Enhancing Grammar Skills with ARES: A Pedagogically Oriented AI-Assisted Extension

Megan Horikawa
6196282
6 ects - ungraded

Ekaterina Nikitina
6365479
6 ects - ungraded

Adrianna Paszkowska
6133209
6 ects - graded

Shahrokh Ahmadinasab Omran
6891970
6 ects - graded

Abstract

This paper presents an intelligent computer-assisted language learning (ICALL) extension for the Annotated Reading Enhancement System (ARES) focused on grammar practice. We developed a servlet using Groq's LLM API to generate fill-in-the-blank exercises. To optimize prompting, we evaluated three strategies–zero-shot, with descriptions, and with descriptions and examples across CEFR levels. Results showed that the zero-shot prompt with the DeepSeek model yielded the most accurate and preferred outputs. Future work includes expanding question types, integrating automated feedback, and incorporating user proficiency information to adapt tasks to learners' difficulty levels.

1 Introduction

Reading is a fundamental skill for academic success and lifelong learning. According to (Grellet, 1981), reading typically serves two purposes: for pleasure or to acquire information. It plays a critical role in learners' educational development and beyond (Küçükoglu, 2013). Moreover, reading comprehension is closely connected to learners' understanding of language structure and their ability to make metalinguistic interpretations(Marjokorpi, 2024).

Research in second language (L2) acquisition has highlighted the contribution of grammar knowledge to L2 reading comprehension, beyond vocabulary alone (Jung, 2009). However, traditional classroom environments often face challenges in providing personalized and interactive reading experiences due to time constraints and student diversity (Verhoeven et al., 2011).

Intelligent computer-assisted language learning (ICALL) systems have shown promise in addressing challenges by enhancing student engagement, supporting language development, and offering automated feedback (Amaral and Meurers, 2011). While many existing systems focus primarily on vocabulary support, few have successfully integrated LLMs to deliver adaptive, real-time grammar exercises and personalized feedback (Seßler et al., 2025).

One of the most recent initiatives in this area is the ARES project(Lee et al., 2024). ARES is a pedagogically oriented, web-based ICALL system designed to enhance the L2 reading experience through interactive and customization tools. This project extends ARES with a grammar practice servlet that allows learners to generate targeted grammar exercises based on CEFR level and specific grammatical constructs. This extension facilitates students in engaging in self-paced, focused practice whenever they require additional reinforcement of grammar concepts.

2 Background

Large Language Models (LLMs) have significantly impacted a variety of domains, including education. Their ability to process and generate coherent, contextually appropriate text has led to innovative approaches in language instruction and learning. In recent years, the application of LLMs in educational contexts has become a rapidly growing area of interest. For example, Vrdoljak et al. (2025) examined how LLMs can be used to tailor educational content to individual learners in medical education, demonstrating that AI-assisted instruction can enhance the learning process. Similarly, Jia et al. (2025) explored the use of LLM-based classroom assistants and found that AI-supported discourse improved student engagement and comprehension.

This project investigates how LLMs can support English language acquisition, particularly in fostering grammar awareness and production skills. The ARES system (Lee et al., 2024), a central tool in this study, leverages LLM capabilities to annotate texts with grammatical constructs and provide learners with explanations and targeted prompts.

The system supports various prompt types, including grammar explanation requests, comprehension questions based on a text, and feedback, all designed to promote active learner engagement. We will describe the ARES system in more detail in the section below.

A key pedagogical challenge in grammar instruction is supporting learners' transition from **declarative knowledge** - the ability to recognize a grammatical structure and understand its rules- to **procedural knowledge** -the ability to use them accurately in writing or speech (DeKeyser, 2014). This transition is critical for long-term retention and language development. As demonstrated by Malmir and Parhizkari (2021), structured written exercises-particularly sentence writing and fill-in-the-blank tasks-are effective in reinforcing grammatical collocations and facilitating the transition from passive recognition to active use. In this context, integrating such practice into an LLM-based learning environment may enhance learners' grammatical accuracy and overall language proficiency.

2.1 ARES System

The ARES (Annotated Reading Enhancement System) is a web-based Intelligent Computer-Assisted Language Learning (ICALL) tool designed to enhance L2 English reading comprehension.

ARES is built on a Java backend deployed in a Jetty server. For the display layer, it uses the Bootstrap framework. In order to enable Learning Analytics, all user activities such as button clicks, grammar and vocabulary searches, reading comprehension question attempts, assignment submissions, viewing of feedback messages, and any other relevant user actions are logged through xAPI and stored in a Learning Record Store (LRS) (Lee et al., 2024).

ARES integrates Natural Language Processing (NLP) tools and LLMs to provide interactive reading support. Each text is run through an NLP pipeline to identify and annotate different vocabulary and grammar constructions. A learner can then click on any word within the text for an explanation of any unfamiliar forms, as shown in Figure **??**. These explanations are designed exclusively towards English learners and include the CEFR level, a description of usage, and examples of the form used in context. On the teacher side, ARES includes a question generation feature that allows teachers to automatically generate comprehension questions based on a given text to test students' comprehension.

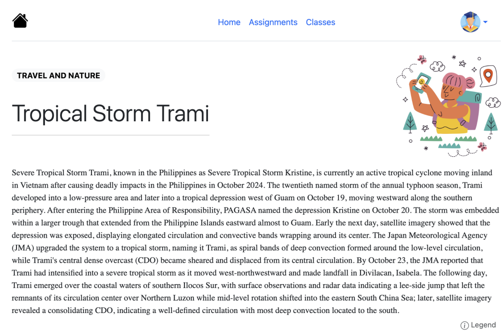

Figure 1: A screenshot of a text included in the ARES system.

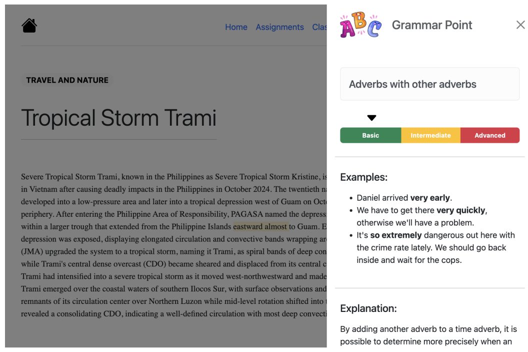

Figure 2: A screenshot of the grammar form explanation included in the ARES system. A grammar construct from the text is selected, and a popup with the grammar explanation appears in a menu on the right.

While ARES offers adaptive and interactive support for reading comprehension, it currently lacks a component for students to practice their skills independently. To address this gap, we introduce an extension to the ARES system that incorporates interactive grammar exercises. This servlet-based extension aims to provide learners with opportunities to engage in self-guided practice, reinforcing their understanding and facilitating the transition from grammar awareness to accurate and fluent use.

3 Methodology

3.1 Prompting techniques

In our system, prompting plays an essential part in shaping the output of the language model, as it is the primary means to control both the format and content of the generated questions. Designing an effective prompt was therefore a critical first step in our development workflow.

We initially explored a range of popular prompting strategies and conducted informal comparisons of the outputs of fill-in-the-blank sentences. Based on these results, we selected three prompting approaches for more detailed and documented evaluation: zero-shot prompting (Wu et al., 2022), zero-shot with descriptions, zero-shot with example sentences.

To assess these strategies, we selected several grammatical constructs across all CEFR proficiency levels. Specifically, we randomly chose 3 constructs for levels A1 and C1 and 6 constructs each for the remaining levels. For each construct, the language model was prompted to generate fill-in-the-blank grammar questions using one of the three strategies.

Each prompt contained four core parameters: the number of questions to generate, the name of the grammatical construct, the CEFR level, and the distribution of sentence types (neutral, negative, and interrogative). We found that explicitly specifying the number of questions for each sentence type yielded better results than using generic instructions such as "Ensure that the questions provide students with opportunities to practice the topic from different perspectives." This helped ensure that the questions reflected varied syntactic structures and were pedagogically useful. sample prompt is provided in Appendix B.

The zero-shot with examples approach included two usage examples of the target construct in context, while the zero-shot with description approach offered a brief explanation of the construct to reduce ambiguity. Both the descriptions and examples were extracted from a custom JSON database built using information from the English Grammar Profile (EGP) (O'Keeffe and Mark, 2017).

3.2 Selection of the Language Model

The first challenge during our prompt evaluation process was selecting an appropriate open-source model from those available through the Groq API. Alongside DeepSeek-R1 Distill LLaMA-70b (DeepSeek-AI et al., 2025), which ultimately chosen for our implementation, We also tested LLaMA 3.3-70B (Grattafiori et al., 2024) and Mixtral 8 7B (Jiang et al., 2024). During testing, both Mixtral and LLaMA models struggled to interpret grammatical construct names, often resulting in tokenization errors or incomplete outputs. These issues were more prevalent at higher CEFR levels, likely due to the increased semantic ambiguity of advanced grammar constructs.

A potential reason for these difficulties lies in the models' training approaches. Unlike LLaMA and Mixtral, which primarily focuses on causal language modeling, DeepSeek incorporates reinforcement learning and structured reasoning tasks during training. These features may have enabled DeepSeek to better interpret prompts containing complex grammar-related metadata, such as sentence types (neutral, negative, interrogative) or elaborate construct names.

A potential reason why we had difficulties during tokenization and parsing with LLaMA and Mixtral, but not with DeepSeek, might be the reinforcement learning and its reasoning strategies which have been used during training for the DeepSeek model. While all of these models are transformer-based, DeepSeek performs better, possibly because it was trained on structured data and reasoning tasks, instead of focusing mainly on causal language modeling like the other two models.

Furthermore, we observed that the LLaMA model, appeared particularly sensitive to ambiguity and grammatically incorrect input data. This is crucial in our case because grammatical constructs (e.g., "PAST SIMPLE FOR EVERYDAY EVENTS AND STATES" (level A1) or "ADVERBS AS MODIFIERS OF TIME" (level B1)) can be interpreted in multiple ways, especially when another required parameter "sentence type" (negative, neutral, interrogative) was added.

In contrast, DeepSeek not only succeeded in generating relevant questions across all proficiency levels but also enhanced the generated sentences by adding additional features, even if they were not explicitly stated in the given prompt. For instance, in beginner levels such as A1 or A2, the model often included the base verb that needed to be modified in parentheses.

Thus, while the task of generating simple sentence structures may not appear to require complex reasoning, our findings suggest that reasoning-enhanced models like DeepSeek may be more suited for generating grammar practice questions compared to other models.

3.3 Validation Results and Challenges

The main criteria used for our evaluation were: (1) whether the generated text correctly used the given grammatical construct in a sentence context, given its type (neutral, negative, interrogative), and

		Percentage of questions that match the grammatical form	Percentage of questions with errors	Preference
A1	Zero-shot	**100%**	8%	66%
A1	Zero-shot with description	33%	**10%**	**33%**
A1	Zero-shot with examples	76%	0%	
A2	Zero-shot	**93%**	**30%**	**80%**
A2	Zero-shot with description	69%	11%	20%
A2	Zero-shot with examples	57%	11%	
B1	Zero-shot	61%	9%	50%
B1	Zero-shot with description	68%	13%	
B1	Zero-shot with examples	**75%**	**17%**	50%
B2	Zero-shot	73%	40%	25%
B2	Zero-shot with description	43%	**52%**	25%
B2	Zero-shot with examples	**86%**	18%	**50%**
C1	Zero-shot	**61%**	10%	50%
C1	Zero-shot with description	20%	25%	
C1	Zero-shot with examples	28%	22%	50%

Table 1: Results of prompt evaluation

(2) whether the sentence contained grammatical or structural errors. Since we used human evaluation to determine the best prompting strategy, 'personal preference' was added as an additional criterion for our evaluation. All criteria were assessed using binary judgments, and the results are presented as percentages for easier summarization, as can be seen in Table 1.

One challenge encountered during our evaluation was that, in several cases-and occasionally across all examples within a given prompt-the model generated questions that contained the answer directly within the sentence. This issue influenced the results significantly for some prompt types, particularly in terms of sentence validity and usefulness for learner practice.

The results indicated that zero-shot prompting—without examples or description—produced the most preferred exercises across the majority of CEFR levels. Furthermore, combining these results with the percentage of questions that contained errors suggests that zero-shot prompts seem to be optimal for this type of generation with the DeepSeek model.

These results were unexpected, knowing that ambiguity of a short construct name was the main challenge encountered during model selection-we initially expected that the zero-shot with description prompting strategy would be the most effective. This however was not the case, as all 3 prompting types had similar percentages of correct and incorrect sentences for different constructs. Our choice was mostly influenced by subjective preferences or by significant errors in formatting for some of the prompts.

4 Servlet Architecture

We developed a Java-based servlet designed to generate fill-in-the-blank grammar practice questions for English learners by leveraging Groq's LLM API. The servlet is deployed on a Jetty server running within a Docker container, allowing for containerized management and easy integration within the existing ARES system.

The servlet is registered with the /question/generation endpoint and responds to HTTP POST requests. Requests are sent in JSON format, where the servlet parses the input parameters and constructs a request to the Groq API for question generation. The servlet accepts three parameters: **Grammar Construct**, **Number of Questions**, and **CEFR level**. The number of questions is the only parameter directly input by the user, whereas the CEFR level and construct name are retrieved from the grammar lookup page pending full integration with ARES[1].

The parameters are passed into the prompt which is then assembled into a request body. The request body is then converted into JSON format and sent to Groq's API using OkHttp client for question generation.

Through experimentation, we found that requesting a large number of practice questions from the LLM resulted in decreased sentence uniqueness and increased repetition. To address this, we have implemented a limit of 20 questions per request to reduce redundancy and prevent excessive computational load on the LLM.

Upon receiving a response from Groq, the servlet processes the output and returns it in JSON for-

[1]CEFR level and construct name are manually provided for testing purposes.

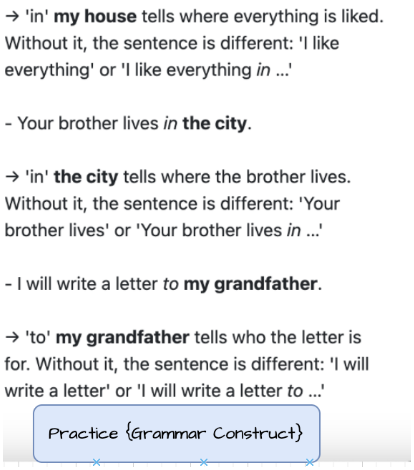

Figure 3: A sketch of how a learner could access the question generation function within the ARES system. The {Grammar Construct} field would include the construct name.

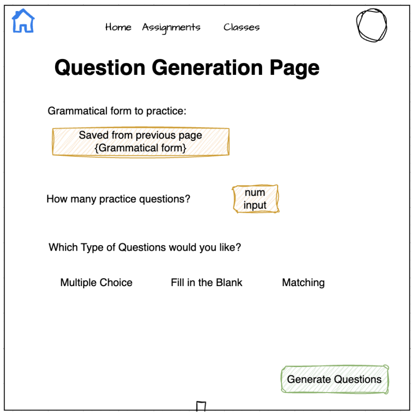

Figure 4: A sketch of the imagined popup to set parameters for question generation. This sketch also accounts for different question types which is not yet implemented in our servlet.

mat. This response will later be formatted to be displayed on the front end, allowing the users to type answers to the generated questions.

4.1 Imagined Implementation

Once integrated into the ARES system, users can access the question generation feature via the grammar lookup tool. As illustrated in Figure 3, a button at the bottom of the grammar explanation page will lead to a popup (Figure 4) where users can specify the number of questions they wish to generate for a particular grammar construct.

In the future, we aim to expand this feature by incorporating additional question formats such as multiple choice and matching exercises. These enhancements will provide learners with a more diverse range of practice opportunities and better align with varied learning preferences.

Our trials prompting the LLM for question generation revealed that certain grammar constructs may not be well-suited for specific question types, highlighting the need for careful prompt engineering and construct-specific question design.

Additionally, we plan to implement an automated feedback feature that provides learners with immediate explanations and corrections. This will allow students to receive guidance on their responses, reinforcing learning and helping them understand the grammatical concepts more effectively.

Further logging of generated questions and student performance could also be leveraged to create more individualized exercises, allowing the LLM to target each student's areas for improvement more effectively.

4.2 Limitations

While LLMs offer valuable applications for both students and teachers, they come with several limitations that require careful consideration when deploying systems reliant on their outputs. One major challenge is that these models are maintained by external companies, which frequently update them without transparency. These updates can alter the model's behavior, potentially affecting the consistency of generated outputs. As a result, prompts may need frequent adjustments to maintain structured and reliable responses.

Another challenge is the inaccuracy of model outputs, as neither their quality nor appropriateness can be guaranteed. During our prompt validation, we observed that many models struggled to consistently output well-formatted JSON. Additionally, frequent errors occurred in the generated questions, such as cases where the answer was mistakenly included within the generated question itself. Given the current architecture of the servlet, these generated questions will be directly accessible to students without prior vetting, raising concerns about the reliability and appropriateness of the content.

Due to time constraints, this project primarily focused on the back-end architecture. As such, no front-end interface was developed, which limits user interaction to test the system capabilities (e.g. see questions from LLM and try to solve them). Currently, the functionality can only be evaluated via API testing tools such as Postman.

4.3 API Testing using Postman

Postman(Postman) is a popular platform widely used by developers for API development and testing, enabling automated regression tests through collections and environments as demonstrated in (Postman Learning Center). As a means to validate our API development, we implemented a collection of automated tests using Postman. It targets the POST/question/generation endpoint for expected and edge case scenarios.

The API accepts a JSON request body containing our prompt on a local development server (localhost). The response content is then parsed to validate the correctness.

The main test scenarios assess:

- That the API returns a 200 OK response.
- The response is in valid JSON format.
- The presence of the expected nested content field (choices[0].message.content).
- That the returned questions array has the correct length.
- Each question includes both a nonempty question and an answer string.

In addition, we included simulated fail cases:

- Invalid grammar construct values.
- Missing required fields
- An empty request body
- Exceeding request limits
- Incorrect data types

Each failure case is logged to the console and marked, as run-time scripting cannot issue multiple internal requests.

Since the first version of the tests, the improvements have included error handling which helped identify and resolve issues with Content-Type headers. Possible future developments could include automating request bodies with dynamic input sets to support broader test coverage across different input sets.

While our extension focuses only on POST requests, the test collection contains fundamental cases to evaluate functioning structure and graceful failures.

5 Conclusion

This project presents an extension for Annotated Reading Enhancement System (ARES) aimed at supporting grammar development among EFL learners. Through the LLM-powered servlet we developed for grammar question generation, we promote grammatical development and learner autonomy through individualized practice.

While the ARES system has demonstrated its effectiveness in delivering immediate feedback in classroom settings, our extension specifically targets grammar reinforcement—an important component of language acquisition. Moreover, the user-friendly interface and feedback mechanisms are designed to support ease of use and continued learner engagement. Future usability studies can be helpful in refining the system's design to ensure it remains accessible and pedagogically sound (Baturay, 2010).

Despite limitations such as potential overreliance on automated suggestions and challenges in accommodating varying learner proficiency levels, our servlet demonstrates strong potential as an effective tool in grammar instruction. Looking ahead, we aim to expand the system to support additional question types such as multiple choice and matching, as well as integrate automated feedback and learner proficiency tracking for personalization.

As artificial intelligence continues to shape the future of education, this project underscores the role of AI-assisted systems in advancing language learning toward more customized, autonomous, and data-driven approaches that target learners' individual needs.

References

Luiz A. Amaral and Detmar Meurers. 2011. On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL*, pages 4–24.

Meltem Huri Baturay. 2010. Enhancement of usability and user friendliness of an online learning material through users' suggestions. *TECHNOLOGICAL APPLIED SCIENCES*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

R.M. DeKeyser. 2014. Skill acquisition theory. In *B. VanPatten J. Williams (Eds.), Theories in Second Language Acquisition: An Introduction*, pages 97–113. Routledge.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing

Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

Françoise Grellet. 1981. *Developing Reading Skills*. Cambridge University Press, New York.

Linzhao Jia, Han Sun, Yuang Wei, Changyong Qi, and Xiaozhe Yang. 2025. Epic: Error pattern informed correction for classroom asr with limited labeled data. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India. IEEE.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

J. Jung. 2009. Second language reading and the role of grammar. *Working Papers in TESOL & Applied Linguistics*, pages 29–48.

Hüseyin Küçükoglu. 2013. Improving reading skills through effective reading. *Procedia - Social and Behavioral Sciences*, pages 709–714.

Mihwa Lee, Björn Rudzewitz, and Xiaobin Chen. 2024. Developing a pedagogically oriented interactive reading tool with teachers in the loop. In *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2024)*, pages 115–125.

Ali Malmir and Nastaran Parhizkari. 2021. The effect of definition, fill-in-the-blank, and sentence writing exercises on the acquisition, retention, and production of lexical vs. grammatical collocations. *Teaching English as a Second Language Quarterly (Formerly Journal of Teaching Language Skills)*, 40(1):33–82.

J. Marjokorpi. 2024. Grammatical understanding predicts reading comprehension in secondary-level students: Insights from a finnish national survey. *Language and Education*.

Anne O'Keeffe and Geraldine Mark. 2017. The english grammar profile of learner competence. *International Journal of Corpus Linguistics*, 22(4):457–489.

Postman. Postman api platform. `https://www.postman.com/`. Accessed: 2025-04.

Postman Learning Center. Data-driven testing. `https://learning.postman.com/docs/testing/data-driven-tests/`. Accessed: 2025-04.

Kathrin Seßler, Arne Bewersdorff, Claudia Nerdel, and Enkelejda Kasneci. 2025. Towards adaptive feedback with ai: Comparing the feedback quality of llms and teachers on experimentation protocols. *arXiv*, pages 1–23.

Ludo Verhoeven, Jan van Leeuwe, and Anne Vermeer. 2011. Vocabulary growth and reading development across the elementary school years. *Scientific Studies of Reading*, pages 8–25.

Josip Vrdoljak, Zvonimir Boban, Marino Vilovic, Marko Kumric, and Joško Božic. 2025. A review of large language models in medical education, clinical decision support, and healthcare administration. *Healthcare*.

Shijie Wu, Benjamin Van Durme, and Mark Dredze. 2022. Zero-shot cross-lingual transfer is under-specified optimization. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 236–248, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

B Prompts

Generate {num_of_questions} grammar questions in a fill-in-the-blank format on the topic of {construct} at the CEFR level {level}. Create exactly {negative_sentences} negative sentences, {neutral_sentences} neutral sentences, and {interrogative_sentences} interrogative sentences. Your response should be in JSON format with the following structure:

```
{
 "level": "assigned_level",
 "topic": "assigned_grammatical_topic",
 "questions": [
    {
        "number_of_the_question": "question_number",
        "type_of_question": "negative or neutral
                             or interrogative",
        "question": "question_text",
        "answer": "answer_text"
    }
    ]
}
```