study are evaluated using the same input indices and evaluation process, ensuring fair comparison of their linguistic-control capabilities. 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.

C Comparison of Injection Methods

Table 5 presents a comparison of different injection methods for integrating linguistic attributes into the generation process. Results are reported on both standard and novel targets test sets in terms of mean squared error for target (l^t) and source (l^s) linguistic attributes, as well as BERTScore (BERTS).

We 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.

D Quality Control Algorithm Details

The quality control (QC) mechanism is designed to optimize the alignment of generated text with target linguistic attributes while preserving semantic equivalence to the source. The QC loop adjusts the text embeddings iteratively using a gradient-based method combined with a line search to minimize attribute errors, following the scientific approach outlined in Padmakumar et al. (2023); Armijo (1966); Boyd and Vandenberghe (2004).

The process consists of two key components:

Example 1: Lexical brevity

Source	U.N. inspectors later said the documents were old and irrelevant – some administrative material , some from a failed and well-known uranium-enrichment program of the 1980s .
Target	2 sophisticated words, 5 total words
LingConv Gen	U.N. inspectors said the documents were old.
QCPG Gen	The documents were old.

Attribute	Target	LingConv	QCPG
Sophisticated words	2	2 (0)	0 (**-2**)
Total words	5	7 (**+2**)	4 (-1)

Summary	LingConv matches the sophistication requirement; QCPG omits it. Both simplify, but only LingConv aligns with the target.

Example 2: High complexity

Source	Named in the complaint were former chief executive officers Paul A. Allaire and G. Richard Thoman and former CFO Barry D. Romeril .
Target	10 sophisticated words, 26 words, 4 complex nominals
LingConv Gen	The executives fined included former Chief Executives Paul A. Allaire and G. Richard Thoman as well as former chief financial officer Barry Romeril, all of the company's board members.
QCPG Gen	The executives fined included former Chief Executives Paul A. Allaire and G. Richard Thoman as well as former Chief Financial Officer Barry Romeril, among others.

Attribute	Target	LingConv	QCPG
Sophisticated words	10	8 (**-2**)	9 (-1)
Total words	26	30 (**+4**)	25 (-1)
Complex nominals	4	3 (-1)	1 (**-3**)

Summary	Both models generate relevant summaries but cannot fully match the high attribute targets.

Example 3: Structural complexity

Source	The puppy tried to get out of the tub.
Target	20 words, 4 verb phrases, 3 clauses, 3 complex nominals
LingConv Gen	A fluffy puppy is trying to get out of the tube while he is holding a plastic drawer.
QCPG Gen	Even though this puppy is extremely sensitive, it is still trying to get out of the cat's water.

Attribute	Target	LingConv	QCPG
Total words	20	18 (-2)	19 (-1)
Verb phrases	4	3 (-1)	3 (-1)
Clauses	3	2 (-1)	2 (-1)
Complex nominals	3	2 (-1)	1 (**-2**)

Summary	Both models fail to reach the structural targets; QCPG's paraphrase also drifts semantically.

Figure 3: **Qualitative comparison of LingConv and QCPG.** For each attribute, the target, each model's value, and the error magnitude are shown. Large errors are bolded.

Injection Method				Novel Targets		
	MSE(l^t)	MSE(l^s)	BERTS	MSE(l^t)	MSE(l^s)	BERTS
Encoder Input Concatenation	0.61	0.89	94.8	4.85	9.46	86.7
Encoder Input Addition	0.62	1.08	94.3	6.31	12.75	85.2
Decoder Input Concatenation	16.52	18.60	85.5	28.99	38.16	82.3
Decoder Input Addition	0.59	0.94	94.2	10.90	15.62	85.7
Decoder Input Addition to First Token	0.58	0.91	95.1	4.32	11.55	69.0
Layer 1 Addition (all tokens)	0.56	1.03	94.0	7.25	8.92	84.3
Layer 1 Addition (first token)	10.01	10.20	82.0	56.63	59.79	80.1
Layer 6 Addition (first token)	12.45	12.89	83.1	62.31	65.44	81.2
Layer 12 Addition (all tokens)	15.67	16.02	80.5	68.92	71.35	78.9
Layer 12 Addition (first token)	16.89	17.25	79.8	71.54	73.88	77.5

Table 5: Comparison of injection methods for linguistic attribute integration. Results include mean squared errors (MSE) for the target (l^t) and source (l^s) attributes and BERTScore (BERTS), measured on in-distribution (ID) and out-of-distribution (OOD) test sets.

1. An iterative refinement process that repeatedly updates the generation until it matches the target attributes or further improvement becomes impossible.

2. A line search algorithm that finds the optimal control strength for each refinement step while preserving semantic coherence.

Algorithm 1 shows this 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 and compute the mean squared error, l_0, between the predicted attributes and the target attributes (line 5). The gradient g of this error with respect to the input embeddings provides the direction for updates (line 6).

The adaptive step size is determined through a modified line search algorithm (lines 11–31) that finds the smallest viable step size that improves the output. The resulting generation must satisfy two conditions: (a) The predicted linguistic attribute error should decrease (i.e., be less than l_0) (b) The semantic equivalence probability should remain above a threshold τ

These conditions ensure both improved attribute control and semantic preservation. The process continues until no viable step size can be found, indicating the generation has reached its optimal state.

E Algorithm Background

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

E.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.

E.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

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.

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.

F 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.

G Training Data Preparation

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, we augment the data with self-paraphrase pairs: $\{s_i, s_i, l_i^s, l_i^s\}$ and $\{t_i, t_i, l_i^t, l_i^t\}$. This ensures diversity in the types of linguistic conversions that the model can learn, and strengthens the semantic consistency within and across paraphrase pairs, which improves model's understanding and generation capabilities. We augment 25% of the training data.

This augmentation strategy significantly increases the diversity of our training data. Starting with our original dataset of approximately 370k samples, our augmentation approach creates a final training set of roughly 840k samples. By including reversed pairs and self-paraphrase pairs, we expose the model to a wider range of attribute transformation patterns. This diversity is crucial for the model to learn flexible linguistic transformations rather than merely memorizing specific source-target attribute pairs.

Importantly, our model learns to map from a dense, normalized 40-dimensional attribute space to generated text. This approach enables generalization to novel attribute combinations not seen during training, as the model learns a continuous function rather than discrete mappings. The inclusion of self-paraphrase pairs where source and target attributes are identical helps the model learn when to preserve aspects of the input, while the reversed pairs teach it bidirectional transformations between different linguistic styles. Together, these augmentation strategies ensure that LINGCONV can generate diverse outputs tailored to various attribute specifications rather than being constrained to a limited set of transformations.

H 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-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

	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
+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.

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): an instruction fine-tuned LLM.

I Experimental Settings

Before adapting all baselines to the `flan-t5-base` backbone for our comparative experiments, we first replicated their original results using the official code and recommended configurations provided by the respective authors. This ensured faithful reproduction of their approach.

For inference efficiency, our model takes approximately 25 ms/token, compared to FUDGE (112 ms/token), BOLT (114 ms/token), and LLama (162 ms/token), making it significantly more efficient for practical applications. Detailed hyper-parameter settings are provided in Appendix I.

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 5.5, 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[3] with the "kmeans" clustering strategy.

The per-dataset results in Table 8 demonstrate the consistent superiority of LingConv across all three datasets. On QQP (the largest dataset), LingConv reduces attribute error by 39.5% compared to QCPG, while maintaining comparable BERTScore. Adding quality control further reduces the error by 15.4%. Similar patterns are observed on STS-B and MRPC, with LingConv+QC achieving the lowest attribute errors of 2.65 and 3.66 respectively, showing that our approach generalizes well across diverse paraphrase collections regardless of domain or size.

J 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:

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

	QQP		STS-B		MRPC	
Model	**B-S**	**MSE(l^t)**	**B-S**	**MSE(l^t)**	**B-S**	**MSE(l^t)**
Ref	**100.0**	10.22	**100.0**	8.63	**100.0**	10.58
Copy	94.6	10.16	94.1	8.68	94.6	10.77
T5-FT	95.0	10.44	94.0	8.64	93.5	10.59
Llama	91.1	7.98	90.7	7.64	90.8	11.25
BOLT	90.5	7.92	89.5	7.04	87.8	7.76
FUDGE	92.6	6.79	91.1	7.07	89.5	8.04
QCPG	90.4	6.12	90.6	4.56	89.9	6.56
Lingconv	90.9	3.70	91.3	3.55	90.5	4.57
+QC	90.8	**3.13**	90.8	**2.65**	89.7	**3.66**

Table 8: Performance breakdown by dataset on the Novel Target Challenge. B-S is BERTScore and MSE(l^t) is mean squared error of target attributes (lower is better). Results show LingConv consistently outperforms baselines across all datasets, with the quality control mechanism providing further improvements.

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 9: 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.

J.1 Controlling Discourse Proves Most Challenging

Table 9 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 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.

Our model achieves a 34% error reduction over LLama in attribute control. While large models like LLama-70B excel at general-purpose generation, our results show they struggle with precise attribute control (MSE(l^t) of 8.90 vs LingConv's 3.69 in novel target scenarios). The quality control mechanism provides an additional 14% error reduc-

tion, demonstrating the value of having full access to model gradients. These results suggest that the black-box nature of prompt-based approaches limits their ability to achieve exact attribute matching.

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 2. 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).

J.2 Handling of Contradictory Attributes

In this paper, we assumed all input control vectors are linguistically valid. To extend our work to handle potentially contradictory attributes, a clear definition of such conflicts is needed. Contradictions can arise from structurally incompatible requirements (e.g., a control vector specifying both "no verbs" and "three verb phrases") or from attributes that are strongly negatively correlated (e.g., requesting a "higher number of clauses" alongside a "lower average sentence length"). In such cases, we hypothesize that the model would prioritize the more commonly observed attribute pattern from its training data, effectively ignoring the outlier request. For instance, it would likely ignore a "no verbs" constraint in favor of producing a grammatically plausible sentence with verb phrases. Analyzing model behavior with conflicting targets is a valuable direction for future work, and could reveal insights into the model's implicit linguistic biases and trade-off strategies.

K 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.

L Imputation of Missing Values

Missing linguistic attribute values are imputed using the Multiple Imputation by Chained Equations (MICE) algorithm (Azur et al., 2011). For each missing attribute, a regression model is fitted using the other observed variables, and missing values are imputed based on this model. This process is repeated for 1000 iterations for each variable with missing data, forming a chain of equations that leads to iterative refinement. We use a Ridge Regression (Golub et al., 1999) linear model as the estimator with $\alpha = 1000$ to ensure robust predictions.

The regression models are fitted using a training set consisting of ground-truth linguistic attribute vectors from the training data of LingConv. Before the initial iteration of MICE, missing values are initialized using the mean value for each attribute, allowing prediction of a missing attribute as a function of all other attributes.

MICE leverages the relationships among variables to handle missing data. In the context of linguistic attributes, there are fixed relations between many of the attributes (e.g., any lexical count cannot be larger than the total number of words, and the number of clauses cannot be larger than the number of sentences in the text). By using ridge regression models within the MICE framework, these relationships are preserved while providing regularized predictions that avoid overfitting. This imputation mechanism enables users to specify only a few attributes of interest rather than all 40 attributes.

L.1 Imputation Experiment Results

We evaluate the performance of the MICE imputation method. We find that it leads to 0.02, 0.05, 0.06, 0.11 mean squared error, for imputing 20%, 40%, 60%, and 80% of the attributes, respectively, in the linguistic attributes that are imputed.

M Distributions of Augmentation Attributes

Figures 5-9 show the distributions of the biased attributes in the strong and weak sets of target linguistic variables.

Figure 5 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 that sentences with a higher diversity of verbs contribute to more effective augmentation, likely by enhancing the semantic richness of the generated data.

Figure 6 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 7 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 8 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 9 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.

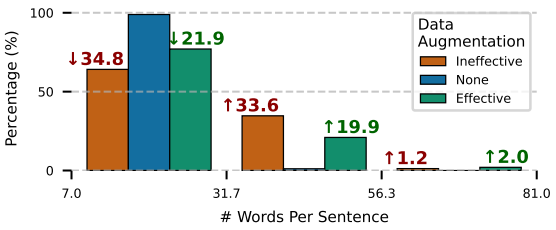


Figure 4: Attribute distributions for effective vs. ineffective augmentation on the RTE (Limited) dataset. Effective augmentation has a greater percentage of shorter sentences.

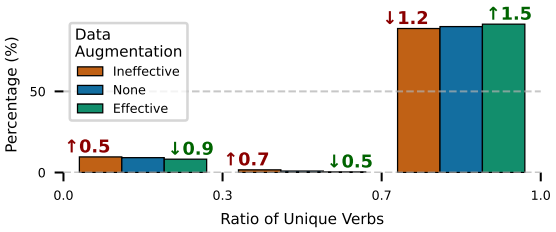
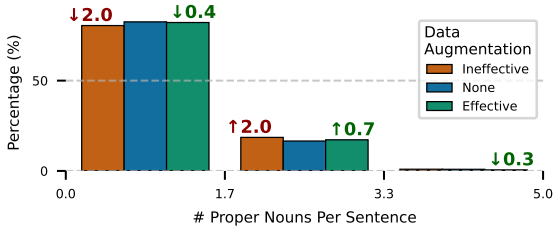
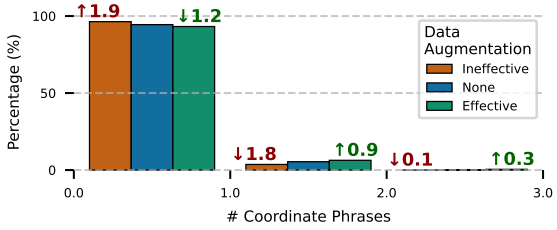


Figure 5: For CoLA (Limited), effective augmentation is associated with increased percentage of sentences with ratio of unique verbs > 0.7.

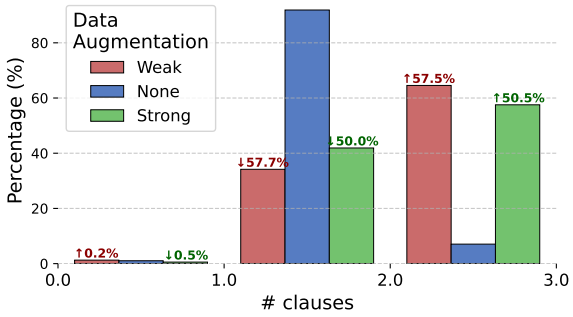


(a) Increase the percentage of sentences with a smaller number of proper nouns.

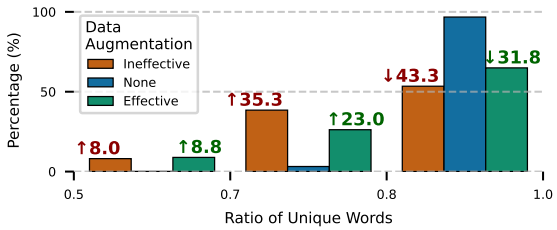


(b) Increase the number of sentences with more than 1 corodinate phrase.

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



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



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

Figure 7: For RTE (Full), we bias two attributes.

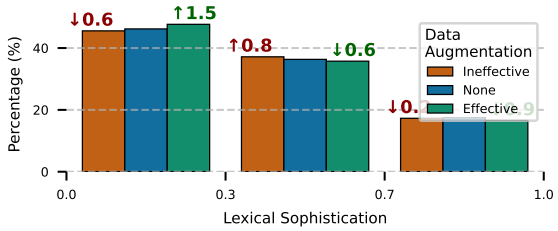
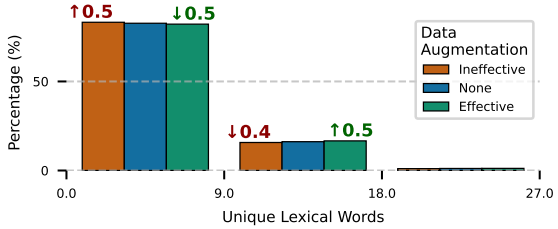
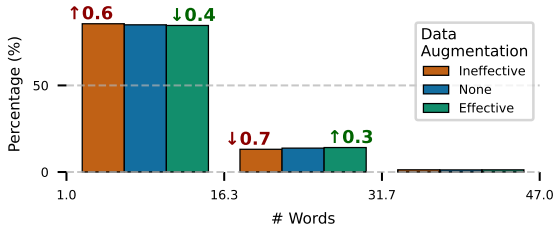


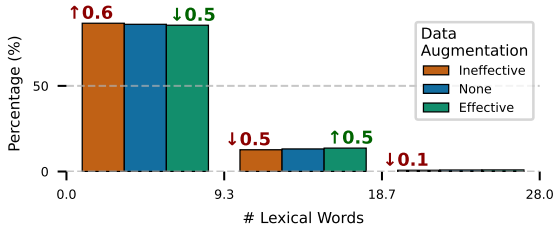
Figure 8: For SST-2 (Limited), decrease the ratio of sophisticated words.



(a) Increase number of unique lexical words.



(b) Increased average sentence length.



(c) Increase sentences with # Lexical Words > 9

Figure 9: For SST-2 (Full), we bias the number of lexical words, total words, and unique lexical words.