Fine-grained Controllable Text Generation through In-context Learning with Feedback

Anonymous ACL submission

Abstract

We present a method for rewriting an input sentence to match specific values of nontrivial linguistic features, such as dependency depth. In contrast to earlier work, our method uses in-context learning rather than finetuning, making it applicable in use cases where data is sparse. We show that our model performs accurate rewrites and matches the state of the art on rewriting sentences to a specified school grade level.

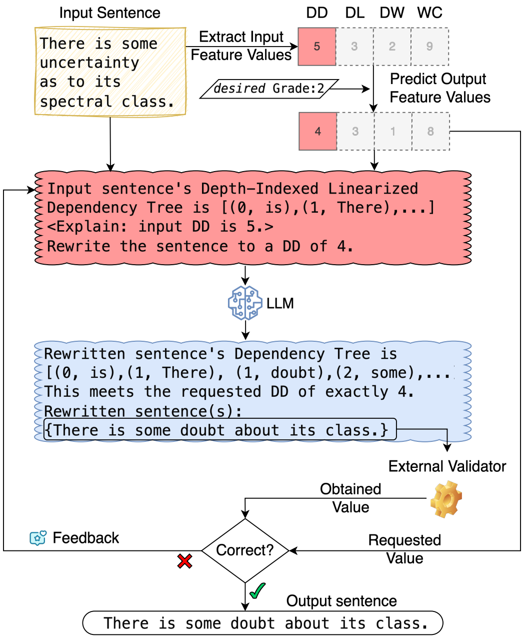
1 Introduction

Individual readers of text vary greatly in how they comprehend language, influenced by diverse cognitive properties such as working memory capacity, lexical access speed, and prior knowledge. We investigate the task of rewriting a given input text to be optimally comprehensible by a given reader.

Previous work (Agrawal and Carpuat, 2023) has shown that this can be done effectively through *controllable text generation with linguistic features (CTG-LFs)* (Martin et al., 2020, 2022): an input sentence is concatenated with a specification of certain linguistic features (e.g. the permissible syntactic complexity), and an LLM rewrites this input into the desired output sentence. A common strategy is to approximate the reader's cognitive abilities with a school grade level to which the text is supposed to be rewritten; such grade levels are annotated in some corpora (Xu et al., 2015) or can be automatically calculated (Kincaid et al., 1975).

Existing methods for CTG-LF generally use finetuning of pretrained language models such as T5 (Sheang and Saggion, 2021; Agrawal and Carpuat, 2023). Such methods require a large amount of parallel training data which pair original sentences with their reader-specific rewrites, which is implausible for reader group categories other than grade levels, or for languages other than English. It would be desirable to perform CTG-LF via in-context learning (ICL), as this does not require a training dataset.

However, existing prompting techniques struggle to accurately rewrite sentences to numeric specifications, such as word count (Sun et al., 2023). Methods for reader-specific rewrites with ICL (Farajidizaji et al., 2024) therefore focus on broad specifications of the rewriting target ("professional") and not on the fine-grained specification of individual linguistic features.



There is some uncertainty as to its spectral class.

Input sentence's Depth-Indexed Linearized Dependency Tree is [(0, is),(1, There),...]
<Explain: input DD is 5.>
Rewrite the sentence to a DD of 4.

Rewritten sentence's Dependency Tree is [(0, is),(1, There), (1, doubt),(2, some),...
This meets the requested DD of exactly 4.
Rewritten sentence(s):
{There is some doubt about its class.}

There is some doubt about its class.

Figure 1: Rewriting an input sentence to dependency depth 4 through prompting.

In this paper, we present a novel method for CTG-LFs based on in-context learning, using nontrivial linguistic features such as the depth and length of syntactic dependencies. We show that our model performs accurate rewrites, with e.g. 81% of test sentences being rewritten to the exact requested dependency depth. By combining our CTG-LFs model with a model that predicts the right values for the linguistic features, we obtain a system that rewrites sentences to a desired grade level. We show that it compares favorably to previous work for this task, using only five in-context examples as opposed to a whole training corpus.

2 Related Work

2.1 Controlled Text Generation with LLMs

The study by Martin et al. (2020, 2022), called ACCESS, conducted fine-grained CTG by fine-tuning LLMs with linguistic features such as character length, Levenshtein similarity ratio, word frequency rank, and dependency depth to modulate various facets of the output. Following ACCESS, (Sheang and Saggion, 2021) further incorporated word count.

Recent studies (Zhou et al., 2023a,b; Sun et al., 2023; Ashok and Poczos, 2024) on CTG with ICL focused on generating text under specified constraints such as keywords, length, or high-level aspects like style. These studies conclude that while LLMs are adept at incorporating broad control signals, they struggle with detailed, complex constraints.

2.2 Reader-specific Text Adaptation via CTG

Grade-level (Scarton and Specia, 2018) controls have been used in text simplification systems to simplify text for specific grade levels. A notable study by Agrawal and Carpuat (2023) demonstrated grade-specific simplification via CTG-LF, where a regressor predicts linguistic feature values for the target grade, followed by an ACCESS-based T5 fine-tuned model for text adaptation. A recent study on paraphrasing tasks by (Farajidizaji et al., 2024) used readability level controls with prompting to adapt the text for different target audiences.

3 Methodology

Our goal is to build a model that takes a sentence w and a specification of a reader as input and rewrites w to be optimal for that type of reader. We will approximate the specification of a reader with *school grade levels* (Scarton and Specia, 2018), which indicate a level of text complexity that is suitable for students of a certain grade in an American school. When grade levels are not annotated, they can be approximately predicted using automatic tools.

More specifically, we split the process of rewriting w for a target grade level into two steps. In the first step, we predict the values of four *linguistic features*, which describe the linguistic properties of the output text in more detail (§ 3.1). Second, we use in-context learning to perform *controlled text generation* to rewrite w to match the predicted feature values (§ 3.2). Our contribution is that we can perform this second step for nontrivial linguistic features with ICL.

3.1 Predicting Linguistic Features

We specify the form of the output sentence using four linguistic features: maximum dependency depth (DD), maximum dependency length (DL), word count (WC), and number of difficult words (DW). The first two features are defined in terms of a dependency tree for the output sentence; dependency depth is the length of the longest path from the root to a leaf, and dependency length is the longest distance in the string between a parent and a child. A word counts as "difficult" if it is not on the Dale-Chall list of easy words (Chall and Dale, 1995).

These features have been recognized as significant factors of processing difficulty (Nicenboim et al., 2015; Gibson, 1998; van Schijndel et al., 2013). In contrast to ACCESS, we directly specify difficult words rather than word frequency ranks; this facilitates in-context learning. We do not include the Levenshtein similarity score from ACCESS because it is a property of a pair of source and target sentence and cannot be calculated for one of the sentences by itself.

We automatically annotate the input and output sentences in a parallel corpus with values for the linguistic features using a dependency parser and word list. Additionally, we approximated the grade levels using the FKGL score (Kincaid et al., 1975). We use this corpus as training data for a *feature value predictor*, i.e. a model that predicts values for the features given the input sentence w and target grade level. We use a decision tree classifier to predict feature values, following Agrawal and Carpuat (2023).

3.2 CTG-LF with In-Context Learning

We can now talk about how to rewrite a sentence to fit the given linguistic features using ICL. As mentioned above, simple prompting techniques (e.g. "Generate a sentence using exactly 5 words") have been shown to work poorly (Sun et al., 2023).

Our approach combines two core ideas. First, we include an *analysis* of the input sentence in the prompt and ask the LLM to generate an analysis of the output sentence, followed by the output sen-

Prompt	**EM↑**				**EM ± 1 ↑**				**RMSE↓**			
Type	**DD**	**DL**	**DW**	**WC**	**DD**	**DL**	**DW**	**WC**	**DD**	**DL**	**DW**	**WC**
ZS SP w/o Input	49	17	20	70	79	61	68	99	1.4	3.9	1.4	0.6
ZS SP	19	18	22	30	60	41	53	68	2.0	4.2	2.9	1.9
ZS E-LF	31	20	18	38	70	38	64	67	1.5	3.3	1.9	2.6
FS E-LF	46	30	33	35	78	59	75	68	1.4	2.8	1.5	2.1
CoT ZS E-LF	22	17	19	40	73	40	59	79	1.6	3.2	2.3	2.2
CoT FS E-LF	40	36	43	50	73	59	72	75	1.4	2.8	1.6	2.6
CoT+FB FS E-LF	**81**	**57**	**86**	**78**	**98**	**75**	**93**	**88**	**0.6**	**1.8**	**0.7**	**1.1**

Table 1: Rewriting to specific linguistic feature values (ZS - Zero-shot, FS - 5 in-context examples)

tence itself. With an "analysis", we mean a representation of the sentence that makes a feature value explicit; the analysis takes the role of a thought in CoT reasoning (Wei et al., 2022). For DD and DL, we use linearized dependency trees in which each word is annotated with its depth or length; Fig. 1 shows an example for DD. For DW, the analysis is the list of difficult words in the sentence; for WC, it is simply the list of words in the sentence. Analyses allow us to incorporate explicit syntactic information into the prompting process; note, however, that the output analysis is generated by the LLM and not by a parser.

Second, we equip our model with a feedback mechanism (Shinn et al., 2024): after each LLM output, we run an *external validator* on the generated output sentence to determine its true feature values; e.g. a dependency parser for DD. If the feature value differs from the requested one, the LLM is called again, after amending the prompt with the true analysis of the generated output sentence and a feedback message such as "The maximum dependency depth of the rewritten sentence is 5; please revise it with a depth of 4." All previous LLM queries for this sentence, with the LLM response and the judgments of the parser, are included in the prompt. We permit up to 10 iterations of this feedback loop; if none yield the correct feature value, we return the output of the final iteration.

Values for multiple features can be specified at the same time by concatenating the descriptions and analyses for all the features.

4 Evaluation

We evaluate first the ability of our CTG model to rewrite to the requested feature values in isolation, and then the ability of the combined model to rewrite to a requested grade level.

Dataset. We use the WikiLarge text simplification dataset (Zhang and Lapata, 2017) as a training corpus. While this paper is *not* about text simplification – the rewritten sentences could in principle be less or more complex than the original –, WikiLarge is a convenient source of (mostly) meaning-equivalent sentence pairs of different linguistic complexity. After feature value annotation and basic cleaning, we create a train/validation/test split (see Appendix A.1 for details). We use the training data primarily to train the feature value predictor and select 5 in-context examples based on the annotated grade levels of the test sentences.

4.1 CTG to Linguistic Features

We begin by evaluating the ability of our model to rewrite an input sentence to match given values for the four linguistic features.

Evaluation measures. We rewrite every source sentence in the test set with respect to the gold feature values of its corresponding target sentence, using our CTG-LF model. For each test instance, we then determine the actual feature values of the rewritten sentence and compare them to the gold feature values of the target sentence. We report the exact match (EM) of the feature values, as well as the root mean squared error (RMSE) and a "softer" exact match that also counts feature values one above or below the gold value as correct (EM ±1).

Compared models. We use GPT-4o (version gpt-4o-2024-05-13) as our LLM for all experiments. We compare the full model of §3.2 (last row in Table 1) against a number of ablated baselines. In the table, "E-LF" means that the prompt includes a detailed definition of the feature and an analysis of the input sentence. "CoT" means that the output reasons step-by-step and includes an analysis of the output sentence. "FB" means

Methods	**EM↑**
Agrawal and Carpuat (2023)	**58.3**
Grade ZS	25.0
Grade FS	25.0
CoT ZS E-LF	41.7
CoT FS E-LF	50.0
CoT+FB FS E-LF	**66.7**

Table 2: Rewriting to specific grade levels.

that the model uses a feedback loop with an external validator. Example prompts are shown in the Appendix A.6. We also include a baseline "w/o Input", which simply prompts the LLM to generate any sentence with the given feature values; there is no input sentence to be rewritten. This mirrors the experimental setup of Sun et al. (2023).

Results. As Table 1 shows, the full model can manipulate the value of each feature quite accurately. Compared to previous studies (Sun et al., 2023), which used ChatGPT, our findings suggest that GPT-4o can handle low-level numerical constraints such as WC quite well, even with simple prompting techniques. However, these techniques still struggle with more complex syntactic constraints, such as DL. By including analyses for the input and output sentence and providing a feedback mechanism, we obtain a prompting mechanism that performs accurate CTG-LF.

4.2 CTG to Grade Levels

We perform an end-to-end evaluation of our model's ability to rewrite text to a specified school grade level by combining the feature value predictor (§3.1) with the CTG-LF model (§3.2). For each source sentence in the test set and each grade level from 1 to 12, we predict feature values and then rewrite the source sentence to those feature values. We never use the target sentences in the test set or any gold feature values; the gold feature values are only used to train the feature value predictor.

We calculate the grade level of the entire rewritten test corpus using the Document-Level Readability Consensus Score (DLRCS), which predicts grade ranges (e.g. "8–9") for the readability level of a whole document. The DLRCS combines a number of readability indices by majority vote - FRE, FKGL, Gunning FOG, SMOG Index, ARI, CLI, LW, and DCR. This guards it against the drawbacks and idiosyncracies of each index (Tanprasert and Kauchak, 2021). We use the implementation of DLRCS in the Textstat library. Because the DLRCS outputs categorical values rather than numbers, we use only exact match as our evaluation measure. We compare the DLRCS score of the predicted document (rewritten sentence by sentence wrt to a requested grade level) against that grade level.

As Table 2 indicates, our model achieves a grade-level rewriting accuracy that outperforms the best previous model (Agrawal and Carpuat, 2023), which fine-tunes T5 using ACCESS features. Additionally, our use of in-context learning makes our model far more flexible when parallel training data is sparse. Our models also decisively outperform prompting methods that simply request a target grade level ("Grade ZS/FS"), demonstrating the value of rewriting to a grade level by first predicting linguistic features.

A more fine-grained analysis (Table 4 in the Appendix) shows that our model outperforms T5 not only in aggregate, but also for most individual grade levels. The outputs of our full model match the requested grade level for 8 out of 12 grade levels (as per the DLRCS score), whereas the baseline only matches it for 7 grade levels. Furthermore, when our system's output grade level is incorrect, it is usually closer to the requested grade level. This further illustrates the ability of our model to generalize flexibly to the requested grade levels.

5 Conclusion

We presented the first method that successfully uses in-context learning to rewrite an input sentence to match fine-grained, nontrivial linguistic features. By combining this model with a predictor for feature values, we showed how to rewrite documents to a desired school grade level with state-of-the-art accuracy. Earlier models for fine-grained rewriting used finetuning; compared to these, our model has the advantage of requiring only some few-shot examples to perform accurate rewriting rather than a large parallel training corpus.

The ultimate goal of our work is to rewrite a sentence to be optimal for an individual reader. In this paper, we have taken grade levels as a proxy for a group of readers with similar cognitive properties, due to the availability of data. Future work will focus on extending our approach to smaller, more implicitly defined groups and individual readers – a step that will be facilitated by our ability to now use in-context learning rather than fine-tuning.

Limitations

Our study is limited in its use of a single LLM, GPT-4o. This is a closed LLM, and we fully recognize the reproducibility challenges that the use of such a model entails. We tried our method with open models in pilot experiments and found that their accuracy lagged behind that of the GPT-4 family significantly. As open instruction-tuned models further catch up to closed ones on complex tasks such as ours, we anticipate that our prompting technique will be effective for open models as well.

We will make the exact inputs and outputs of our evaluation experiments available upon acceptance in order to facilitate the comparison of future work to ours.

References

Sweta Agrawal and Marine Carpuat. 2023. Controlling pre-trained language models for grade-specific text simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12807–12819, Singapore. Association for Computational Linguistics.

Dhananjay Ashok and Barnabas Poczos. 2024. Controllable text generation in the instruction-tuning era.

J.S. Chall and E. Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books.

Asma Farajidizaji, Vatsal Raina, and Mark Gales. 2024. Is it possible to modify text to a target readability level? an initial investigation using zero-shot large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9325–9339, Torino, Italia. ELRA and ICCL.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

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

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Angela Fan, Eric De La Clergerie, Antoine Bordes, and Beno^ıt Beno^ıt Sagot. 2022. Muss: Multilingual unsupervised sentence simplification by mining paraphrases.

Bruno Nicenboim, Shravan Vasishth, Carolina Gattei, Mariano Sigman, and Reinhold Kliegl. 2015. Working memory differences in long-distance dependency resolution. *Frontiers in Psychology*, 6.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Kim Cheng Sheang and Horacio Saggion. 2021. Controllable sentence simplification with a unified text-to-text transfer transformer. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

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

Teerapaun Tanprasert and David Kauchak. 2021. Flesch-kincaid is not a text simplification evaluation metric. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14, Online. Association for Computational Linguistics.

Marten van Schijndel, Andy Exley, and William Schuler. 2013. A model of language processing as hierarchic sequential prediction. *Topics in cognitive science*, 5(3):522–540.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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.

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.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and

Language	Dataset Name		Size
English (EN)	WikiLarge	Train	216,881
		Dev	1,286
		Test	200

Table 3: Dataset details after data preprocessing

Le Hou. 2023a. Instruction-following evaluation for large language models.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023b. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

A Experimental setup

A.1 Data Preparation

The WikiLarge training dataset (Zhang and Lapata, 2017) which holds an MIT License, is presented in a tokenized format[1]. Upon examination, we observed that it incorporates various annotation styles. These include the usage of two single quotation marks (' '<text>' ') for standard double quotation marks("<text>"), as well as the utilization of specific abbreviations to denote different categories of parentheses and brackets, namely 'LRB' (Left Round Bracket), 'RRB' (Right Round Bracket), 'LCB' (Left Curly Bracket), 'RCB' (Right Curly Bracket), 'LSB' (Left Square Bracket), and 'RSB' (Right Square Bracket).

Our initial preprocessing step involved the substitution of these abbreviations with their respective typographic symbols to homogenize the text presentation. Subsequently, we de-tokenized the WikiLarge dataset utilizing the Moses de-tokenizer script[2]. Each source and target sentence was then annotated with linguistic feature values and grade levels. Following this, we randomly split the data into train, validation, and test sets, and removed duplicates from the training data.

Evaluation of CTG to linguistic features (§4.1) was conducted using 100 sentences from the test data (Table 3). This evaluation employed the gold target feature value, i.e., the actual feature value from the reference data, as the requested value to minimize any potential uncertainty. Our grade-level evaluation of CTG (§4.2) utilized 200 test sentences with predicted feature values corresponding to each grade level.

A.2 LLM setup

For the hyperparameter configuration, we set the temperature to 0 and the seed to 123, while maintaining default values for the remaining parameters.

A.3 Baseline

We fine-tune the T5-base model following (Agrawal and Carpuat, 2023) with default parameters from the Transformers library except for a batch size of 6, maximum length of 256, learning rate of 3e-4, weight decay of 0.1, Adam epsilon of 1e-8, 5 warm-up steps, and 5 epochs. For generation, we use a beam size of 8. Results were reported from a single run, trained on a Tesla V100-PCIE-32GB GPU with seed 12.

We use a learning rate of 0.1 and a tree depth of 6 for training feature predictor model as (Agrawal and Carpuat, 2023).

For grade-level prompting, we prompted the model with a specific grade level and FS by giving 5 examples according to the source and specified target grade level.

A.4 CTG to Grade Levels Evaluation

Table 4 shows the obtained DLRCS scores for rewriting to a specific grade level, ranging from 1st to 12th grade. Here, "2nd and 3rd" means the readability is suitable for students between 2nd and 3rd grade.

A.5 Example Outputs

Table 5 presents a few examples of outputs generated by the LLM using the CoT+FB FS E-LF method to adapt text to grade levels 1, 6, and 12.

A.6 Prompt examples

Sample prompts are provided below (in boxes) from page 8, illustrating how to rewrite a sentence to achieve a specific dependency depth. In these examples, the blue text represents the LLM-generated output, while the black text indicates our input prompt. The prompts are formatted as a user-assistant chat conversation.

We have also included our final prompt template (from page 8) that combines all linguistic features. Placeholders are indicated within curly braces (e.g., {print_dependency_tree_with_depth}).

---

[1] https://github.com/XingxingZhang/dress

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl

s	(Agrawal and Carpuat, 2023)	Grade ZS	Grade FS	CoT ZS E-LF	CoT FS E-LF	CoT+FB FS E-LF
1	2nd - 3rd	6th - 7th	7th - 8th	**1st - 2nd**	2nd - 3rd	**1st - 2nd**
2	**2nd - 3rd**	7th - 8th	7th - 8th	3rd - 4th	3rd - 4th	**2nd - 3rd**
3	**2nd - 3rd**	5th - 6th	7th - 8th	**2nd - 3rd**	**3rd - 4th**	**2nd - 3rd**
4	**4th - 5th**	8th - 9th	7th - 8th	**3rd - 4th**	**3rd - 4th**	**3rd - 4th**
5	**5th - 6th**	8th - 9th	8th - 9th	**4th - 5th**	**4th - 5th**	**4th - 5th**
6	**6th - 7th**	8th - 9th	8th - 9th	**5th - 6th**	8th - 9th	**6th - 7th**
7	**6th - 7th**	9th - 10th	9th - 10th	4th - 5th	9th - 10th	**6th - 7th**
8	5th - 6th	**8th - 9th**	9th - 10th	6th - 7th	9th - 10th	6th - 7th
9	**9th - 10th**	**9th - 10th**	**8th - 9th**	6th - 7th	**9th - 10th**	6th - 7th
10	8th - 9th	**9th - 10th**	**10th - 11th**	8th - 9th	**9th - 10th**	**9th - 10th**
11	8th - 9th	9th - 10th	**10th - 11th**	8th - 9th	**10th - 11th**	9th - 10th
12	8th - 9th	9th - 10th	10th - 11th	8th - 9th	10th - 11th	9th - 10th

Table 4: Obtained DLRCS for rewriting to specific grade levels.

Input	Cars did not finish the Grand Prix, but were classified as they completed over 90% of the race distance.
Grade 1	Cars did not finish the race. They completed over 90% of it. They were then classified.
Grade 6	Cars did not finish the Grand Prix but were still ranked as they completed over 90% of the race.
Grade 12	The cars did not finish the Grand Prix, but they were classified because they completed over 90% of the race distance, which met the required criteria.
Input	Worf is the first Klingon main character to appear in Star Trek, and has appeared in more Star Trek episodes than any other character.
Grade 1	Worf is a main character. He appears in Star Trek.
Grade 6	Worf is the first Klingon main character to appear in Star Trek history. He has appeared in more episodes than any other character in the Star Trek series.
Grade 12	Worf is the first main Klingon character in Star Trek, and he has appeared in more episodes than any other character, making him a key figure in the series.

Table 5: Example output generated for Grade 1, 6, and 12 by GPT-4o using our proposed method CoT+FB FS E-LF.

ZS SP w/o Input

User: Generate a sentence with a maximum depth of the dependency tree of exactly 5.

Assistant: Sure, here’s a sentence with a dependency tree depth of exactly 5: "The cat that the boy who lives next door adopted last week is very playful." In this sentence, the depth of the dependency tree is 5, as follows: 1. "is" (root) 2. "playful" (complement of "is") 3. "cat" (subject of "is") 4. "that" (relative pronoun modifying "cat") 5. "adopted" (verb in the relative clause modifying "cat")6. "boy" (subject of "adopted") 7. "who" (relative pronoun modifying "boy") 8. "lives" (verb in the relative clause modifying "boy") 9. "next door" (prepositional phrase modifying "lives") 10. "last week" (adverbial phrase modifying "adopted")The longest path from the root "is" to a leaf node (e.g., "door" or "week") involves 5 steps.

ZS SP

User: Input Text: Mar del Plata is an Argentine city located on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.
Rewrite the input text to maintain its original meaning while strictly adhering to a maximum depth of the dependency tree of exactly 5.

Assistant: Mar del Plata is a city in Argentina on the Atlantic coast, 400 km south of Buenos Aires.

ZS E-LF

System: You are an expert in text adaptation, skilled at meticulously rewriting texts to align with requested linguistic features while preserving the original meaning. Your task is to rewrite the provided Input Text to maintain its original meaning while strictly adhering to the specified maximum depth of the dependency tree. Here, maximum depth of the dependency tree indicates the largest number of hierarchical levels from the root word (typically the main verb) to the deepest node in the sentence, indicating the most complex relationship chain in the grammatical structure. If the rewrite requires multiple sentences, each must conform to the specified maximum dependency tree depth. Phrasing or structure modifications are permissible, as long as the essential meaning is retained. Ensure each sentence rigorously conforms to the specified constraints, utilizing your extensive knowledge in linguistics, without fabricating any responses. Always output all Rewritten sentence(s) within one curly brace.

User: Input Text: Mar del Plata is an Argentine city located on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.

Input text’s Depth-Indexed Linearized Dependency Tree: [’(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, located)(3, on)(4, coast)(5, the) (5, of)(6, Ocean)(7, the) (7, Atlantic) (2, ,) (2, south)(3, 400) (3, k) (3, of)(4, Aires)(5, Buenos) (1, .)’]. The maximum depth of dependency tree of the input text, as determined by the highest value in its "Depth-Indexed Linearized Dependency Tree" is 7. The input text currently holds a maximum dependency depth of 7.

Rewrite the input text to maintain its original meaning while strictly adhering to a maximum depth of the dependency tree of exactly 5.

Always output all Rewritten sentence(s) within curly braces.

Assistant: {Mar del Plata is a city in Argentina. It is on the coast of the Atlantic Ocean, 400 km south of Buenos Aires.}

CoT ZS E-LF

System: You are an expert in text adaptation, skilled at meticulously rewriting texts to align with requested linguistic features while preserving the original meaning. Your task is to rewrite the provided Input Text to maintain its original meaning while strictly adhering to the specified maximum depth of the dependency tree. Here, maximum depth of the dependency tree indicates the largest number of hierarchical levels from the root word (typically the main verb) to the deepest node in the sentence, indicating the most complex relationship chain in the grammatical structure. If the rewrite requires multiple sentences, each must conform to the specified maximum dependency tree depth. Phrasing or structure modifications are permissible, as long as the essential meaning is retained. Ensure each sentence rigorously conforms to the specified constraints, utilizing your extensive knowledge in linguistics, without fabricating any responses. Always output all Rewritten sentence(s) within curly braces.

User: Input Text: Mar del Plata is an Argentine city located on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.

Input text’s Depth-Indexed Linearized Dependency Tree: [’(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, located)(3, on)(4, coast)(5, the) (5, of)(6, Ocean)(7, the) (7, Atlantic) (2, ,) (2, south)(3, 400) (3, k) (3, of)(4, Aires)(5, Buenos) (1, .)’]. The maximum depth of dependency tree of the input text, as determined by the highest value in its "Depth-Indexed Linearized Dependency Tree" is 7. The input text currently holds a maximum dependency depth of 7.

Rewrite the input text to maintain its original meaning while strictly adhering to a maximum depth of the dependency tree of exactly 5.

Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Let’s think through this step by step. Solve each step and explain how you arrived at your answer. Always output all the rewritten sentence(s) within one curly brace.

Continuation of CoT ZS E-LF

Assistant: To rewrite the input text while adhering to a maximum dependency tree depth of 5, we need to simplify the sentence structure. The original sentence has a depth of 7, so we will break it down into simpler sentences while retaining the original meaning.

1. Identify the main components of the sentence:

- Mar del Plata (subject)
- is (verb) - an Argentine city (predicate)
- located on the coast of the Atlantic Ocean (additional information)
- 400 km south of Buenos Aires (additional information)

2. Simplify and break down the sentence:

- Mar del Plata is an Argentine city.
- It is located on the coast of the Atlantic Ocean.
- It is 400 km south of Buenos Aires.

3. Ensure each sentence has a maximum dependency tree depth of 5:

- "Mar del Plata is an Argentine city." (Depth: 3)
- "It is located on the coast of the Atlantic Ocean." (Depth: 5)
- "It is 400 km south of Buenos Aires." (Depth: 5)

By breaking the original sentence into three simpler sentences, we ensure that each sentence adheres to the maximum depth of 5.

Here is the rewritten text:

{Mar del Plata is an Argentine city. It is located on the coast of the Atlantic Ocean. It is 400 km south of Buenos Aires.}

CoT FS E-LF

System: You are an expert in text adaptation, skilled at meticulously rewriting texts to align with requested linguistic features while preserving the original meaning. Your task is to rewrite the provided Input Text to maintain its original meaning while strictly adhering to the specified maximum depth of the dependency tree. Here, maximum depth of the dependency tree indicates the largest number of hierarchical levels from the root word (typically the main verb) to the deepest node in the sentence, indicating the most complex relationship chain in the grammatical structure. If the rewrite requires multiple sentences, each must conform to the specified maximum dependency tree depth. Phrasing or structure modifications are permissible, as long as the essential meaning is retained. Ensure each sentence rigorously conforms to the specified constraints, utilizing your extensive knowledge in linguistics, without fabricating any responses. Always output all Rewritten sentence(s) within curly braces.
User: "Input Text: 1385 - Union of Krewo - a personal union that brought Grand Duke of Lithuania Jogaila to Polish throne;
Input text's Depth-Indexed Linearized Dependency Tree: ['(0, Union)(1, 1385) (1, -) (1, of)(2, Krewo) (1, -) (1, union)(2, a) (2, personal) (2, brought)(3, that) (3, Duke)(4, Grand) (4, of)(5, Jogaila)(6, Lithuania) (3, to)(4, throne)(5, Polish) (1, ;)']. The maximum depth of dependency tree of the input text, as determined by the highest value in its "Depth-Indexed Linearized Dependency Tree" is 6. The input text currently holds a maximum dependency depth of 6.
Rewrite the input text to maintain its original meaning while strictly adhering to a maximum depth of the dependency tree of exactly 4.
Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Let's think through this step by step. Solve each step and explain how you arrived at your answer. Always output all the rewritten sentence(s) within one curly brace.

Continuation of CoT FS E-LF

Assistant: Summary of compliance with rewriting criteria:
Rewritten sentence's Depth-Indexed Linearized Dependency Tree: ['(0, Union)(1, 1385) (1, -) (1, of)(2, Krewo)(3, -) (3, union)(4, a) (4, personal) (1, ;)'].
The maximum depth of the dependency tree for the rewritten sentence, as measured by its "Depth-Indexed Linearized Dependency Tree" is 4. This meets the requested value of exactly 4.
Rewritten sentence(s): {1385 - Union of Krewo - a personal union;}
User: In-context example-2 input
Assistant: In-context example-2 output
User: In-context example input-3
Assistant: In-context example-3 input
User: In-context example-4 output
Assistant: In-context example-4 input
User: In-context example-5 output
Assistant: In-context example-5 input
User: Input Text: Mar del Plata is an Argentine city located on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.
Input text's Depth-Indexed Linearized Dependency Tree: ['(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, located)(3, on)(4, coast)(5, the) (5, of)(6, Ocean)(7, the) (7, Atlantic) (2, ,) (2, south)(3, 400) (3, k) (3, of)(4, Aires)(5, Buenos) (1, .)']. The maximum depth of dependency tree of the input text, as determined by the highest value in its "Depth-Indexed Linearized Dependency Tree" is 7. The input text currently holds a maximum dependency depth of 7.
Rewrite the input text to maintain its original meaning while strictly adhering to a maximum depth of the dependency tree of exactly 5. Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Let's think through this step by step. Solve each step and explain how you arrived at your answer. Always output all the rewritten sentence(s) within one curly brace.

Continuation of CoT FS E-LF

Assistant: Summary of compliance with rewriting criteria:
Rewritten sentence’s Depth-Indexed Linearized Dependency Tree: [’(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, on)(3, coast)(4, the) (4, of)(5, Ocean)(5, Atlantic) (2, ,) (2, south)(3, 400) (3, k) (3, of)(4, Aires)(5, Buenos) (1, .)’].
The maximum depth of the dependency tree for the rewritten sentence, as measured by its "Depth-Indexed Linearized Dependency Tree," is 5.
This meets the requested value of exactly 5.
Rewritten sentence(s): {Mar del Plata is an Argentine city on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.}

CoT+Feedback FS E-LF

System: You are an expert in text adaptation, skilled at meticulously rewriting texts to align with requested linguistic features while preserving the original meaning. Your task is to rewrite the provided Input Text to maintain its original meaning while strictly adhering to the specified maximum depth of the dependency tree. Here, maximum depth of the dependency tree indicates the largest number of hierarchical levels from the root word (typically the main verb) to the deepest node in the sentence, indicating the most complex relationship chain in the grammatical structure. If the rewrite requires multiple sentences, each must conform to the specified maximum dependency tree depth. Phrasing or structure modifications are permissible, as long as the essential meaning is retained. Ensure each sentence rigorously conforms to the specified constraints, utilizing your extensive knowledge in linguistics, without fabricating any responses. Always output all Rewritten sentence(s) within curly braces.
User: In-context example-1 input
Assistant: In-context example-1 output
User: In-context example-2 input
Assistant: In-context example-2 output
User: In-context example input-3
Assistant: In-context example-3 input
User: In-context example-4 output
Assistant: In-context example-4 input
User: In-context example-5 output
Assistant: In-context example-5 input

Continuation of CoT+Feedback FS E-LF

User: Input Text: Mar del Plata is an Argentine city located on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.

Input text’s Depth-Indexed Linearized Dependency Tree: [’(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, located)(3, on)(4, coast)(5, the) (5, of)(6, Ocean)(7, the) (7, Atlantic) (2, ,) (2, south)(3, 400) (3, k) (3, of)(4, Aires)(5, Buenos) (1, .)’]. The maximum depth of dependency tree of the input text, as determined by the highest value in its "Depth-Indexed Linearized Dependency Tree" is 7. The input text currently holds a maximum dependency depth of 7.

Rewrite the input text to maintain its original meaning while strictly adhering to a maximum depth of the dependency tree of exactly 5. Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Let’s think through this step by step. Solve each step and explain how you arrived at your answer. Always output all the rewritten sentence(s) within one curly brace.

Assistant: Summary of compliance with rewriting criteria: Rewritten sentence’s Depth-Indexed Linearized Dependency Tree: [’(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, on)(3, coast)(4, the) (4, of)(5, Ocean)(5, Atlantic) (2, ,) (2, south)(3, 400) (3, k) (3, of)(4, Aires)(5, Buenos) (1, .)’].

The maximum depth of the dependency tree for the rewritten sentence, as measured by its "Depth-Indexed Linearized Dependency Tree," is 5.

This meets the requested value of exactly 5.

Rewritten sentence(s): {Mar del Plata is an Argentine city on the coast of the Atlantic Ocean, 400 k south of Buenos Aires.}

Continuation of CoT+Feedback FS E-LF

User: Feedback: Summary of compliance with rewriting criteria: Rewritten sentence(s): Mar del Plata is an Argentine city on the coast of the Atlantic Ocean. It is 400 km south of Buenos Aires. Rewritten sentenceś maximum dependency depth = maximum value in "Depth-Indexed Linearized Dependency Tree" = max([(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, on)(3, coast)(4, the) (4, of)(5, Ocean)(6, the) (6, Atlantic) (1, .); (0, is)(1, It) (1, south)(2, km)(3, 400) (2, of)(3, Aires)(4, Buenos) (1, .)]) = 6. The rewritten sentence(s) currently hold a maximum dependency tree depth of 6.

Unfortunately, rewritten sentence(s) do not align with the requested maximum dependency tree depth of exactly 5.

To ensure clarity and meet specific requirements, kindly revise the sentence with a maximum dependency tree depth of exactly 5.

Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Always output all Rewritten sentence(s) within curly braces.

Assistant: Let's break down the task and ensure the rewritten sentence adheres to the maximum dependency tree depth of exactly 5. Rewritten sentence's Depth-Indexed Linearized Dependency Tree: ['(0, is)(1, Plata)(2, Mar) (2, del) (1, city)(2, an) (2, Argentine) (2, on)(3, coast)(4, the) (4, of)(5, Ocean)(6, the) (6, Atlantic) (1, .)', '(0, is)(1, It) (1, south)(2, km)(3, 400) (2, of)(3, Aires)(4, Buenos) (1, .)'].

The maximum depth of the dependency tree for the rewritten sentence, as measured by its "Depth-Indexed Linearized Dependency Tree," is 5.

This meets the requested value of exactly 5.

Rewritten sentence(s): {Mar del Plata is an Argentine city on the coast of the Atlantic Ocean, 400 km south of Buenos Aires.}

Template for CTG with all four linguistic features

System Prompt: You are an expert in text adaptation, skilled at meticulously rewriting texts to align with requested linguistic features while preserving the original meaning. Your task is to rewrite the provided Input Text to maintain its original meaning while strictly adhering to the specified maximum depth of the dependency tree, specified maximum dependency length, by including no more than specified number of 'difficult' words with precisely using specified number of word count.
Here,
(1) Maximum depth of the dependency tree indicates the largest number of hierarchical levels from the root word (typically the main verb) to the deepest node in the sentence, indicating the most complex relationship chain in the grammatical structure.
(2) The dependency length measures the linear distance between grammatically connected words in a sentence.
(3) 'Difficult' words are defined as those not found in the Dale-Chall list, which contains about 3000 words typically understood by 4th-grade students in the U.S.
(4) Word count is the total number of words in a sentence, defined by sequences of characters separated by whitespace.
If the rewrite requires multiple sentences, each must conform to the specified maximum dependency tree depth, maximum dependency length and word count. Also, ensure the total includes no more than the specified number of 'difficult' words. Phrasing or structure modifications are permissible, as long as the essential meaning is retained. Ensure each sentence rigorously conforms to the specified constraints, utilizing your extensive knowledge in linguistics, without fabricating any responses. Always output all Rewritten sentence(s) within curly braces.

Template for CTG with all four linguistic features

User Prompt: Input Text: {input_src}

(1) Input text's Depth-Indexed Linearized Dependency Tree: {print_dependency_tree_with_depth}. The maximum depth of dependency tree of the input text, as determined by the highest value in its "Depth-Indexed Linearized Dependency Tree" is {src_MaxDepDepth}.

(2) Input text's Length-Indexed Linearized Dependency Tree: {print_dependency_tree_with_length}. The maximum dependency length of the input text, as determined by the highest value in its "Length-Indexed Linearized Dependency Tree," is {src_MaxDepLength}.

(3) These are the 'difficult' words in the input text: {print_difficult_words_list}, totalling {src_DiffWords} 'difficult' words.

(4) Input text total words list: {print_word_list}. The number of sentences in the input text is: {no_of_sentences}. The average word count per sentence is calculated by dividing the total word count by the number of sentences in the input text, resulting {src_WordCount} words.

The input text currently holds a maximum dependency depth of {src_MaxDepDepth}, maximum dependency length of {src_MaxDepLength}, 'difficult' words of {src_DiffWords}, and word count of {src_WordCount}.

Rewrite the input text to maintain its original meaning while strictly adhering to the following criteria:

(1) The maximum depth of dependency tree of exactly {tgt_ideal_MaxDepDepth}.

(2) The maximum dependency length of exactly {tgt_ideal_MaxDepLength}.

(3) A total of no more than tgt_ideal_DiffWords 'difficult' words.

(4) An exact word count of tgt_ideal_WordCount for each rewritten sentence.

Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Let's think through this step by step. Solve each step and explain how you arrived at your answer. Always output all the rewritten sentence(s) within one curly brace.

Template for CTG with all four linguistic features

Assistant-prompt-output: "Summary of compliance with rewriting criteria:
(1) Rewritten sentence's Depth-Indexed Linearized Dependency Tree: {print_dependency_tree_with_depth}. The maximum depth of the dependency tree for the rewritten sentence, as measured by its "Depth-Indexed Linearized Dependency Tree," is {tgt_ideal_MaxDepDepth}. This meets the requested value of exactly {tgt_ideal_MaxDepDepth}.
(2) Rewritten sentence's Length-Indexed Linearized Dependency Tree: {print_dependency_tree_with_length}. The maximum dependency length for the rewritten sentence, as measured by its "Length-Indexed Linearized Dependency Tree" is {tgt_ideal_MaxDepLength}. This meets the requested value of exactly {tgt_ideal_MaxDepLength}.
(3) These are the 'difficult' words in the Rewritten sentence: {print_difficult_words_list}. The rewritten sentence contains {tgt_ideal_DiffWords} 'difficult' words, adhering to the requirement of containing no more than {tgt_ideal_DiffWords} 'difficult' words.
(4) Total words in the rewritten sentence: {print_word_list}. Number of sentences in the rewritten sentence: {no_of_sentences}. The average word count per sentence is calculated by dividing the total word count by the number of sentences in the rewritten sentence, resulting in an average of {tgt_ideal_WordCount} words per sentence. This meets the requested word count of exactly {tgt_ideal_WordCount}.
Rewritten sentence(s): {{output_text}}

Template for CTG with all four linguistic features

User-with-Feedback-Prompt Feedback: Summary of compliance with rewriting criteria:
Rewritten sentence(s): {{output_text}}
(1) Rewritten sentence’s maximum dependency depth = maximum value in ””Depth-Indexed Linearized Dependency Tree”” = max({print_dependency_tree_with_depth}) = {src_MaxDepDepth}. The rewritten sentence(s) currently hold a maximum dependency tree depth of {src_MaxDepDepth}.
(2) Rewritten sentence’s maximum dependency length = maximum value in ”Length-Indexed Linearized Dependency Tree” = max({print_dependency_tree_with_length}) = {src_MaxDepLength} The rewritten sentence(s) currently hold a maximum dependency length of exactly {src_MaxDepLength}.
(3) Identified ’difficult’ words in generated Rewritten sentence(s) = len({print_difficult_words_list}) = {src_DiffWords}. The rewritten sentence(s) currently hold total of {src_DiffWords} ’difficult’ words.
(4) Total words in the generated rewritten sentence(s): {print_word_list}. Number of sentences in the rewritten sentence: {no_of_sentences}. The average word count per sentence is calculated by dividing the total word count by the number of sentences in the rewritten sentence, resulting in an average of {src_WordCount} words per sentence.
The rewritten sentence(s) currently holds a maximum dependency depth of {src_MaxDepDepth}, maximum dependency length of {src_MaxDepLength}, ’difficult’ words of {src_DiffWords}, and word count of {src_WordCount}.
Unfortunately, rewritten sentence(s) do not align with all the requested criteria such as a maximum dependency tree depth of exactly {tgt_ideal_MaxDepDepth}, maximum dependency length of exactly {tgt_ideal_MaxDepLength}, total ’difficult’ words of {tgt_ideal_DiffWords} and word count of {src_WordCount}.

Template for CTG with all four linguistic features

To ensure clarity and meet specific requirements, kindly revise the sentence with the following criteria:
(1) The maximum depth of dependency tree of exactly {tgt_ideal_MaxDepDepth}.
(2) The maximum dependency length of exactly {tgt_ideal_MaxDepLength}.
(3) A total of no more than {tgt_ideal_DiffWords} 'difficult' words.
(4) An exact word count of {tgt_ideal_WordCount} for each rewritten sentence.
Please note, that if the initial rewrite does not meet these specific requirements, you are encouraged to modify and regenerate the output until the criteria are satisfactorily met. Always output all Rewritten sentence(s) within curly braces."