LINGCONV: An Interactive Toolkit for Controlled Paraphrase Generation with Linguistic Attribute Control

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

Abstract

We introduce LINGCONV, an interactive toolkit for paraphrase generation enabling fine-grained control over 40 specific lexical, syntactic, and discourse linguistic attributes. Users can directly manipulate target attributes using sliders, and with automatic imputation for unspecified attributes, simplifying the control process. Our adaptive Quality Control mechanism employs iterative refinement guided by line search to precisely steer the generation towards target attributes while preserving semantic meaning, overcoming limitations associated with fixed control strengths. Applications of LINGCONV include enhancing text accessibility by adjusting complexity for different literacy levels, enabling personalized communication through style adaptation, providing a valuable tool for linguistics and NLP research, and facilitating second language learning by tailoring text complexity. The system is available at https://mohdelgaar-lingconv.hf.space, with a demo video at https://youtu.be/wRBJEJ6EALQ.

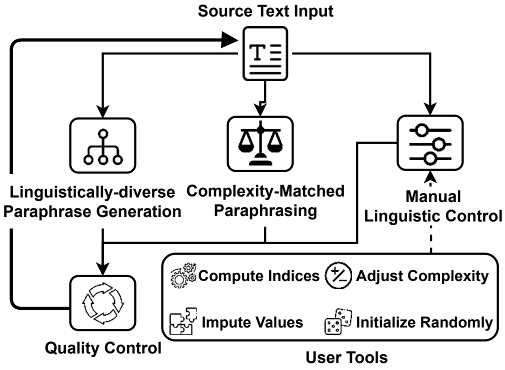

Figure 1: System architecture of LINGCONV. The system provides three modes of operation: Linguistically-diverse Paraphrase Generation for creating multiple diverse outputs, Complexity-Matched Paraphrasing for style transfer, and Manual Linguistic Control for fine-grained attribute adjustment. All modes utilize the Quality Control mechanism to ensure output quality. The set of User Tools support the manual specification of linguistic attributes.

1 Introduction

Controllable text generation, the task of producing text conforming to specified attributes like sentiment or formality (Jin et al., 2022), has seen widespread application in areas such as text simplification (Lee and Lee, 2023a,b; Vajjala and Lučić, 2018; Zhang and Lapata, 2017; Xu et al., 2015), toxicity control (Zheng et al., 2023; Zhang and Song, 2022; Liu et al., 2021), and personalized dialogue (Huang et al., 2023b; Niu and Bansal, 2018). While general large language models (LLMs) can be prompted to modify text style, achieving reliable, fine-grained, and verifiable control over multiple specific linguistic properties simultaneously remains a challenge (Shi et al., 2024).

We address this gap by introducing LINGCONV, an interactive toolkit specifically designed for controlled paraphrase generation (CPG) with explicit, fine-grained manipulation of 40 distinct linguistic attributes spanning lexical, syntactic, and discourse dimensions. LINGCONV allows users to generate paraphrases of a source text that precisely match a target linguistic style. This capability offers significant utility: for accessibility, text can be simplified or complexified for different reading levels; for personalization, communication can be tailored to specific user styles; for linguistics/NLP research, the system provides a platform for systematically studying the effects of linguistic variations; and for education, text complexity can be adjusted for second language learners.

The system is based on the CPG model described in Elgaar and Amiri (2025), which builds upon a T5 encoder-decoder model (Raffel et al., 2020), integrating the target linguistic attribute vector di-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 42–51
November 4-9, 2025 ©2025 Association for Computational Linguistics

rectly into the decoding process. Moreover, its adaptive Quality Control (QC) mechanism (§ 2.2) allows the system to iteratively refine the generation, matching target linguistic attributes while preserving semantic meaning, even when target attributes differ significantly from the source or involve complex transformations. Our QC approach employs gradient-based updates and an adaptive line search to dynamically adjust control strength.

The system architecture (Figure 1) supports three distinct modes of operation for different user needs: linguistically-diverse paraphrase generation, complexity-matched paraphrasing by example, and manual slider-based control for precise adjustments. Furthermore, the system provides tools for facilitating the manual specification of linguistic attributes (§ 3).

The system architecture (Figure 1) supports multiple modes of operation and includes tools to facilitate attribute specification (§ 3). As linguistic complexity attributes, we consider lexical, syntactic, and discourse psycholinguistic indices (Appendix A).

While some prior work focused on controlling syntactic structure through manipulations of parse trees or AMR graphs (Huang et al., 2023a; Goyal and Durrett, 2020; Iyyer et al., 2018), LINGCONV provides control over a set of 40 lexical, syntactic, and discourse attributes within an interactive framework with an adaptive QC mechanism for precision and robustness.

LINGCONV offers a range of advanced features and functionalities to provide users with greater control and flexibility over the generation process. Users can choose from three distinct paraphrase generation strategies: *Randomized Paraphrase Generation*, *Complexity-Matched Paraphrasing*, and *Manual Linguistic Control*. Additionally, the system allows users to select between exact or approximate linguistic index computation, which is used in interpolation and the manual setting of linguistic attributes. The system provides the option to show the intermediate sentences generated during the quality control process, enhancing transparency and interpretability.

2 System Architecture

2.1 Model

LINGCONV employs a T5-Base (Raffel et al., 2020) encoder-decoder model augmented with a linguistic complexity control approach to perform complexity-controlled paraphrase generation. Given a dataset $\mathcal{D} = \{s, t\}$ of paraphrase source and target pairs s and t, we compute the linguistic attributes of the target l^t. Thus, the task is to find a mapping from $(s, l^t) \to t$.

First, LINGCONV employs a linguistic embedding layer $h(l) = \mathbb{R}^k \to \mathbb{R}^{d_{model}}$, where k is the number of linguistic attributes (represented as a vector), and d_{model} is the input embedding dimension of T5. The embeddings for k linguistic attributes are learned jointly with the encoder-decoder model's parameters. Linguistic attributes are injected into the decoding process through element-wise addition to the embeddings of the first token of the decoder. The decoder attends to the linguistic embeddings at each generation step using self-attention (Vaswani et al., 2017). Through the training process, the decoder learns to steer the generation towards the desired target attributes. The model is trained using cross-entropy loss of the translation from source to target paraphrases.

2.2 Quality Control

In controlled text generation, achieving precise control over multiple linguistic attributes while maintaining text quality presents significant challenges. Not all combinations of linguistic attributes are feasible (such as having more unique words than total words), and even valid combinations may be difficult for the model to achieve in a single generation step.

To address these challenges, LINGCONV employs an iterative refinement process. Starting with an initial generation, the system gradually adjusts the output to better match the target linguistic attributes while preserving semantic meaning.

This refinement is achieved by computing the gradient of the linguistic attribute error with respect to the input embeddings. However, determining the appropriate update strength (steering factor) is critical: too small is ineffective, too large degrades quality. Fixed control strengths, explored in prior work (Durmus et al., 2024; Yang et al., 2024), are insufficient here because the norm of our gradient-based steering vector varies significantly with the attribute discrepancy.

LINGCONV addresses this through an adaptive control strength approach that uses line search to dynamically determine the optimal step size for each refinement iteration. The quality control mechanism consists of two key components: 1) An iterative refinement process that repeatedly updates

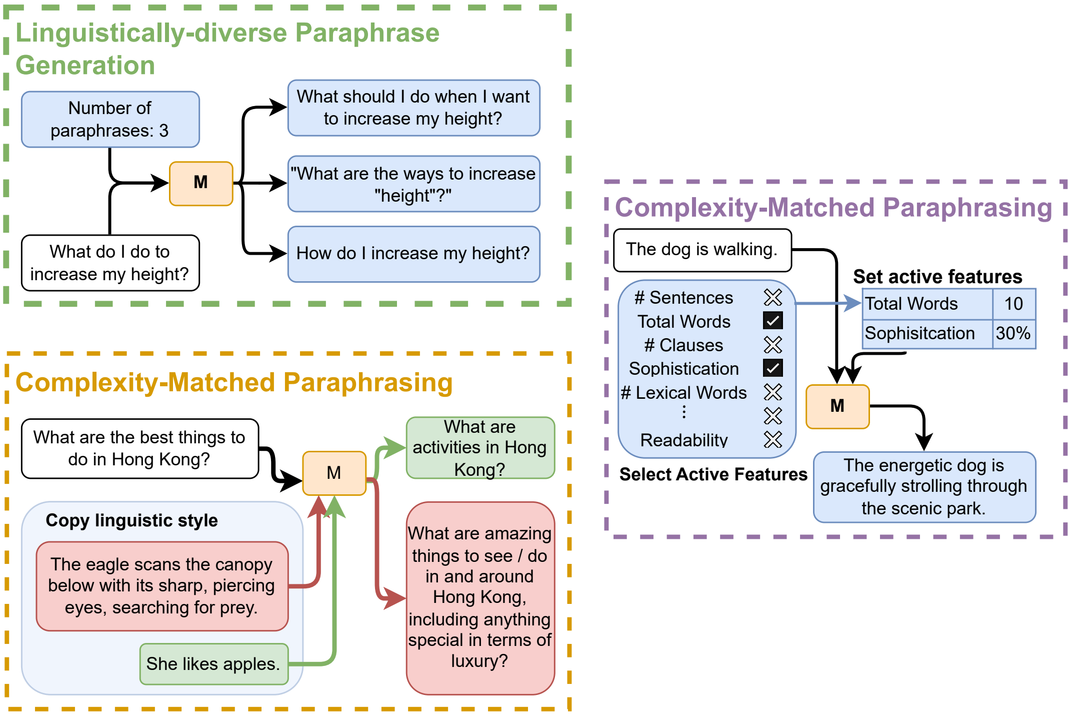
Linguistically-diverse Paraphrase Generation
Number of paraphrases: 3
What do I do to increase my height?
M
What should I do when I want to increase my height?
"What are the ways to increase "height"?"
How do I increase my height?
Complexity-Matched Paraphrasing
What are the best things to do in Hong Kong?
M
Copy linguistic style
The eagle scans the canopy below with its sharp, piercing eyes, searching for prey.
She likes apples.
What are activities in Hong Kong?
What are amazing things to see / do in and around Hong Kong, including anything special in terms of luxury?
Complexity-Matched Paraphrasing
The dog is walking.
# Sentences ☒
Total Words ☑
# Clauses ☒
Sophistication ☑
# Lexical Words ☒
⋮ ☒
Readability ☒
Select Active Features
Set active features
Total Words 10
Sophisitcation 30%
M
The energetic dog is gracefully strolling through the scenic park.

Figure 2: Examples of the three paraphrase generation modes of LINGCONV.

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

The quality control algorithm is illustrated in Figure 3. The iterative refinement process starts by generating an initial candidate output based on the input sentence and target attributes. Figure 4 illustrates this process with examples of intermediate outputs generated during refinement. It then enters a loop where it repeatedly refines the generation until it closely matches the target attributes, or further refinement is not possible.

The iterative process (Figure 3) starts with an initial generation and enters a refinement loop. Each iteration computes the attribute error and performs line search to find an optimal update strength (derived from the negative gradient). The process continues until the MSE between the candidate's attributes and the target falls below a threshold τ, or until line search fails to find an improvement, ensuring convergence even for challenging targets.

Backpropagation of the linguistic attribute error requires differentiable linguistic index computation. Thus, we pre-train **linguistic discriminator (LD)** and a **semantic embedding module (SEM)**.

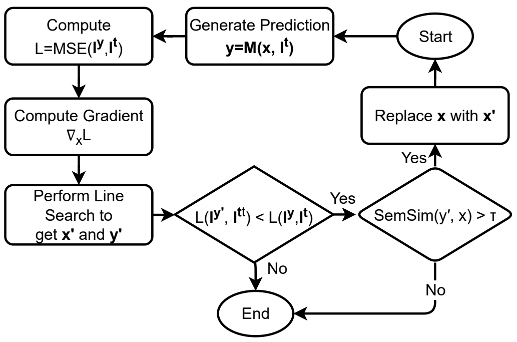

Figure 3: The quality control algorithm flowchart. The algorithm starts with initial generation and enters a refinement loop until no further improvement is possible. Each iteration computes the attribute error, performs line search to find optimal update strength, and generates a new candidate output.

The linguistic discriminator (LD) learns to predict the linguistic attributes of a sentence, providing the error signal for attribute control during QC. The semantic embedding module (SEM) is trained to predict the probability that the source sentence s and the generated paraphrase $\hat{t}$ are semantically equivalent. Their specific training objectives and architectures are detailed in Appendix C.

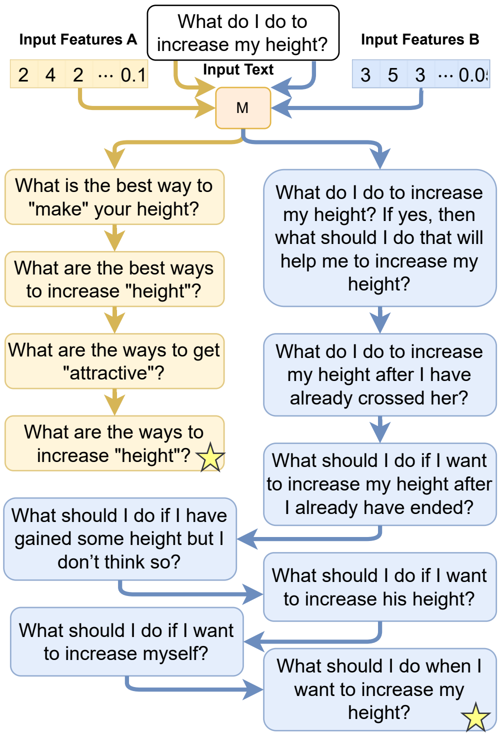
Input Features A
2 4 2 ··· 0.1
What do I do to increase my height?
Input Text
Input Features B
3 5 3 ··· 0.0
M
What is the best way to "make" your height?
What are the best ways to increase "height"?
What are the ways to get "attractive"?
What are the ways to increase "height"?
What do I do to increase my height? If yes, then what should I do that will help me to increase my height?
What do I do to increase my height after I have already crossed her?
What should I do if I want to increase my height after I already have ended?
What should I do if I have gained some height but I don't think so?
What should I do if I want to increase his height?
What should I do if I want to increase myself?
What should I do when I want to increase my height?

Figure 4: Two examples of intermediate texts iteratively generated at inference time using the quality control approach for text generation. Both examples are generated using the same source sentence and two different target attributes. The star indicates the returned value.

3 Features and Functionalities

As shown in Figure 2, LINGCONV offers three modes for fine-grained attribute adjustment.

The **Randomized Paraphrase Generation** mode serves users seeking diverse paraphrases; it automatically generates multiple linguistically varied paraphrases, discovering stylistic possibilities without requiring users to specify target attributes.

The **Complexity-matched Paraphrasing** mode addresses the need for textual style transfer; users provide a reference text, and the system generates a paraphrase of the source that mimics the reference's linguistic style, useful for adapting content to specific audiences or contexts. This also allows users to define the target linguistic style implicitly through an example, which can be easier than manually specifying numerous individual attributes.

The **Manual Linguistic Control** mode offers fine-grained manipulation; it allows expert users or those with specific requirements (e.g., researchers studying linguistic effects) to precisely adjust individual linguistic attributes using sliders, providing maximum control over the output.

Advanced Options allow toggling exact/approximate attribute computation and viewing intermediate QC steps. An **Examples** tab provides 150 validation set samples.

The linguistic attributes we use are extracted from three sources: lexical attributes developed by Lu (2012), syntactic attributes by Lu (2010), and a diverse set of semantic, lexical, discourse, and traditional attributes by Lee and Lee (2023a). A detailed list of the linguistic attributes used can be found in Appendix A.

The system implements error handling: Input validation for text length and content, automatic correction of invalid linguistic attribute combinations, graceful handling of model prediction failures, and user feedback for invalid operations.

3.1 Randomized Paraphrase Generation

The Paraphrase Generation feature provides a straightforward yet powerful way to generate multiple paraphrases of a given source sentence. Users begin by inputting the source sentence and indicating the desired number of paraphrases to be generated. The system then employs its linguistic attributes sampling and text generation algorithm to generate a set of distinct paraphrases, each adhering to a unique set of target linguistic attributes.

LINGCONV employs a large-scale repository of precomputed linguistic index sets. These sets, extracted from the training dataset, encompass a wide spectrum of linguistic attributes. By randomly selecting index sets from this collection, the system ensures that the generated paraphrases carry diverse linguistic characteristics. Figure 2 shows examples of paraphrases generated by LINGCONV using different generation modes.

3.2 Complexity-matched Paraphrasing

Given a source sentence and a reference sentence. The model extracts the linguistic attributes from the reference. Utilizing these extracted attributes as a guide, the system generates a paraphrase of the source sentence that mirrors the linguistic attributes of the reference. This form of textual style transfer enables users to seamlessly adapt their content to match a specific style or level of complexity. It is a valuable tool for authors, marketers, communicators, and clinicians looking to tailor their text to distinct audiences or contexts while maintaining semantic coherence. Figure 2 displays examples of using different modes, including complexity-matched paraphrasing.

3.3 Manual Linguistic Control

The system provides manual control through sliders corresponding to specific linguistic attributes, with bounds determined through statistical analysis. The interface implements: **Attribute Activation** for selective control, **Automatic Imputation** for inactive attributes, and **Range Constraints** to prevent invalid combinations. Users can activate specific linguistic attributes of interest and adjust their values using sliders, which are constrained by minimum and maximum values to ensure valid settings.

The Manual Linguistic Control mode provides a set of tools to increase accessibility for users. Users can focus solely on activating and adjusting the specific linguistic attributes of interest. The system then automatically imputes reasonable values for all other inactive attributes (see § 3.4), alleviating the need to manually specify the entire set of attributes.

To further assist manual setting of linguistic values, the system offers a set of tools accessible through the **Tools to assist in setting linguistic attributes** interface. As illustrated in Figure 1, these tools include several key functionalities. The **Random Target** generator produces valid target linguistic indices from the training dataset, while the **Impute Missing Values** function fills in remaining attributes to maintain coherence when only a subset is specified. Users can analyze existing text through the **Computing Linguistic Attributes** tool, which calculates or estimates linguistic attributes of input sentences. The **Copying Attributes** functionality enables streamlined transfer from source to target, particularly useful for incremental adjustments. Additionally, the **Adding and Subtracting ϵ** feature serves to incrementally increase text complexity and generate controlled variations through minor perturbations.

3.4 Imputation Process

The imputation of missing linguistic attributes is performed using the Multiple Imputation by Chained Equations (MICE) algorithm (Azur et al., 2011). For each missing linguistic attribute, a ridge regression model (Golub et al., 1999) with $\alpha = 1000$ is fitted using the other variables as predictors. The missing values are then imputed based on this model's predictions. This process is repeated for up to 1000 iterations for each variable with missing data, forming a chain of equations that leads to an iterative refinement process.

MICE models are trained on 1000 diverse attribute vectors selected greedily from the training data. Missing values are mean-initialized. MICE is well-suited for linguistic attributes as it leverages inter-attribute correlations (e.g., counts, clauses/sentences) via ridge regression, preserving relationships.

Appendix B.1 shows imputation performance using standard metrics: Mean Squared Error (MSE), and Root Mean Square Error (RMSE), and Pearson correlation coefficient (ρ). MSE and RMSE quantify the average magnitude of errors between imputed and ground-truth attribute values, and ρ measures the linear relationship between imputed and true values. Accuracy improves as more attributes are provided, with the best results at 80% known attributes, though performance remains reasonable even at 20%. See Appendix B for implementation details.

4 Data and Evaluation

4.1 Data and Experimental Setup

We utilize a combination of three paraphrase and semantic similarity corpora for training and evaluation: 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 dataset [1]. From these datasets, we retain only the positively labeled pairs, indicating semantic equivalence, resulting in a total corpus of 140,000 sentence pairs suitable for paraphrase generation. This combined dataset is randomly partitioned into training (80%), validation (10%), and testing (10%). The same data splits are used consistently across all experiments, and only sequences up to 100 tokens were included in training.

We train and evaluate eight baselines on the task of CPG. Three baselines produce paraphrases with no attribute control, and serve to demonstrate the quality of outputs in the case of non-controlled generation. The remaining five baselines are all recent and strong CPG approaches.

We evaluate our approach against several controlled generation baselines. As control baselines, we include a direct **Copy** of the source

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

Model	**BERTScore**^{F} ↑	**MSE(**$\boldsymbol{l^t}$**)**↓	**MSE(**$\boldsymbol{l^s}$**)**↑	**Overall**↑
Ref	94.4	9.82	0.96	0.19
Copy	100.0	9.86	0.00	0.33
T5-FT	97.8	9.86	0.29	0.27
Llama3.1-70B	**92.8**	8.90	2.44	0.26
BOLT	90.4	7.47	1.83	0.21
Fudge	92.5	7.22	3.11	0.37
QCPG	91.4	5.61	3.25	0.41
Lingconv	92.0	3.69	4.39	0.59
Lingconv+QC	91.5	**2.89**	**6.20**	**0.71**

Table 1: Controlled generation performance across evaluation metrics. BERTScore measures the semantic similarity between the generated paraphrase and the source sentence. Mean squared error (MSE) values reflect how close the linguistic attributes of the generated paraphrase are to the target (MSE($\boldsymbol{l^t}$)↓) or source (MSE($\boldsymbol{l^s}$)↑).

sentence and the ground-truth **Reference** paraphrase from the dataset. For learned models, we compare against **T5-FT** (Raffel et al., 2020), a vanilla T5 model fine-tuned on our paraphrase datasets, and **Llama3.1-70B** (Dubey et al., 2024), an instruction-tuned language model directed to generate attribute-controlled paraphrases. We also evaluate against recent controlled generation approaches: **BOLT** (Liu et al., 2023), which learns logit biases through attribute discriminator loss minimization; **Fudge** (Yang and Klein, 2021), which performs token-level attribute control during generation; and **Quality Controlled Paraphrase Generation (QCPG)** (Bandel et al., 2022), which uses special character prefixes for attribute control. Finally, we evaluate our base **LingConv** model described in §2.1, as well as an enhanced version incorporating the quality control algorithm (**+QC**).

We evaluate the quality of generation using the following four automatic metrics of text generation. **BERTScore**^{F} (Zhang et al., 2020) measures the semantic similarity between the generation and the source sentence; F refers to the reference-free metric (Shen et al., 2022). **MSE(**l^t**)** is the error in the attributes of the generation compared to the target attributes. MSE metrics are evaluated on the normalized attribute values to equalize the scale across attributes. **MSE(**l^s**)** is the distance between the attributes of the generation and the source attributes. This measures the bias of CPG methods towards the style of the source sentence, and their ability to generate a paraphrase with significantly different linguistic attributes from the source. **Overall** score normalizes each of the other three metrics to the range [0,1], such that a higher value is better, and computes the average. The overall score highlights the approach with the best trade-off between semantic similarity and accurate attribute control ability.

4.2 Results

Results in Table 1 show that the attribute control of LingConv is 34% more accurate than the second-best baseline, while being comparable in semantic equivalence. The addition of QC results in a further 14% decrease in attribute error. We attribute this higher performance to the effective use of linguistic complexity attributes in the decoding phase of LingConv.

Figure 4 presents examples of intermediate text outputs generated at inference time during the interpolation process for quality control of the generated texts. These results provide a clear illustration of the step-by-step generation process that progressively moves towards generating target sentences that meet desired levels of linguistic complexity.

5 Conclusion

We developed a new text conversion system, LingConv, which offers comprehensive features for complexity-controlled text generation. Through the careful integration of linguistic attributes and model architecture, LingConv provides users with the tools to generate text that adheres to both specific and diverse linguistic complexity levels.

The system's evaluation against recent controlled generation baselines and the vanilla T5 model verifies its reliability and effectiveness. The iterative refinement process with adaptive control strength enhances the system's ability to improve the generation process, ensuring high-quality outputs that closely match the desired linguistic attributes while maintaining semantic coherence. A current limitation is the system's focus on sentence-level paraphrasing, stemming from training on sequences up to 100 tokens; extending LingConv to effectively handle paragraph-level or longer document processing remains future work.

References

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.

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.

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 arXiv:2407.21783*.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Henighan, and Deep Ganguli. 2024. Evaluating feature steering: A case study in mitigating social biases.

Mohamed Elgaar and Hadi Amiri. 2025. Linguistically-controlled paraphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China. 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. *ArXiv*, abs/2005.02013.

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

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

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. 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.

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.

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 1774–1779.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. Dexperts: Decoding-time controlled text generation with experts and anti-experts. In *Annual Meeting of the Association for Computational Linguistics*.

Xin Liu, Muhammad Khalifa, and Lu Wang. 2023. 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.

Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, 15:474–496. Citation Key: Lu2010.

Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners' oral narratives. *Source: The Modern Language Journal*, 96(2):190–208. Citation Key: Lu2012.

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

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

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.

Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. 2024. Why larger language models do in-context learning differently? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44991–45013. PMLR.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

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.

Shu Yang, Shenzhe Zhu, Ruoxuan Bao, Liang Liu, Yu Cheng, Lijie Hu, Mengdi Li, and Di Wang. 2024. What makes your model a low-empathy or warmth person: Exploring the origins of personality in llms. *arXiv preprint arXiv:2410.10863*.

Hanqing Zhang and Dawei Song. 2022. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

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.

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

Table 2: List of linguistic attribute names controlled by LingConv.

A Linguistic Attributes

Table 2 is a list of the linguistic attributes controlled by LingConv.

Index Descriptions Below we provide brief descriptions for a selection of the linguistic indices controlled by LingConv.

- **Lexical Words**: Content-bearing parts of speech, specifically nouns, verbs, adjectives, and adverbs.
- **Sophisticated Words**: Words considered less common in general usage, which we define as the 2,000 least frequent words in the American National Corpus.

Given Attributes	**MSE**	**RMSE**	ρ
20%	0.842	0.917	0.763
40%	0.625	0.791	0.851
60%	0.413	0.643	0.912
80%	0.286	0.535	0.945

Table 3: Performance of the MICE imputation algorithm with varying percentages of given attributes. Lower values are better for MSE, and RMSE, and higher values are better for ρ.

- **Age of Acquisition**: The typical age at which a word is learned and integrated into a person's vocabulary.
- **T-unit**: A minimal unit of syntax, consisting of one main clause and all associated subordinate clauses.
- **Complex Nominals**: A category of syntactic structures including: (i) nouns modified by elements such as adjectives, possessives, prepositional phrases, relative clauses, participles, or appositives; (ii) nominal clauses; and (iii) gerunds or infinitives serving as the subject.
- **Readability Level**: An estimate of the U.S. grade level required for a reader to comprehend a text, based on the Automated Readability Index.
- **GPE (Geopolitical) Entity**: Named countries, cities, and states.
- **NORP Entity**: Named nationalities, as well as religious and political groups.

Further details on these attributes can be found in the original works of Lu (2012), Lu (2010), and Lee and Lee (2023a).

B Imputation of missing values

Imputation of missing values is performed using the Multiple Imputation by Chained Equations (MICE) algorithm (Azur et al., 2011). For each missing linguistic attribute, a regression model is fitted using the other variables. The missing values are then imputed based on this model. This process is repeated for t iterations for each variable with missing data, forming a chain of equations that leads to an iterative refinement process. We use a ridge regression (Golub et al., 1999) linear model as the estimator.

The regression models are fitted using a training set consisting of N ground-truth linguistic attribute vectors (described below), coming from the training data of LINGCONV. Before the initial iteration of MICE, and to allow for predicting a missing attribute as a function of all other attributes, the missing values are initialized using the mean value for the attribute.

MICE provides a solution to handle missing data by leveraging the relationships among variables. In linguistic attributes, there are fixed relations between many of the attributes. Two examples are: 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 linear regression models within the MICE framework, it ensures that the linear relationship assumption is maintained.

B.1 Performance of the MICE imputation algorithm

Table 3 shows the performance of the MICE imputation algorithm with varying percentages of given attributes. Lower values are better for MSE, and RMSE, and higher values are better for ρ.

B.2 The stored set of linguistic attributes vectors

The stored set of linguistic attributes vectors is selected from the training data using a greedy algorithm that maximizes the distance between the selected normalized linguistic attribute vectors. The idea is to select a subset of data points that are most representative of the entire dataset. We start by computing pairwise Euclidean distances between all points in the original dataset. Then, we select the next data point with the maximum average distance from already chosen points, ensuring that the selected subset is diverse. We set $N = 1000$, while the full training dataset of LINGCONV contains over 250k samples. To apply the MICE algorithm, we concatenate this representative subset with the vector of missing values, and perform the imputation. This subset also serves as a bank of valid linguistic attribute vectors, from which we sample **random targets**.

B.3 Adding or Subtracting Complexity

This feature allows users to increase or decrease the overall complexity of the target attributes. To ensure the linguistic attributes remain valid, any changes must be proportionally scaled according to their linear relation. We derive a set of attribute ratios from the training data to guide this process.

During the modification, we randomly select a scaling factor between 0.5 and 4. This factor adjusts the attribute ratios before applying the changes to the attributes. For example, if the number of sentences is increased by 1.0, the total number of words increases by 9.0 on average, while other attributes increase proportionally according to their linear relations.

C Training of Auxiliary Modules

The linguistic discriminator (LD) takes a tokenized sentence as input and is trained to minimize the MSE between the predicted attributes and the ground-truth attributes of the sentence. The objective is formulated as:

$$\ell_{disc}(x) = \|\mathrm{LD}(x) - l^x\|_2^2, \qquad (1)$$

where x is the input sentence and l^x are its ground-truth linguistic attributes.

To ensure that the generated text remains semantically coherent with the source, the semantic embedding module (SEM) takes the source sentence s and the generated sentence $\hat{t}$ as input. It is trained using a contrastive objective to minimize the distance between embeddings of semantically equivalent pairs while maximizing the distance for non-equivalent pairs:

$$\ell_{sem}(s, \hat{t}) = -\log \frac{\mathrm{SE}(s, \hat{t})}{\sum\limits_{t' \in \mathcal{N}(s)} \mathrm{SE}(s, t')}, \qquad (2)$$

where $\mathcal{N}(s)$ represents the set of negative (non-paraphrase) examples for source s within the mini-batch. Both LD and SEM typically utilize architectures based on pre-trained encoders like T5, followed by appropriate projection layers for their respective tasks.