Grammar Construction Classification with LLMs

Dominik Glandorf
CPSC 588
Yale University
dominik.glandorf@yale.edu

Abstract

Large Language Models generate flawless grammar but their explicit understanding of grammar remains unclear. This work investigates the representation of pedagogically relevant grammar rules in transformer-based embeddings and the possibility to constrain text generation on grammar. The approach consists of augmenting the limited examples of a taxonomy of English grammar used by learners and training per-rule classification heads on sentences embeddings. A proof of concept details the workflow to integrate the outputs into text generation. While data augmentation and sentence classification of the augmented dataset worked well, constraining generation shows desired properties but lacks reliability. The results emphasize the need for a more diverse dataset and potentially more annotation for robust rule detection. Code available at https://github.com/dominikglandorf/LLM-grammar.

1 Introduction

Language learners benefit from reading material that aligns with their proficiency level – *graded readers*, defined as simplified literature for learners, not only adapt lexical features but their grammatical complexity (Zakaria et al., 2023). If Large Language Models (LLMs) are controllable with respect to used grammar constructions, students can receive personalized and thus more engaging but also grammatically challenging reading tasks (Mart, 2013). LLMs potentially possess internal knowledge about grammatical structures as they have been successfully used for automatic essay scoring, text simplification and detecting single grammar constructions.

However, due to the lack of labeled training data and systematic evaluation, it remains unclear to what extent LLMs can classify a comprehensive set of pedagogically relevant and teachable grammatical constructions. This study builds on an established taxonomy of grammar, which English learners practically use, and uses a state-of-the-art LLM to augment this small-scale dataset. The result is used to train grammar classifiers based on pre-trained transformer embeddings. Furthermore, it proposes how to control grammar in text generation utilizing such classifiers by doing candidate sampling and selection of entire texts.

The results are promising, although the quality loss in each step hurts the final result. When using CEFR levels as proxies for grammar complexity, the trained classifiers do reflect their differences. However, classification seems noisy, and a larger amount of manual validation is required. The work needs to be scaled up, and text generation needs to be more fine-granular to serve real-world use cases in education.

2 Related Work

2.1 Grammar in language learning

Krashen's influential ideas about language learning feature the comprehensible input hypothesis, which states that language examples understandable to the learner are essential. Though criticized

for the vagueness of the theory's predictions, the role of input is broadly accepted in the literature (Lichtman and VanPatten, 2021). The success of graded readers that dispose of reduced grammar complexity suggests the importance of controlled grammar for effective language learning and the potential impact of automatized control (Zakaria et al., 2023).

O'Keeffe and Mark (2017) compiled and published the English Grammar Profile (EGP), the only dataset known to the authors of this type, comprised of 1,222 grammar constructs that learners use on different levels, categorized by the CEFR scale, from A1 (beginner) to C2 (native). It includes a brief description in the form of a can-do statement and one to three empirical examples of each structure (see example in Appendix **??**). Such instance-based measures of grammar complexity are valuable for language teaching because each construct is teachable in contrast to traditional holistic measures of grammar complexity. Thus, this repository can be a milestone in measuring the grammar complexity of language input, especially when generating material for learners in earlier stages of their learning path.

2.2 Grammar-related tasks in NLP

LLMs are performant on holistic tasks such as automatic essay scoring (Yancey et al., 2023) and text simplification (Jeblick et al., 2023), suggesting their grasp of grammatical complexity. LLMs can also correct grammatical errors but have not made grammar constructions explicit in previous studies (Wang et al., 2021). Low-level tasks include grammar annotation, where LLMs are probed for sensitivity to single grammar structures. Weissweiler et al. (2022) used BERT sentence embeddings to classify a single grammar construction, the comparative correlative in English. Yu et al. (2023) also argued for the potential of LLMs for linguistic annotation compared to traditional NLP techniques, especially for features without a mapping to lexical forms. Their results for annotating acts of apologizing hint that LLMs can distinguish complex grammatical functions of words and are thus likely to possess grammatical knowledge. However, I have not encountered work that evaluates LLMs on grammar classification on a broad set of empirical constructions.

Controlled text generation (CTG) has developed from constrained decoding and supervised fine-tuning approaches (Zhou et al., 2023; Xiao et al., 2023) over pure prompt engineering approaches Koraishi (2023) to reinforcement learning and direct preference optimization (DPO) without a reward model just using preference data (Rafailov et al., 2023). The proposed grammar classifiers may be incorporated into all of these approaches.

3 Method

The analysis was conducted mainly with Jupyter notebooks in Python with a standard set of libraries for Machine Learning and NLP (pandas, scikit-learn, PyTorch, nltk, Spacy) on an NVIDIA RTX A5000 GPU, provided by the High-Performance Computing Cluster of Yale University. The text-generation LLMs were accessed via APIs offered by OpenAI and Google. The seed used for all random choices was 26.

3.1 Augmenting the English Grammar Profile dataset

Data was downloaded from the EnglishProfile website[1]. Its structure is characterized in Section 2.1 and did not require any preprocessing but extracting the type of construction from the guideword column and converting it into the .csv format. I used the OpenAI Chat Completion API[2] to generate more examples, comprising of positive instances of the rule and negatives that ought to have the same content without using the construct (i.e., a minimal pair). Even though the underlying model is closed-source, it is known that their transformer models are instruction-tuned and improved by Reinforcement Learning from Human Feedback (Ouyang et al., 2022). I tested three configurations of prompting: With the model checkpoint `gpt-3.5-turbo-0613`, I used few-shot in-context learning and chain-of-thought-like (CoT) prompting. With `gpt-4-0613`, I used only the first strategy due to costs. The system message for both strategies was "You are an English as a foreign language teacher who is knowledgeable about grammar." The in-context learning prompt uses a template to describe the grammar rule, appends the available examples, and specifies the output format as a list. After

[1] `https://www.englishprofile.org/english-grammar-profile/egp-online`
[2] `https://platform.openai.com/docs/models/overview`

the list of positive examples was returned, a second prompt was made that asked for rewriting every positive example without using the construction. The CoT prompts led the model through the process of first understanding and explaining the rule and examples and then applying it. All prompts are included in Appendix B.

For 12 randomly selected constructions (two per difficulty level), each configuration was used to create 20 positive and 20 negative examples, resulting in 1,488 examples (including 48 extra examples due to the model not sticking to the requested format), that I manually evaluated in a blinded manner, i.e., without knowing the configuration or positivity of each sample. Note that I have not had any special training as a linguist or in English grammar. The data quality was assessed by the correctness per configuration and type of instance, calculated as

$$C_{c,p} = \frac{1}{|r_{c,p}|} \sum_{i=1}^{|r_{c,p}|} \mathbb{1}[r_{c,p,i} = p]$$

where c is the configuration, $p = 1$ if it was a positive instance, $p = 0$ if negative, $r_{c,p,i}$ the rating for a specific instance i in this group.

For the following step, a larger number of examples per construction was required. I used the more affordable `gpt-3.5-turbo-0613` to create 500 positive and 500 negative examples in batches of 20 (to keep the required context small) for three randomly selected constructions of each combination of difficulty and type (FORM, USE, FORM/USE), resulting in 53 constructions (one combination only had two constructions). For an automatic quality assessment, the average ratio of unique sentences per construction was calculated for positive and negative examples. Furthermore, the average cosine similarity of embeddings between all positive example sentences and between all negative sentences was calculated per construction and compared to the baseline within the Brown corpus (Kučera et al., 1967). The formula of this metric is

$$CosineSimilarity(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

3.2 Training grammar classifiers

I used the embedding model `ember-v1` by llmrails[3] via the SentenceTransformers library to compute sequence embeddings for each example in the dataset from the previous step. `ember-v1` reached state-of-the-art performance on the Massive Text Embedding Benchmark on Huggingface[4] in the category Classification at the time of this study. The model supports sequence lengths of up to 512 tokens, which is sufficient for typical sentences, and outputs 1024-dimensional embeddings. I used two-layer neural networks with the ReLU activation function and a sigmoidal output function as binary classification heads to classify the presence of single EGP constructions based on the sentence embedding. Each network had a hidden dimension of 32, resulting in 32,833 trainable parameters.

Due to the quality of the negative examples in GPT3.5, three-quarters of the generated negative instances were replaced by positive examples from other rules, assuming that these will usually not use the same rule. This also aimed at diversifying the training data. The batch size was 64, the learning rate for the Adam optimizer was 0.0001, and training was stopped after a maximum of 200 epochs. As soon as the validation loss did not improve for ten consecutive epochs, training was stopped to prevent overfitting. Those hyperparameters were chosen manually by iteratively adapting them to achieve a smooth training loss curve that steadily improves. This strategy ensured that the model capacity was large enough to reach maximum performance and prevent overfitting.

Due to the limited training data size, 5-fold cross-validation was used. The evaluation metrics were the average of accuracy, recall, and precision across the resulting five test sets, calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

[3] `https://huggingface.co/llmrails/ember-v1`

[4] `https://huggingface.co/spaces/mteb/leaderboard`

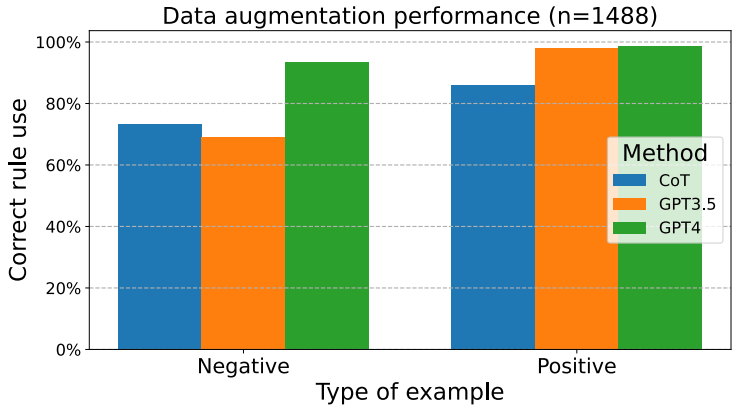

Figure 1: Amount of correctly generated instances by strategy and type of example. CoT: Chain-of-thought prompting with GPT3.5.

For the next step, each construction classifier was trained on the entire dataset, and its checkpoint was saved as a .pth file. The number of epochs was set to 60 based on the observation that the training usually stopped early at this point.

3.3 Grammar-controlled text generation

The goal of this step was to use the trained classifiers to judge model output in terms of present grammar constructions and control it. As a baseline, a CEFR-labeled dataset was used that was compiled from online resources[5]. The classifier scores grouped by level should roughly correlate with the respective text level. Note that a perfect match is unrealistic since there are always less difficult (and some more difficult) constructions in texts of all levels, but a trend should be visible. To avoid potential confounding with the data augmentation step using OpenAI models, Google's Gemini model was prompted to write a story of a certain difficulty level (with an explanation from the CEFR) about a topic given in a prompt. The prompt comprised the first 100 characters from 50 random texts in the CEFR dataset to set different topics of the stories (see an example in Appendix B). In the last step, five candidate texts were sampled for each level and 15 topics, and as a heuristic, the one with the highest score when summing across all classifiers that detected constructions of the requested levels was chosen.

4 Results

4.1 Data Augmentation

Figure 1 summarizes the manually evaluated quality of the three data augmentation strategies. GPT4 and GPT3.5 perform outstanding for positive instances when directly instructed to create more examples. When asking for examples without using a given rule, only GPT4 performs reliably. The Chain-of-Thought-like prompting strategy negatively impacted the performance for positive examples. In the case of negative examples, it is possible that it may help the LLM to understand the task slightly better. Overall, the evaluation confirms the ability of modern LLMs to augment a dataset from a class description and a few examples.

The larger augmented dataset created for the next step (in total, n=53,000) only had an average of 7.2% of duplicated positive examples across all constructions. This amount exceeded 20% for four constructions that were all on the beginner level. The results for the average cosine similarity between sentence embeddings are in Appendix C. The positive and negative examples are usually less diverse than the baseline corpus. Mixing positive examples from other constructions into the negative examples decreases the average cosine similarity almost to the baseline value and is hence preferred in the next step.

[5] https://www.kaggle.com/datasets/amontgomerie/cefr-levelled-english-texts

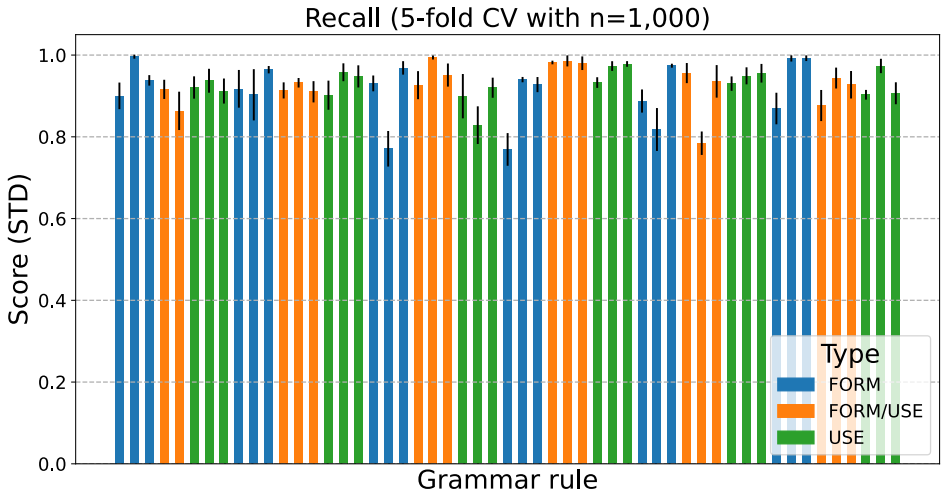

Figure 2: Recall of the grammar construction classifiers for 53 constructions. The difficulty increases from left to right (8-9 bars per level).

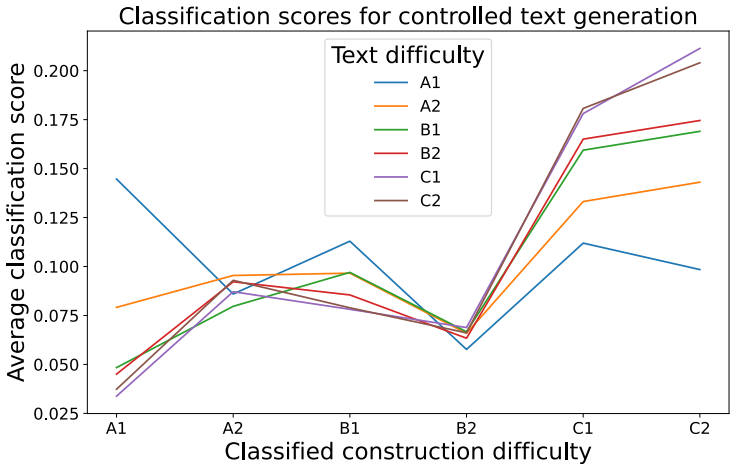

Figure 3: Distribution of grammar classification scores of different complexities across 90 grammar-controlled texts.

4.2 Grammar Classification

Figure 2 depicts the quality of detecting whether a given grammar construction is present in a sentence (recall). Accuracy and precision show a similar picture, with precision usually being slightly weaker than recall. They are plotted in Appendix D. All metrics are clearly above random chance, although not perfect. Recall attains at least 80% for all but five constructions, which is important since detecting a used rule is more crucial than mistakenly predicting it. Remarkably, among these five constructions, three concern the Form, which points out that transformers may better represent the semantics (Use) than the pure grammatical forms in sentences. Note that the performance of this step is upper bound by the data quality of the prior step.

4.3 Text Generation

The results for the level-conformance of the baseline and the prompt-controlled generated output are in Appendix E. Based on the classifiers from the previous step, no text level in the baseline corpus uses exclusively grammar constructions from only one level. However, a general trend is visible: the scores for A1 constructions are declining, while the scores of C1 and C2 constructions are higher

within the most difficult texts. Remarkably, the B2 classifiers generally have the lowest scores across all text difficulties. For the texts generated by Gemini, a similar pattern emerges; it is even more pronounced. This confirms that the generated texts use different sets of grammar rules on different difficulty levels. Figure 3 shows the grouped classification scores for the chosen candidates with the highest scores. The heuristic used seems to impact the scores across the different classifiers only minimally compared to the prompt-only strategy.

5 Discussion

This work has shown how LLMs can be potentially controlled to serve pedagogical use cases better, such as grammar teaching. The data augmentation of the English Grammar Profile worked well, especially using the most expensive closed-source model, GPT4. Nevertheless, the quality of positive examples generated by GPT3.5 could almost keep up with the flagship model. The grammar construction classification, based on the dataset augmented with GPT3.5 for cost reasons, was not trivial to evaluate due to the non-perfect training data. The results strongly confirmed above-chance-level performance and a satisfying recall for most of the 53 tested constructions. The controlled text generation did not improve much over the prompt-based text generation.

5.1 Limitations

As the diversity analysis has shown, the data augmentation is not as trivial as the main results may imply. For the simplest constructions, such as the construction "DETERMINER + NOUN" it is difficult to construct negative examples because almost all correct English sentences include it. Luckily, the classifiers for level A1 may not be necessary when only the complexity of sentences matters, as this level can just be inferred as the inverse from the remaining levels. The grammar classifiers are also imperfect and may need additional annotation for reliable results. At least, it seems that, in general, there is some correspondence between the grammar complexity and the classification scores. In the text generation step, it would be better to control the output sentence by sentence. Unfortunately, the APIs of these models do not support this kind of decoding anymore. In general, more manual validation of the results of each step would make this work more conclusive. Unfortunately, as this was a solo project after the project partner dropped out, the bandwidth for manual coding was even more limited.

5.2 Ethical considerations

The EGP dataset was expert-curated to describe the use of language in the learning process empirically. It includes examples from real students and can hence reflect biases in their opinions. Instructing the LLM to focus on grammatical structure instead of content hopefully mitigates biases in the augmented data set, but it is not guaranteed. In the worst case, the grammar classification works better for those topics that are more interesting or helpful to only a certain student subgroup. These biases can arise from the initial choice of native speakers that were included in the examples of this reference. The authors acknowledged the critique of the eurocentrism of the CEFR classification for learner levels, which is underlying the dataset and this work (O'Keeffe and Mark, 2017).

Another issue which is related to the use of Large Language Models is a potentially toxic or biased language that is especially sensitive when underage students are working with an LLM-based language learning tool (Meyer et al., 2023). On the pedagogical side, it may also be questionable how authentic artificially generated text may be and if a lack of authenticity hinders language development. Interacting with a machine instead of humans for language acquisition can also have negative effects.

5.3 Future Work

The goal should be to scale the classification to all 1,222 constructions in the EGP to have a more comprehensive grammar assessment. Moreover, the grammar constructions could be further identified within the sentences to increase the detection quality and enable annotations. Apart from that, a user study to see the impact on the learning progress is necessary.

References

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Oliver Sabel, Jens Ricke, and Michael Ingrisch. 2023. ChatGPT makes medicine easy to swallow: an exploratory case study on simplified radiology reports. *European Radiology*.

Osama Koraishi. 2023. Teaching English in the age of AI: Embracing ChatGPT to optimize EFL materials and assessment. *Language Education and Technology*, 3(1).

Henry Kučera, Winthrop Francis, William Freeman Twaddell, Mary Lois Marckworth, Laura M. Bell, and John Bissell Carroll. 1967. Computational analysis of present-day American English. *(No Title)*.

Karen Lichtman and Bill VanPatten. 2021. Was Krashen right? Forty years later. *Foreign Language Annals*, 54(2):283–305. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/flan.12552.

Çağri Tuğrul Mart. 2013. Teaching grammar in context: why and how? *Theory & Practice in Language Studies*, 3(1).

Jesse G. Meyer, Ryan J. Urbanowicz, Patrick C. N. Martin, Karen O'Connor, Ruowang Li, Pei-Chen Peng, Tiffani J. Bright, Nicholas Tatonetti, Kyoung Jae Won, Graciela Gonzalez-Hernandez, and Jason H. Moore. 2023. ChatGPT and large language models in academia: opportunities and challenges. *BioData Mining*, 16(1):20.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. ArXiv:2203.02155 [cs].

Anne O'Keeffe and Geraldine Mark. 2017. The English Grammar Profile of learner competence: Methodology and key findings. *International Journal of Corpus Linguistics*, pages 457–489.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. ArXiv:2305.18290 [cs].

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A Comprehensive Survey of Grammatical Error Correction. *ACM Transactions on Intelligent Systems and Technology*, 12(5):1–51.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your Syntax, the better your Semantics? Probing Pretrained Language Models for the English Comparative Correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Changrong Xiao, Sean Xin Xu, Kunpeng Zhang, Yufang Wang, and Lei Xia. 2023. Evaluating Reading Comprehension Exercises Generated by LLMs: A Showcase of ChatGPT in Education Applications. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 610–625.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating Short L2 Essays on the CEFR Scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 576–584, Toronto, Canada. Association for Computational Linguistics.

Danni Yu, Luyang Li, Hang Su, and Matteo Fuoli. 2023. Assessing the potential of AI-assisted pragmatic annotation: The case of apologies.

Azrifah Zakaria, Willy A Renandya, and Vahid Aryadoust. 2023. A Corpus Study of Language Simplification and Grammar in Graded Readers. 16(2).

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023. Controlled text generation with natural language instructions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pages 42602–42613, Honolulu, Hawaii, USA. JMLR.org.

A English Grammar Profile Example Construction

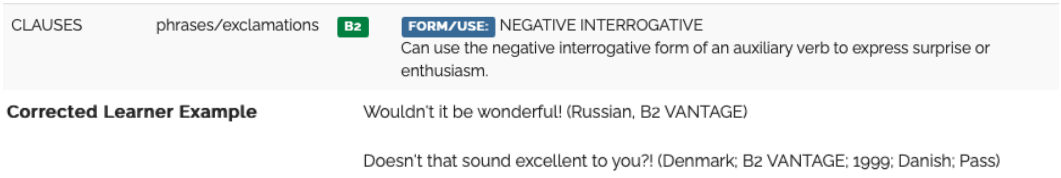

Figure 4: Grammar construction on level B2 with examples in the English Grammar Profile.

B Prompts

Those are the two in-context learning prompts:

1. Prompt:

```
Create {num_examples} more examples for the grammatical construction on
CEFR level {Level} in the category "{SuperCategory}: {SubCategory}" with
guideword "{guideword}" and the rule: {Can-do statement}

Examples:
{Example}

Output format:
1. [EXAMPLE 1]
2. [EXAMPLE 2]
```

2. Prompt:

```
Rewrite every example with the same content but without using the rule.
```

Those are the four CoT prompts, fed into the model in this order:

1. Prompt:

```
Learn to analyse the grammatical construction on CEFR level
{construction["Level"]} in the category "{construction["SuperCategory"]}:
{construction["SubCategory"]}" with guideword "{construction["guideword"]}"
and the rule: "{construction["Can-do statement"]}"

Examples:
{construction["Example"]}

Task:
Explain how each example uses the rule.
```

2. Prompt:

```
Now rewrite each example avoiding this rule but maintaining the content.
```

3. Prompt:

```
Now write {NUM_EXAMPLES} more examples using the rule. Output format:
1. [EXAMPLE 1]
2. [EXAMPLE 2]
```

4. Prompt:

```
Rewrite each example with the same content but without using the rule. Output format:
1. [EXAMPLE 1]
2. [EXAMPLE 2]
```

This is an example text generation prompt to Gemini:

```
Write a story using the following prompt on CEFR level C2 (Description: Can understand
and interpret critically virtually all forms of the written language including abstract,
structurally complex, or highly colloquial literary and non-literary writings. Can
understand a wide range of long and complex texts, appreciating subtle distinctions
of style and implicit as well as explicit meaning.)

Five years ago, telling a friend that you interact regularly with a verbose orange
traffic cone migh...
```

C Diversity of augmented dataset

Table 1: Average cosine similarities between sentences in real text (Brown corpus) and the positive and negative examples generated by GPT3.5

Corpus	Mean Cosine Similarity	Std. Dev.
Brown	0.334	0.002
Positive	0.462	0.052
Negative	0.451	0.045
Negative with random other positives	0.369	0.007

D Classification performance

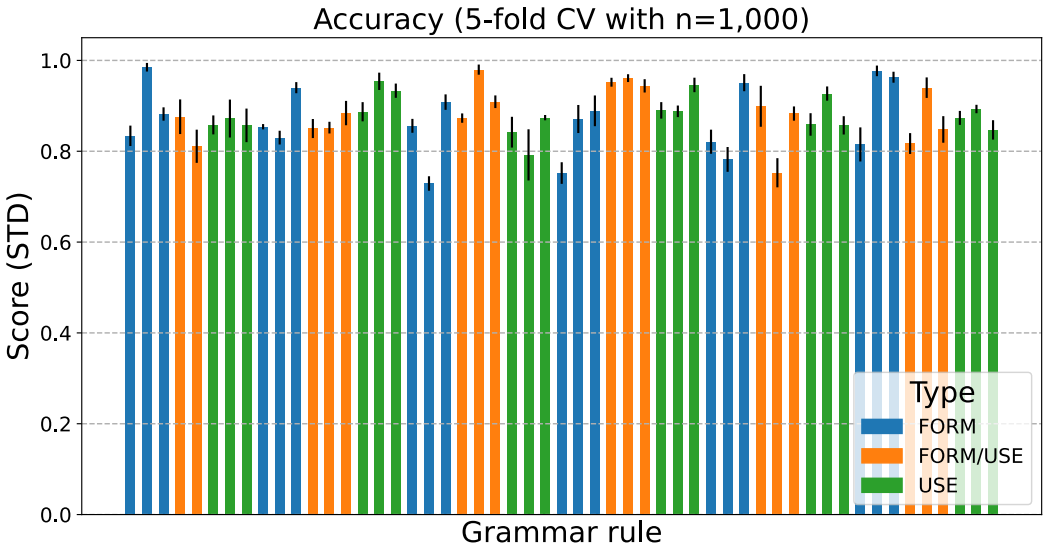

Figure 5: Accuracy of the grammar construction classifiers for each construction. The difficulty is increasing from left to right.

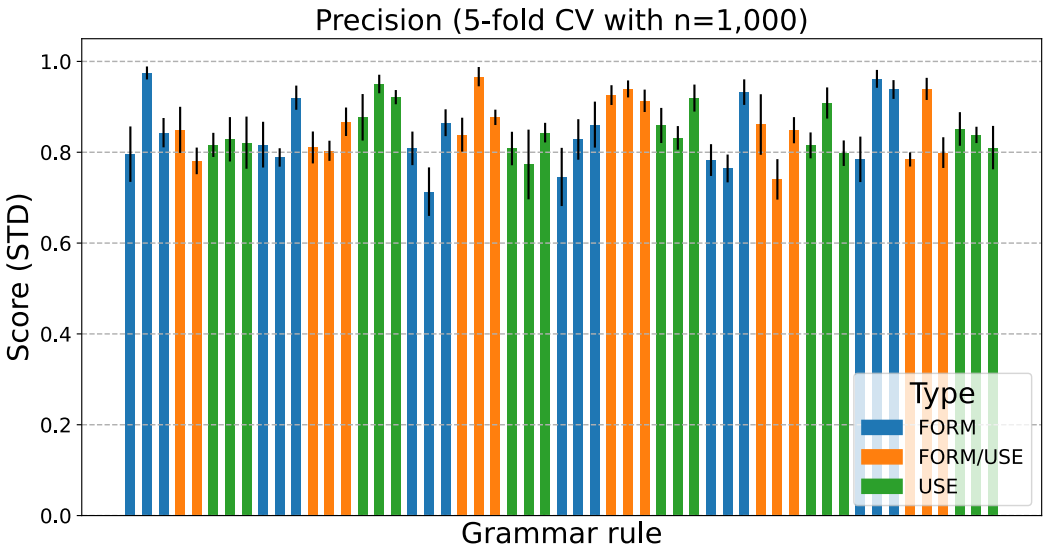

Figure 6: Precision of the grammar construction classifiers for each construction. The difficulty is increasing from left to right.

E Text generation performance

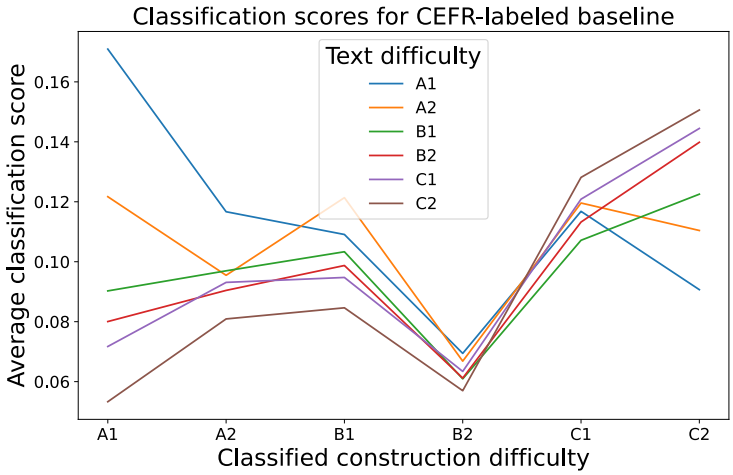

Figure 7: Distribution of grammar classification scores of different complexities across 1,493 texts from the CEFR-labeled dataset.

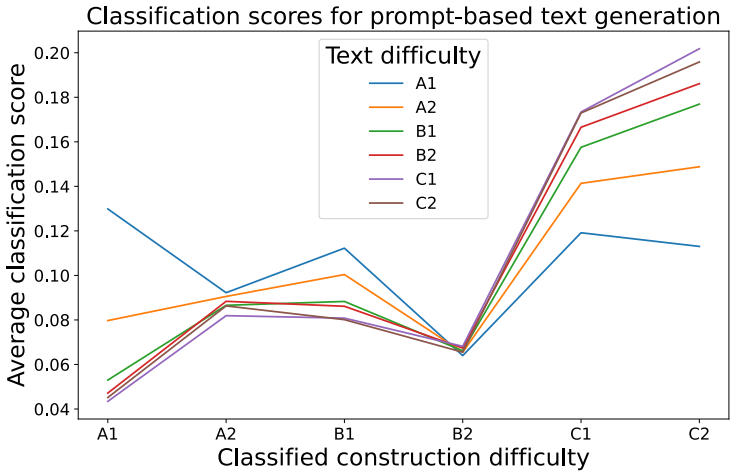

Figure 8: Distribution of grammar classification scores of different complexities across 300 texts generated by Gemini about 50 different topics on 6 difficulty levels.

Reproducibility checklist:

Mathematical Setting, Algorithm, and Model Description:

- ☐ **not applicable** Include a thorough explanation of the mathematical framework, algorithmic approach, and the model's architecture.
- ☑ Ensure clarity in the methodology and theoretical underpinnings, as needed.

Source Code Accessibility:

- ☑ Provide a link to the source code on github.
- ☑ Ensure the code is well-documented
- ☑ Ensure that the github repo has instructions for setting up the experimental environment.
- ☑ Clearly list all dependencies and external libraries used, along with their versions.The seed used for all random choices was 26.

Computing Infrastructure:

- ☑ Detail the computing environment, including hardware (GPUs, CPUs) and software (operating system, machine learning frameworks) specifications used for your results
- ☑ Mention any specific configurations or optimizations used.

Dataset Description:

- ☑ Clearly describe the datasets used, including sources, preprocessing steps, and any modifications.
- ☑ If possible, provide links to the datasets or instructions on how to obtain them.

Hyperparameters and Tuning Process:

- ☑ Detail the hyperparameters used and the process for selecting them, including any search strategies like grid or random search.
- ☑ Provide rationale for hyperparameter choices, if applicable.

Evaluation Metrics and Statistical Methods:

- ☑ Clearly define the evaluation metrics and statistical methods used in assessing the model.
- ☑ Include details on how these metrics are calculated.

Experimental Results:

- ☑ Present a comprehensive set of results, including performance on test sets and/or any relevant validation sets.
- ☐ **not applicable** Include comparisons with baseline models and state-of-the-art, where applicable.

Random Seed Reporting:

☑ If applicable, state the random seeds used in experiments to ensure reproducibility of results.

Ethical Considerations and Limitations:

☑ Discuss any ethical considerations related to the dataset or model use.

☑ Clearly state the limitations of your approach and potential areas for future work.