Paragraph-level Error Correction and Explanation Generation: Case Study for Estonian

Martin Vainikko[1], Taavi Kamarik[2], Karina Kert[2], Krista Liin[1], Silvia Maine[2], Kais Allkivi[2], Annekatrin Kaivapalu[3], Mark Fishel[1],

[1]Institute of Computer Science, University of Tartu;
[2]School of Digital Technologies, Tallinn University;
[3]Department of Finnish, Finno-Ugrian and Scandinavian Studies, University of Helsinki

Correspondence: martin.vainikko@ut.ee, taavi.kamarik@tlu.ee, karina.kert@tlu.ee, krista.liin@ut.ee, silvia.maine@tlu.ee, kais.allkivi@tlu.ee, annekatrin.kaivapalu@helsinki.fi, mark.fisel@ut.ee

Abstract

We present a case study on building task-specific models for grammatical error correction and explanation generation tailored to learners of Estonian. Our approach handles whole paragraphs instead of sentences and leverages prompting proprietary large language models for generating synthetic training data, addressing the limited availability of error correction data and the complete absence of correction justification/explanation data in Estonian. We describe the chosen approach and pipeline and provide technical details for the experimental part. The final outcome is a set of open-weight models, which are released with a permissive license along with the generated synthetic error correction and explanation data.

1 Introduction

Language models with emergent abilities are increasingly showing capacity for performing natural language processing tasks via prompting (OpenAI et al., 2024; Grattafiori et al., 2024; DeepSeek-AI et al., 2025, etc.). However, it has been shown that targeted effort can result in surpassing the most advanced proprietary models with more task-oriented models, e.g., for grammatical error correction (Luhtaru et al., 2024a). Furthermore, hybrid combinations of large language model (LLM) prompting and tuning for synthetic data generation, as well as tuning for the final task, show even more promise (Luhtaru et al., 2024b).

Here we present a case study on the development of grammatical error correction (GEC) and grammatical error explanation (GEE) generation for learners of Estonian. The overall goal is to create task-specific models reliable enough to correct learners' grammar and justify the corrections. Most importantly, while we use proprietary LLMs in this work for data generation, the final result consists of independent open-weight models that can be used for both research and commercial purposes.

The central theme of all the presented work is dealing with data scarcity. The amount of training data for GEC has recently improved but still shows imbalance between English and other languages (Masciolini et al., 2025b) and Estonian is no exception. More specifically, there is a modest amount of Estonian GEC data but no data for GEE. We address both data deficiencies by utilising synthetic data, obtained by prompting OpenAI LLMs (detailed later in the paper) to either introduce grammatical errors into correct texts, in a manner characteristic for language learners (for GEC), or by generating and filtering explanations of gold-standard corrections (for GEE).

Below we describe the developed pipeline and details of generating the data and training the final models in Section 3. Then we present a comprehensive qualitative and quantitative evaluation of the results in Section 4. Finally, Section 5 describes the user feedback, collected from two groups of users: teachers and learners of Estonian as a second language (L2). Since the presented project is an ongoing effort, we finish with a brief description of lessons learned and future work in Conclusion 6.

2 Related Work

2.1 Grammatical Error Correction

The task of grammatical error correction (GEC) is to automatically detect and correct erroneous text. Bryant et al. (2023) argue that although the denomination of the task refers to grammatical errors, the scope of the task is not strictly limited to grammatical errors but other types of errors as well, such as spelling and fluency errors.

Recent approaches have moved from neural MT (Yuan and Briscoe, 2016) to LLM-based (Masciolini et al., 2025a). Even without downstream fine-tuning, LLMs have shown to generate grammatically correct text as an emergent ability (Cao et al., 2023; Coyne et al., 2023), but the edits tend to be

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 953–967

fluency edits as opposed to minimal edits (Fang et al., 2023; Davis et al., 2024).

Automatic error generation (AEG) is a widely applied approach in GEC, consisting of injecting automatically generated errors into correct sentences in order to generate synthetic GEC data. Approaches to AEG include rule-based (Sidorov et al., 2013; Ma et al., 2022), statistical methods (Felice and Yuan, 2014; Kasewa et al., 2018), and neural networks-based work (Grundkiewicz et al., 2019; Bout et al., 2023).

Korotkova et al. (2019) used neural MT for GEC for Estonian. Luhtaru et al. (2024b) used fine-tuned LLMs for both artificial error generation and correction. They used L2 essays for generating errors and evaluated the results on the Estonian learner language (EstGEC-L2) corpus[1]. They concluded that using Llama-2-based fine-tuned models gave the most human-like distribution of generated errors. Another dataset, the EKI error-annotated L2 (EKI-L2) corpus[2], was released in 2024. The two corpora are included in the MultiGEC-2025 shared task (Masciolini et al., 2025b). The best results were achieved by the multilingual LLM based model of Staruch (2025), providing the whole essay at once for correction. Most GEC approaches and evaluation methods are sentence-based (Bryant et al., 2023), including previous work in Estonian GEC, which limits the system's access to the broader context necessary for correctly detecting and correcting paragraph- or document-level errors.

2.2 Grammatical Error Explanation

Alongside GEC, the task of grammatical error explanation (GEE) has received increasing attention. Providing a reason for each correction helps language learners and other users to understand and learn from their errors.

Chen et al. (2017) extracted grammar patterns from a reference corpus to assist L2 learners of English in academic writing. E.g., the correction *chance for giving* → *chance to give* would be explained by the edit pattern *chance: N for -ing* → *N to v*. Lai and Chang (2019) also detected problem words co-occurring with grammar edits. They formulated feedback templates depending on the error type, classified by ERRANT (Bryant et al., 2017).

Another enhanced GEC system by Kaneko et al. (2022) presents related language examples based on k-nearest-neighbour machine translation trained with incorrect-correct sentence pairs from English learner corpora. However, GEE-specific datasets allow to train models that give more detailed responses. Hanawa et al. (2021) experimented with neural retrieval and generation methods using L2 English essays manually annotated with feedback comments. Fei et al. (2023) introduced a dataset with error type and problem word annotations, using it for BERT-based token classification and error class prediction.

While it is costly to produce human-annotated or carefully engineered corpus-induced training data, the prompting of LLMs can offer a more accessible solution for GEE. Maity et al. (2024) prompted various LLMs in one-shot mode to correct erroneous Bengali sentences and obtain a brief explanation of each error. Song et al. (2024) achieved a better GEE performance with a two-step pipeline for explaining German and Chinese error corrections. First, they prompted and fine-tuned LLMs to extract atomic edits (insert, delete, replace, relocate). Then, explanations were generated by few-shot prompting GPT-4. This significantly improved the results compared to using only sentence pairs as input. Kaneko and Okazaki (2024) and Ye et al. (2025) similarly leveraged the in-context learning capabilities of the GPT models to synthesize English and Chinese error explanation data, respectively. Ye et al. (2025) used their dataset to fine-tune open-source LLMs both in a pipeline and multi-task setting, integrating GEC and GEE.

We adopt the LLM-based pipeline approach and include error types in addition to atomic edits. As a novel contribution, we provide each edit with two explanations of different detail levels: 1) a brief overview of the error cause and 2) a more comprehensive reasoning mainly aimed at advanced learners and teachers. We create synthetic data with both types of explanations by few-shot prompting GPT-4o and fine-tune a Llama-2-based LLM adapted for Estonian.

3 System Development

The system development consisted of data generation and fine-tuning for GEC and GEE. The resulting system pipeline consists of three steps: 1) grammatical error correction, 2) error tagging and 3) error explanation. The models, alongside the generated synthetic training datasets, are public and have a permissive license.

[1] https://github.com/tlu-dt-nlp/EstGEC-L2-Corpus/
[2] https://doi.org/10.15155/27bh-ny83

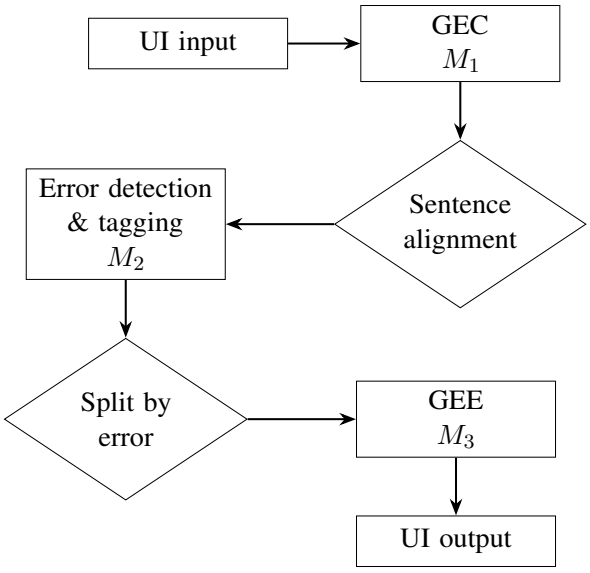

Figure 1: A high-level overview of the system. Each M denotes a model fine-tuned for the given task.

Figure 1 gives a high-level overview of the system (see a detailed example in Appedix A). The user's input text, i.e., a paragraph, is passed to the first model M_1 as a whole, which then outputs the corrected text.[3] The input and corrected text are split into sentences and aligned with the sentence aligner, resulting in input-output sentence pairs. If the input sentence is not equal to the output sentence, thus an error was corrected, the pair is passed to the second model M_2, which outputs a list of tagged error corrections for each sentence pair.[4] If input and output are equal, the following models are skipped. Otherwise, the sentence pair and the list of tagged errors are passed to the third model M_3 error by error, which explains the error correction.[5]

Next, we discuss these steps in detail. Fine-tuning is elaborated in Subsection 3.1. Subsections 3.2 and 3.3 delve into GEC and GEE model development, which involves experiments for data generation and finally generating the final datasets and fine-tuning the models. Although the GEC model works on paragraph level, GEE was performed on the sentence level due to context window limitations; to that extent, we performed sentence alignment, details of which are given in Subsection 3.4.

[3]https://huggingface.co/tartuNLP/Llammas-base-p1-GPT-4o-human-error-mix-paragraph-GEC

[4]https://huggingface.co/tartuNLP/Llammas-base-p1-GPT-4o-human-error-pseudo-m2

[5]https://huggingface.co/tartuNLP/Llammas-base-p1-GPT-4o-human-error-explain-from-pseudo-m2

3.1 Fine-tuning details

Base model For the base model we chose Llammas-base, which is a Llama 2 7B model that has been trained on Estonian texts in continued pre-training setting (Kuulmets et al., 2024) and has shown SOTA results by fine-tuning for Estonian sentence-level GEC (Luhtaru et al., 2024b).

Training parameters Apart from the maximum sequence length, which was 4096, 2048 and 4096 for M_1, M_2 and M_3, we used the same parameters as Luhtaru et al. (2024b) for all three models. For prompts, see Appendix B.

Hardware The models were trained on 2 AMD MI250X GPUs in the LUMI supercomputer,[6] which totals 4 GPUS because AMD MI250X GPU is considered as two GPUs from both hardware and software perspectives in LUMI, each having access to 64 GB of memory.

3.2 Error correction and error type detection

For human-annotated training data, we used the EKI-L2 GEC corpus, which is part of MultiGEC-2025 (Masciolini et al., 2025b). It consists of 1,503 learner essays and 17,361 sentences, nearly 3/4 of which include at least one grammatical error. The dataset includes both minimal and fluency edits, we used the part with minimal corrections. Similarly to EstGEC-L2, its annotation follows the M^2 format and ERRANT error classification (Bryant et al., 2017) adapted for Estonian.

To increase the amount of pre-training data, we also generated synthetic data for Estonian. Moreover, as Luhtaru et al. (2024b) show that inserting errors into out-of-domain texts can actually hurt performance, we compare synthetic error addition to the original GEC data (corrected essays in EKI-L2), similar-domain human texts and synthetically generated texts. Human data from the similar domain consists of similar-sized excerpts of fiction, extracted from the Estonian National Corpus[7]).

We used OpenAI's GPT-4o[8] to generate essays based on EKI-L2. For each original corrected essay, we gave it as a 1-shot sample and prompted the model to generate a similar, correctly written essay given the original proficiency level. With temperature 1 the generated essays followed the argumentative structure of the sample with a new

[6]https://www.lumi-supercomputer.eu/

[7]https://doi.org/10.15155/3-00-0000-0000-0000-08D17L

[8]https://platform.openai.com/docs/models/gpt-4o

topic, at higher temperature levels the amount of noise rose sharply, so the essays were generated at 1.1. While we filtered out most of the noise, a few generated essays did include some grammatical errors.

GPT-4o few-shot prompting (at temperature 1.0) was then used to generate errors in these texts, sentence-by-sentence, given 5 randomly picked corrected-mistaken sentence pair samples from the original EKI-L2 corpus. Each of the four datasets was used as error correction training data to fine-tune a Llammas-base (Kuulmets et al., 2024) model for 3 epochs, which was then tested on 141 essays of the development set of EstGEC-L2 corpus (levels A2–C1).[9] For training details, see Subsection 3.1. As some grammatical errors can only be detected when considering the context, the error correction models were given the whole essay as input, splitting only if the essay was too long to fit into the context window. The results can be seen in Table 1.

	P	R	$F_{0.5}$
Human errors			
EKI-L2	**76.64**	40.35	**64.95**
Synthetic errors			
EKI-L2	71.40	41.45	62.39
Fiction excerpts	70.55	46.55	63.96
Generated essays	69.70	**49.19**	64.33

Table 1: Error correction precision (P), recall (R) and F-score ($F_{0.5}$) after 3 epochs of training on different genres. Scores of the model trained on human errors versus models trained on synthetic errors generated into the listed datasets. EKI-L2 synthetic errors were generated into the target sentences, leaving with synthetic source sentences.

As automatically generated essays proved to yield good results compared to other training corpora, we 1) generated a 10 times larger set of essays, 2) introduced artificial errors to the generated essays and 3) employed a two-stage fine-tuning procedure for GEC. We first fine-tuned Llammas-base on a randomly shuffled 10:1 mixture of synthetic-human data. We then fine-tuned the best-performing checkpoint from the first stage on EKI-L2 human dataset. The checkpoint with the highest $F_{0.5}$ score on the development set served as the base model for the second stage of the fine-tuning, which was fine-tuning the model again on the human dataset. The optimizer state was reinitialized and the hyperparameters remained the same as in the first stage of fine-tuning. The third checkpoint of the final model served as the GEC model M_1 in the workflow.

For error detection and classification, we transformed EKI-L2 M^2 edits into simplified atomic edits with error type information, to be given as input for GEE. We fine-tuned a Llammas-base on the EKI-L2 set with atomic edits, resulting in the error tagging model M_2 in the workflow.

3.3 Error explanation

To generate training examples for GEE, we evaluated three approaches using OpenAI's GPT models: 1) single-prompt parallel input, where the model was given original and corrected sentence pairs; 2) single-prompt error-tagged input, which provided correction edits and error-type information; and 3) prompt chaining with parallel input, which identified and explained corrections through separate prompts. These approaches were assessed using zero-shot and few-shot prompting.

The evaluation was based on 40 random sentence pairs from the EstGEC-L2 development set, including 10 pairs per proficiency level (A2–C1). In case of multiple error annotations, the first version was chosen. For each error, we requested either a single explanation or paired explanations: one brief and one more comprehensive. We rated their quality using colour codes based on traffic lights, so that green indicates good, yellow fair and red poor explanations. More precisely, green represents clear and sufficient information. Yellow denotes partial or nonfluent information that may still be helpful and does not mislead the user. Red explanations contain incorrect statements and terms, or simply describe the correction, but do not offer a justification. The explanation accuracy was defined as the percentage of good and fair explanations.

Annotators were three research group members with a linguistic background and previous experience in L2 error annotation. There was one annotator per each explanation. The annotation was reviewed by an L2 teaching expert participating in our project. The expert-guided evaluation principles were jointly discussed and specified throughout the evaluation process.

Initial experiments used Estonian and English zero-shot prompts and compared the performance of GPT-4o, GPT-4, and GPT-3.5 Turbo with Microsoft Azure's default settings (temperature 0.7, top p 0.95) and reduced variability (lowering ei-

[9] Originally 102 essays; longer essays were split.

ther of the parameters). GPT-4o with default settings outperformed other models, producing fewer factual or logical errors, particularly in detecting Estonian case forms and sentence interpretation. Notably, Estonian prompts yielded more precise and fluent explanations. Requesting paired explanations provided higher-quality responses. In particular, comprehensive explanations could be considered more accurate and informative compared to single ones. Brief explanations were more problematic, often describing corrections (e.g., word *x* should be *y*) without any additional context.

The results generally improved by first generating the longer explanation instead of the shorter one. This way, the accuracy of long explanations increased from 29% to 64% with single-prompt parallel input and from 62.5% to 83% with error-tagged input. Brief explanation accuracy went from 0% to 18% with single-prompt parallel input and dropped from 54% to 48% with error-tagged input.

Adopting the paired explanation approach, we refined the best Estonian prompts to avoid redundant or insufficient information. For few-shot prompting, we constructed examples based on eight Estonian learner sentences, representing the 12 main error types and some combined errors (see (1) for an example of explanation input and output, translated into English). The single-prompt approach proved more effective with the few-shot method, whereas the prompt chaining did not yield better results. Its long explanation accuracy decreased from 58% to 36% and brief explanation accuracy from 56% to 44% compared to the zero-shot method. The main limitation was detecting atomic edits despite the few-shot examples. In a test where GPT-4o had to identify GEC edits three times per sentence pair and select the correct answer, it chose the right output for 28 out of 40 sentence pairs.

(1) Source sentence: Head aega.
Target sentence: Head aega! (‘Goodbye!’)
Correction(s):
1. incorrect punctuation: . -> !

Explanation 1: . -> !
Long: In Estonian, a greeting sentence ends with an exclamation mark, e.g., “Tere hommikust!” (‘Good morning!’), “Head uut aastat!” (‘Happy new year!’).
Brief: An exclamation mark is used after a greeting or wish.
Error type: incorrect punctuation

Since the error-tagged input provided full alignment with actual edits and error types, we decided to use it for training data generation. In addition, this approach lead to significantly higher accuracy in longer explanations (91% compared to 65% with parallel input). The accuracy of brief explanations was lower, equally 52%, mostly due to merely descriptive explanations. Therefore, we further improved the prompt to provide more meaningful clarifications. We synthesized error explanations based on the EKI L2 corpus from Subsection 3.2, using the 12,580 sentences that contain the atomic edits. We fine-tuned a Llammas-base model to generate explanations error-by-error on the synthesized dataset, resulting in model M_3 in the workflow.

3.4 Sentence alignment

As the error explanation model required input on sentence level, the essay from model M_1 error correction output had to be aligned with its input on sentence level. The same need came up when evaluating M_1 output. Complications rose when a sentence was split into several or several sentences joined as part of the correction, also when the M_1 model hallucinated new sentences, such as a greeting to start a letter. Sometimes a mismatch was caused by the sentence tokenizer mistaking the sentence boundaries in uncorrected text.

To solve this problem, we developed a simple many-to-many sentence aligner based on the Levenshtein distance. When aligning the gold standard and output essays during evaluation we considered the distance between corrected sentences of both essays, merging the gold M^2 representations as necessary. Testing on 400 essays of the training corpus, this rule-based aligner found correct alignments for 98% of the original sentences.

4 System Evaluation

4.1 Error correction performance

Error correction scores were automatically evaluated on the EstGEC-L2 development set using a modified version[10] of the M^2 scorer (Dahlmeier and Ng, 2012). This yields error-level F-score comparing the output sentence with all given gold corrections, as well as a broad statistics of recall by error type. The modified version also takes into account that the word order error type (*R:WO*) used in train and test corpora can encompass other errors, as word order in Estonian tends to be rather

[10] https://github.com/TartuNLP/estgec/

free and the scope of that error type may include a large part of the sentence. The results on the development set of EstGEC-L2 corpus can be seen in Table 2, comparing the models trained on smaller or larger datasets of synthetic essays, and the latter model post-fine-tuned on human errors. While using a large number of generated essays did significantly raise recall, it is surprising that additional fine-tuning on human errors brought it down without much increase in precision. This may be partly due to the larger training set containing human errors already contributing to higher precision.

	P	R	$F_{0.5}$
GPT-4o	69.56	**54.13**	65.81
Synth_S	69.70	49.19	64.33
Synth_L-EKI-L2 Mix	75.61	45.68	**66.85**
+ EKI-L2 FT	**76.45**	42.45	65.90

Table 2: Scores of models trained on datasets with synthetic errors, based on the best $F_{0.5}$ score across 3 epochs, compared to prompting GPT-4o at temperature 1 for GEC in a 1-shot setting (see Appendix B for details). Synth_S was trained on 1,503 generated essays with synthetic errors, while Synth_L-EKI-L2 Mix was trained on a dataset 10× larger, mixing synthetic and EKI-L2 errors. Synth_L-EKI-L2 Mix was also post-fine-tuned on EKI-L2.

While our $F_{0.5}$ score is notably higher than the 49.44 reported by Staruch (2025), theirs was a multilingual system tested on a smaller subset of the EstGEC-L2 corpus. GPT-4o achieves a higher recall but a lower precision, resulting in a lower $F_{0.5}$ score. We incorporated the EKI-L2 fine-tuned Synth_L-EKI-L2 Mix model trained for 3 epochs in our workflow as the final M_1 model, although its $F_{0.5}$ score was better after epoch 1, recall was highest after 3 epochs.

The modified M^2 scorer shows recall by error type even if M_1 does not assign types. Considering the results (see Figure 2), the most difficult type by far is word order, mostly because the correction is considered accurate only if all possible encompassing errors are corrected as well. E.g., the phrase '*raamat loen ma*' (*'book-nom read I'*) should be corrected not only as '*ma loen raamat*' (*'I read book-nom'*), but also with the correct case '*ma loen raamatut*' (*'I read book-part'*). Note that the error correction model does not assign an error type, so even if it detects an error in the same scope as a nominal form error, it might try to replace or erase the whole word.

Leaving word order aside, the more difficult types to correct are word and punctuation choice (*R:LEX*, *R:PUNCT*), although missing punctuation marks (*M:PUNCT*) tend to be rather easy. This is somewhat expected as the choice of words for replacing an unsuitable one is rather large and not all suitable words are listed in human corrections. Inserting missing punctuation marks, correcting the capitalization (*R:CASE*) or whitespace, i.e., compounding errors (*R:WS*) as well as picking the right nominal or verb form (*R:NOM:FORM*, *R:VERB:FORM*) are all handled with slightly higher recall, as could be expected from a strong language model. As was seen from the evaluation scores, adding synthetic data helped raise recall. This seems to be mostly due to better detection of word order and compounding errors. The final model also detects capitalization errors noticeably better than the one trained on human errors, but if we consider corrections, then they are around the same level, as the model has trouble providing correct replacements. If we consider what may have contributed to the drop of precision in the final model, then most noticeable bottlenecks are detecting unnecessary words (*U:LEX*) and correcting complex errors where there are several mistakes in one word (e.g., wrong verb form with a spelling mistake – *R:VERB:FORM:SPELL*).

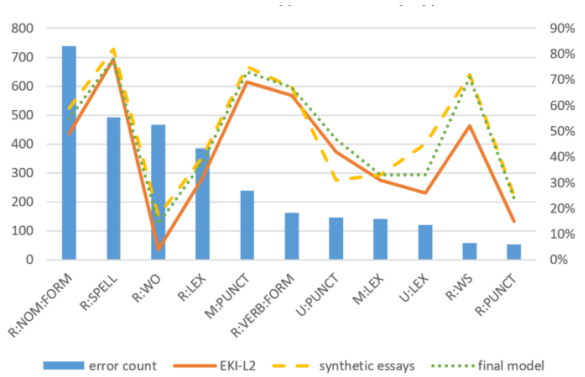

Figure 2: Recall by type for the 11 most frequent error types in the test corpus. Number of errors of each type present in test corpus is shown in columns for reference.

The test dataset has essays from proficiency levels A2–C1, whereas C1 was not present in the training corpus that contains K-12 student essays. When comparing error correction performance of the fine-tuned model across language proficiency levels (see Table 3), the results are quite uniform. The scores are a little better for A2, possibly because the sentences used are still rather simple. Recall is no-

	P	R	$F_{0.5}$
A2	74.12	48.99	67.22
B1	71.48	48.13	65.16
B2	74.00	43.32	64.82
C1	71.43	47.23	64.79
All	72.91	46.58	65.51

Table 3: GEC scores by language proficiency level.

ticeably lower for B2. The drop is most clear for two error types: word order and wrong punctuation mark. Out of 26 cases of wrongly chosen punctuation marks in B2 texts, none received the correct replacement, although a mistake was detected in more than third of these. B2 texts had more word order mistakes (196, compared to 127 in B1 texts and less than 100 on other levels), but even the recall of partial scope overlap was lower than for other levels: 32% compared to 47% or more. The distribution of more common error types and their detection rate can be seen in Figure 3. As for other proficiency levels, the model has relatively more difficulties detecting missing words in A2 texts, missing punctuation in B1 texts, and wrong capitalisation in C1 texts, although in the last case there were only 4 such mistakes present, which may be too few to draw conclusions.

4.2 Qualitative analysis of system output

For qualitative assessment of the three system components — corrector, error detector/classifier, and explainer — we randomly selected 40 sentences from the EstGEC-L2 test corpus, balanced for proficiency level (A2–C1). We compared two settings: a uniform 0.7 temperature and a varied higher temperature (1.0 for M_1, 0.8 for M_2, 0.9 for M_3) to encourage creativity.

In comparison with golden edits, we distinguished four correction types: necessary and suitable, necessary but incorrect, unnecessary but suitable, and unnecessary and incorrect. We calculated precision based on both types of suitable edits. The macro-averaged precision of error classification was assessed according to proposed changes, even if incorrect. Explanations were graded as good, fair, or poor, as described in section 3.2 (see translated examples in Appendix D). We separately evaluated explanations for necessary suitable corrections, since it is challenging or even futile to explain unnecessary or incorrect edits.

Lower temperature entailed higher correction precision (89% *vs.* 76%) and fewer edits (63 *vs.* 71), while the number of suitable corrections was similar (56 *vs.* 54). The 0.7 setting also resulted in a greater overlap with reference edits (60% *vs.* 46%). However, the correction model then failed to detect word order errors. We suggest using an intermediate temperature for GEC. The average precision of error classification was comparable in the two conditions: 84% with higher and 87% with lower temperature.

The quality of explanations was generally better at the 0.7 temperature (see Table 4). Long explanations were more likely to be rated good or fair compared to the 0.9 temperature both in case of necessary corrections and all corrections, including optional and unjustified edits. Necessary brief explanations followed a different trend, being more accurate at the higher temperature. Nonetheless, when considering all system corrections, the proportion of good and fair explanations remained similar in the 0.7 setting, whereas radically dropping in the 0.9 setting. This refers to a better capability to justify optional edits at the lower temperature.

	Temp 0.7	Temp 0.9
Long explanations		
Necessary: good	**51%**	37%
Necessary: good/fair	**66%**	48%
All: good	**51%**	27%
All: good/fair	**63.5%**	37%
Brief explanations		
Necessary: good	45%	**50%**
Necessary: good/fair	70%	**76%**
All: good	**46%**	34%
All: good/fair	**67%**	51%

Table 4: GEE quality with two temperature settings.

In terms of GEE, our results can be compared with Maity et al. (2024) and Ye et al. (2025), who reported accuracy over 60%, and outperform Hanawa et al. (2021), who reached 40%–50% accuracy in explaining preposition errors and below 40% with various error types. One shortcoming was the inaccuracy of linguistic terms, such as using an existing term in a wrong context (e.g., false association of nominal case and word form) or forming a nonexistent term. This concerned 24% of long and 3% of brief explanations in the lower temperature setting. Furthermore, the prompt could be improved to explain context-dependent errors like grammatical form or word choice errors.

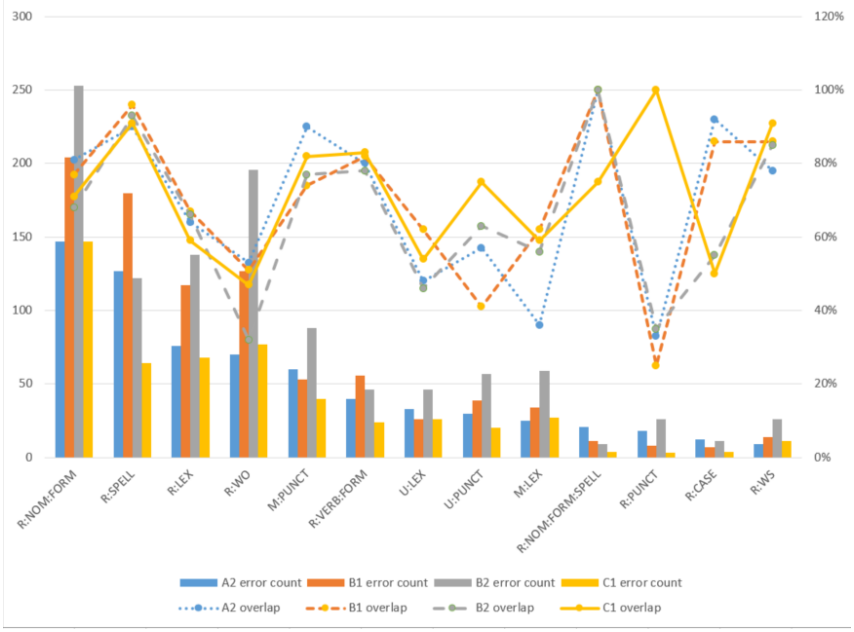

Figure 3: Error detection recall by language proficiency level for more common error types, considering at least partial overlap in scope.

5 User Feedback

5.1 Participants and questionnaire

The user test involved six learners and five teachers and/or testing experts of Estonian as L2. Volunteers were recruited using snowball sampling, considering their experience of learning and teaching at different proficiency levels. The teachers' expertise covered the whole range from pre-A1 to C1 level. Three teachers had taught adults and students from pre-A1- or A1- to C1-level. One had focused on adult learners with diverse backgrounds at levels A1 and A2. One had assessed exams at all tested levels but primarily at B2 and C1, prepared exam and screening test tasks, and briefly taught language courses. Three of the teachers were native Estonian speakers, one an Estonian-Russian bilingual and one a native Russian speaker who uses Estonian at home, at work and in daily life. The learners were Russian- and Ukrainian-speaking and bilingual (Ukrainian-Russian or Russian-Estonian). Three of them had lived in Estonia for many years or most of their lives and rated their language level as B2 and C1. The others reported their level to be A2 or B1, having spent 2.5–5.5 years in Estonia. Their exposure to Estonian ranged from rare use in lessons or grocery store to everyday use at work.

We used Google Forms to gather feedback through a semi-structured questionnaire. The respondents were given the option to answer the survey in English. We developed a demo application for testing (see Appendix C). First, we asked the users to assess the output for a sample B1-level text fragment. Repeated analysis of the same text may give varying results, so we presented a pre-given version of five corrections and explanations as screenshots to ensure response comparability. Subsequently, users interacted with the tool directly, correcting their own text or a student's writing, commenting on each correction and explanation and their general experience.

5.2 Results

Both teachers and learners found that the system makes most of the needed corrections and the majority of explanations could be useful in existing form or with some changes. 10 out of 11 test users considered the corrections somewhat useful or useful (corresponding to 4–5 on a 5-point Likert scale). Explanations were rated similarly by nine respondents. Three teachers and learners noted their plan to use the application in the future, one teacher and two learners would probably use it and one respondent from each group was not sure about it.

All corrections in the provided sample were considered appropriate, although two teachers noted that a lexical choice correction was not strictly necessary. Each explanation was rated on a three-point scale: useful – somewhat useful – not useful. Depending on the correction, long explanations

were found useful by 5–9 and brief explanations by 5–10 test users, averaging to 2/3 of the respondents. About 1/3 and 1/4 of the users, respectively, considered long and brief explanations somewhat useful. On average, one user did not think the shorter explanation was useful. The respondents agreed to defined error types, except for the case where word order and nominal form error occurred together but only the former was detected.

As expected, analysing user texts revealed more issues because these texts were generally longer than the sample and the system made more corrections in them. Teachers found an average of 82% and the learners 87% of corrections relevant. Both groups considered about 70% of long explanations at least somewhat useful, whereas brief explanations seemed useful to more than half of the learners and 3/4 of the teachers. Fixing and explaining structural errors in long complex sentences that contained numerous errors turned out to be challenging. The tool also had some trouble identifying and explaining combined errors. While our aim was to classify and explain all co-occurring error types, only one type may be detected and covered in the explanations (e.g., a spelling error is ignored alongside the choice of correct word form).

The explanations were said to give a comprehensive overview of the errors and help language learners notice errors they might be making systematically. Long explanations were rated higher in terms of content and wording, although there were also instances of complex language use or no added value compared to the shorter version. Users claimed that long explanations should complement short ones, while short explanations should still be informative. In some cases, two explanation layers may not be needed. Both teachers and learners recommended to put more emphasis on simple language comprehensible for A2- and B1-level learners, especially in brief explanations. Some suggestions were made to generalise or specify error classification and improve the user interface, however, the overall assessment was positive in both respects.

6 Conclusion

We trained a workflow of three fine-tuned models for GEC and GEE. Using synthesized L2 texts with introduced errors seems promising, but a larger training set might be necessary. Our first model corrects errors at the paragraph level and performs well on L2 texts with a proficiency level not present in training data. The model yields higher precision with similar recall at lower temperatures but then struggles with word order errors, so we suggest using medium temperature.

We achieved better quality explanations in GEE by incorporating error types in addition to atomic edits in input and requesting two explanations (longer and shorter) for each error. This could be further improved as both LLM prompting and our fine-tuned model have low recall on detecting error types. It is also necessary to filter out low-quality explanations, such as including nonexistent nominal cases, by possibly using LLMs to evaluate the quality of a given explanation.

While our GEC model was paragraph-based, we used a sentence-based approach due to model limitations. In future work, we will apply a new methodology to preserve context and fit within hardware limits with the context window size. For each sentence in the essay, we will split essays into tuples of N consecutive sentences up to the given sentence. The new methodology could allow us to combine GEC and GEE into one model. In future work, we will also explore reversing the pipeline for fine-tuning for AEG.

Acknowledgments

The project "Autocorrect for Estonian as a 2nd language for learners and teachers" has been co-funded by Estonian Ministry of Education and Research and the European Union. We acknowledge University of Tartu, Estonia for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through University of Tartu, Estonia.

Limitations

Although our GEC system is based on paragraphs, the GEE pipeline, including both error detection and explanation, is based on sentence level due to context window limitations, limiting the model's capability to explain errors at the document level. Future work needs to classify and explain errors with context, as well as combine related errors for explanation.

The GEE pipeline relies on atomic edits with error-type information, which we found necessary for reasonable explanations. However, atomic edits are based on M^2, thus making it costly to obtain

new data. Future work should explore the automatic generation of atomic edits.

Additionally, GEC scores rely highly on the sentence alignment method since the M^2 scorer works on the sentence level. Poor sentence alignment affects the scores negatively.

References

Kais Allkivi, Pille Eslon, Taavi Kamarik, Karina Kert, Jaagup Kippar, Harli Kodasma, Silvia Maine, and Kaisa Norak. 2024. ELLE – estonian language learning and analysis environment. *Baltic Journal of Modern Computing*, 12(4):560–569.

Andrey Bout, Alexander Podolskiy, Sergey Nikolenko, and Irina Piontkovskaya. 2023. Efficient grammatical error correction via multi-task training and optimized training schedule. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5800–5816, Singapore. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pages 643–701.

Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Unnatural error correction: GPT-4 can almost perfectly handle unnatural scrambled text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8898–8913, Singapore. Association for Computational Linguistics.

Jhih-Jie Chen, Jim Chang, Ching-Yu Yang, Mei-Hua Chen, and Jason S. Chang. 2017. Extracting formulaic expressions and grammar and edit patterns to assist academic writing. In *EUROPHRAS 2017 - Computational and Corpus-based Phraseology: Recent Advances and Interdisciplinary Approaches*, pages 95–103.

Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction. *Preprint*, arXiv:2303.14342.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 568–572.

Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. Prompting open-source and commercial language models for grammatical error correction of English learner text. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.

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, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *Preprint*, arXiv:2304.01746.

Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden. Association for Computational Linguistics.

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, and 542 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Masahiro Kaneko and Naoaki Okazaki. 2024. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3955–3961, Torino, Italia. ELRA and ICCL.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium. Association for Computational Linguistics.

Elizaveta Korotkova, Agnes Luhtaru, Maksym Del, Krista Liin, Daiga Deksne, and Mark Fishel. 2019. Grammatical error correction and style transfer via zero-shot monolingual translation. *arXiv preprint arXiv:1903.11283*.

Hele-Andra Kuulmets, Taido Purason, Agnes Luhtaru, and Mark Fishel. 2024. Teaching llama a new language through cross-lingual knowledge transfer. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3309–3325, Mexico City, Mexico. Association for Computational Linguistics.

Yi-Huei Lai and Jason Chang. 2019. TellMeWhy: Learning to explain corrective feedback for second language learners. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 235–240, Hong Kong, China. Association for Computational Linguistics.

Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024a. No error left behind: Multilingual grammatical error correction with pre-trained translation models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1222, St. Julian's, Malta. Association for Computational Linguistics.

Agnes Luhtaru, Taido Purason, Martin Vainikko, Maksym Del, and Mark Fishel. 2024b. To err is human, but llamas can learn it too. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12466–12481, Miami, Florida, USA. Association for Computational Linguistics.

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, Haitao Zheng, and Ying Shen. 2022. Linguistic rules-based corpus generation for native Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Subhankar Maity, Aniket Deroy, and Sudeshna Sarkar. 2024. How ready are generative pre-trained large language models for explaining bengali grammatical errors? *Preprint*, arXiv:2406.00039.

Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, and Robert Östling. 2025a. The MultiGEC-2025 shared task on multilingual grammatical error correction at NLP4CALL. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 1–33, Tallinn, Estonia. University of Tartu Library.

Arianna Masciolini, Andrew Caines, Orphée De Clercq, Joni Kruijsbergen, Murathan Kurfalı, Ricardo Muñoz Sánchez, Elena Volodina, Robert Östling, Kais Allkivi, Špela Arhar Holdt, Ilze Auzina, Roberts Darģis, Elena Drakonaki, Jennifer-Carmen Frey, Isidora Glišić, Pinelopi Kikilintza, Lionel Nicolas, Mariana Romanyshyn, Alexandr Rosen, and 11 others. 2025b. Towards better language representation in natural language processing. *International Journal of Learner Corpus Research*, 11(2):309–335.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena, and Sandrine Fuentes. 2013. Rule-based system for automatic grammar correction using syntactic n-grams for English language learning (L2). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 96–101, Sofia, Bulgaria. Association for Computational Linguistics.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2024. GEE! grammar error explanation with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.

Ryszard Staruch. 2025. UAM-CSI at MultiGEC-2025: Parameter-efficient LLM fine-tuning for multilingual grammatical error correction. In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 42–49, Tallinn, Estonia. University of Tartu Library.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, and 1 others. 2025. Excgec: A benchmark for edit-wise explainable chinese grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25678–25686.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 conference of the north American Chapter of the Association for computational linguistics: Human language technologies*, pages 380–386.

A System Overview

Figure 6 gives a detailed overview of the system with an example. The input paragraph translates to "I very much like classical music. We earlier bought the tickets. Sometimes already in January." The explanations translate to "The correct spelling is "väga" (very)", "Instead of the word "klassikat" ("classic" in the wrong case form), the word "klassikaline" should be used to show that the music is of classical type", "In Estonian, the verb is generally before the adverb (adverb of time), for example, "ostsime piletid" (we bought the tickets) and "varem" (earlier). The original word order "varem ostsime piletid" is wrong."

B Prompts

Tables 5, 6 and 7 display the prompts used for fine-tuning M_1, M_2 and M_3. The prompt format is from Luhtaru et al. (2024b), which in turn is loosely based on Alpaca (Taori et al., 2023) format. Table 8 shows the prompt used for GPT-4o for GEC.

C Demo application

For user testing, we developed a demo application[11] based on the Writing Evaluator proofreading tool of an Estonian language learning and analysis environment (Allkivi et al., 2024). The demo reuses existing interface components, such as approving or rejecting corrections and grouping errors by type.

After the user inserts their text, the back-end returns the corrected sentences along with error annotations, including error type and two accompanying explanations. Users can interact with corrections in two ways: **inline view** — moving the cursor over a highlighted segment in the text triggers a popup displaying a short explanation and options to accept or reject the correction (See Figure 4); **sidebar view** — on the right side of the interface, corrections are grouped by type (see Figure 5). Clicking on a category reveals a list of related corrections with longer explanations.

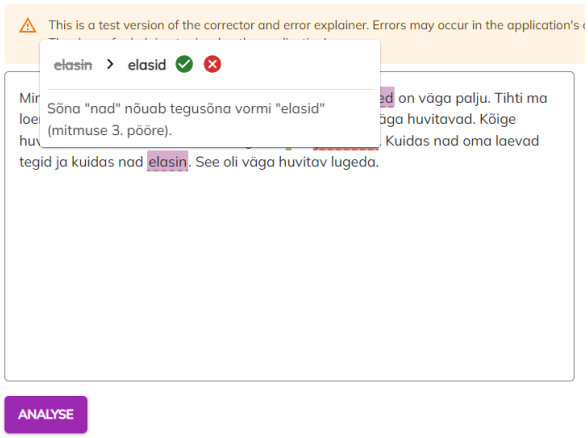

Figure 4: Popup view with a short explanation when hovering over a highlighted correction.

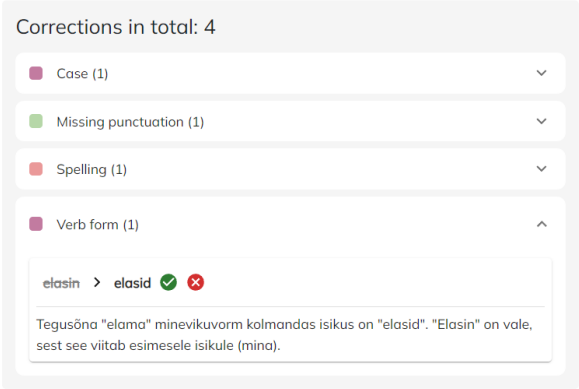

Figure 5: Sidebar displaying error categories and longer explanations under selected corrections.

D Example explanations

We rated the quality of system error explanations on the following scale: good — clearly presented and sufficient information; fair — partial or nonfluent but correct information; poor — use of incorrect statements and terms or edit description without additional context. Example (2) demonstrates a comprehensive and brief explanation annotated as good. Example (3) includes a brief explanation rated as fair and a longer explanation rated as poor.

[11] https://elle.tlu.ee/corrector-test

Instruction:
Reply with a corrected version of the input essay in Estonian with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original essay.

Input:
`{input}`

Response:
`{correction}`

Table 5: The GEC model M_1 fine-tuning prompt. The GEC instruction is a modification of the prompt by Coyne et al. (2023).

Instruction:
Sa võrdled kahte eestikeelset lauset: keeleõppija kirjutatud algne lause ja parandatud lause. Genereeri vea kaupa paranduste loend, kus on vealiik, algne tekst ja parandatud tekst.

Input:
`Algne lause: {original sentence}`

`Parandatud lause: {corrected sentence}`

Response:
`Parandused: {list of atomic edits}`

Table 6: The error detection and classification model M_2 fine-tuning prompt.

The correct word "terrassil" is in the adessive case form, not inessive, although both express location in Estonian. The brief explanation only considers the nominal form error, disregarding the spelling error (the base form is "terrass", not "terras").

(2) Source sentence: Pärast kontserdi me otsustasime juua kohvi restoranis ja koju minna jalgsi.
Target sentence: Pärast kontserti me otsustasime juua kohvi restoranis ja koju minna jala.
('After the concert, we decided to drink coffee in a restaurant and walk home.')

Explanation: kontserdi -> kontserti
Long: The word "pärast" ('after') requires the partitive case, so the correct form is "kontserti". The form "kontserdi" is in the genitive case and is not appropriate here.
Brief: The word "pärast" ('after') requires the partitive form "kontserti".
Error type: nominal form

(3) Source sentence: Linnas ma istun terras ja joon siider.
Target sentence: Linnas ma istun terrassil ja joon siidrit.
('In the city, I sit on a terrace and drink cider.')

Explanation: terras -> terrassil
Long: The inessive case form of the word "terrass" is "terrassil". It expresses location (where?).
Brief: The correct case form is "terrassil" (where?).
Error type: nominal form

Instruction:
Sa võrdled kahte eestikeelset lauset: keeleõppija kirjutatud algne lause ja parandatud lause. Sulle antakse paranduste loend, kus on vealiik, algne tekst ja parandatud tekst. Su ülesanne on selgitada ühte parandust. Selgita seda parandust, mis järgneb sõnale 'Selgitus'. Esiteks too välja põhjalikum selgitus, miks parandust vaja on. Teiseks anna lühike selgitus lihtsamas keeles. Pärast selgitust nimeta vealiik. Mitu vealiiki võivad kokku langeda. Omavahel seotud parandusi (näiteks sõnaühend, kus muutub mõlema sõna vorm) selgita koos. Sõnajärje parandusega kattuvaid muid parandusi selgita eraldi.

Input:
`{sentence pair, list of errors and input error to explain}`

Response:
`{explanation for input error}`

Table 7: The GEE model M_3 fine-tuning prompt.

Reply with a corrected version of the input text in Estonian with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the input text. There is one example of the task.

Input text: `{example input paragraph}`
Corrected: `{example corrected paragraph}`

Input text: `{input paragraph}`
Corrected:

Table 8: The 1-shot prompt used for GEC with GPT-4o. The example was randomly sampled from the EKI-L2 set. The GEC instruction is a modification of the prompt by Coyne et al. (2023).

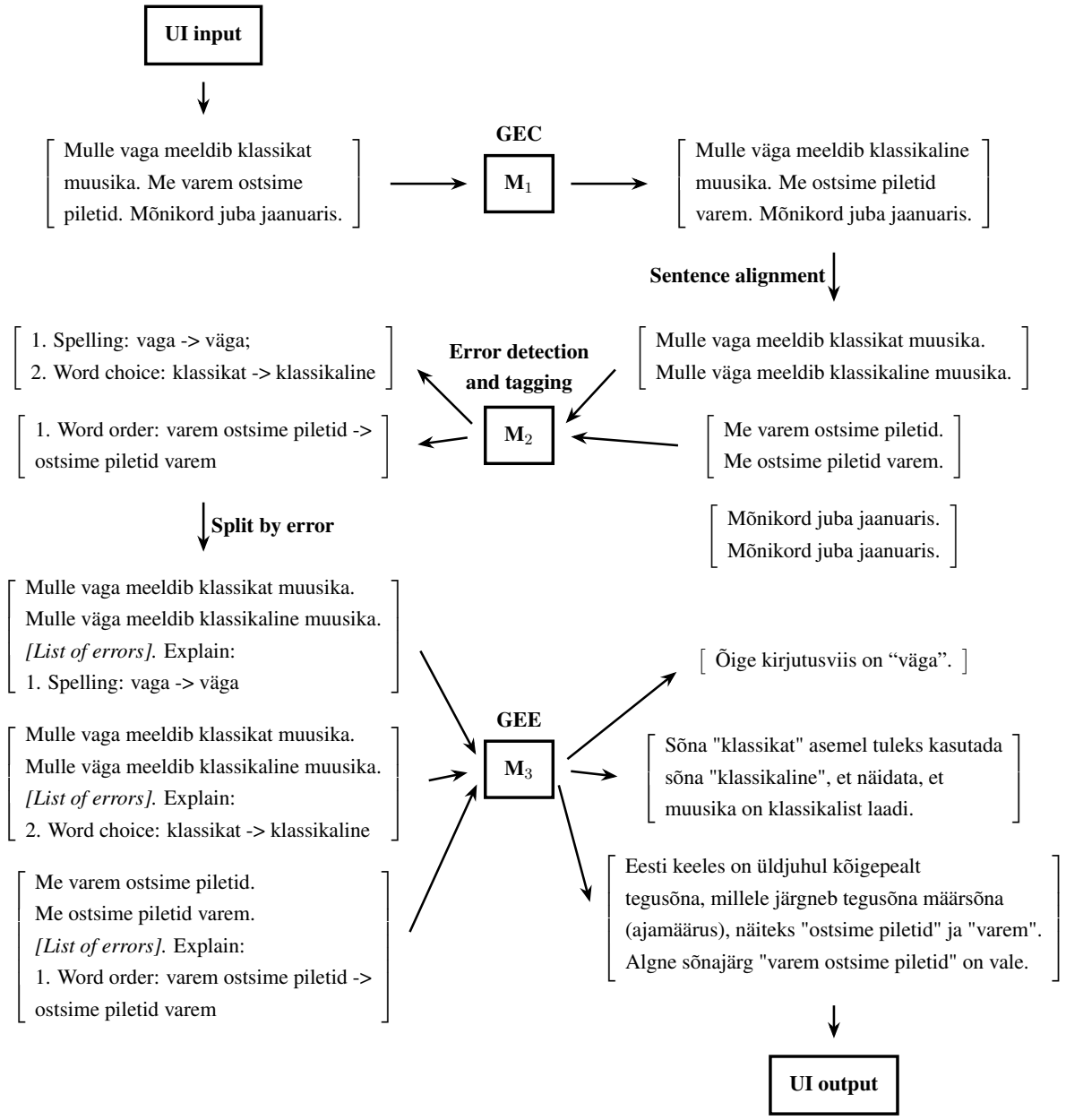

Figure 6: A detailed overview of the grammatical error correction and explanation system with an example. M denotes model.