Structured Reasoning for Grammatical Feedback -A MAC-T5 Hybrid Approach-

Shusuke Hashimoto
Indiana University Bloomington
Department of Linguistics

Abstract

Grammatical error correction (GEC) systems have become increasingly accurate, yet most still prioritize surface-level corrections over pedagogically meaningful feedback. This paper introduces a hybrid model, **MAC-T5**, that integrates symbolic reasoning with natural language generation to produce step-by-step grammatical explanations tailored to second language learners. Built on the Memory, Attention, and Control (MAC) network and the T5 encoder-decoder architecture, our model generates feedback aligned with a four-step chain-of-thought (CoT) structure. Using transcribed learner speech from the SLaTE 2025 Shared Task, we train and evaluate MAC-T5 in few-shot settings (one- and two-shot per error type), comparing it to a standard Transformer-based T5 model. Evaluation combines BERTScore for semantic similarity and a rubric-guided large language model (LLM) comparison for pedagogical clarity. While T5 yields higher scores across current metrics, MAC-T5 exhibits sharper gains with additional data, suggesting its potential to outperform when scaled. Our results highlight the challenges and opportunities in building explanation-oriented GEC systems, paving the way toward more interpretable and instructionally effective feedback tools.

1 Introduction

Grammatical Error Correction (GEC) systems have progressed significantly with neural architectures, achieving near-human accuracy on benchmark datasets. However, most models focus solely on producing corrected sentences, offering little insight into the grammatical rules being violated. In educational contexts, this limits their value—learners benefit more when feedback not only corrects but also explains the reasoning behind the correction.

Recent work has aimed to make GEC more interpretable through tagging-based models and error-type annotations. Yet even these improvements fall short of generating fluent, explanatory feedback that mimics instructor-like guidance. To address this, we propose **MAC-T5**, a hybrid model that integrates the structured reasoning capabilities of the MAC (Memory, Attention, Control) network with the expressive generation abilities of T5.

MAC-T5 is designed to produce chain-of-thought (CoT) feedback in four steps, each aligned with a reasoning phase: identifying the error, applying the correction, explaining the grammatical rule, and reviewing the fix. Our model is trained and evaluated using learner speech data from the SLaTE 2025 Shared Task, which includes automatic speech recognition (ASR), grammatical error correction (GEC), and grammar explanation feedback (GEF) components.

We compare MAC-T5 with a Transformer-based T5 baseline across one-shot and two-shot training settings. Evaluations are conducted using BERTScore for semantic similarity and a rubric-guided large language model (LLM) framework for instructional quality. While T5 currently performs better in both metrics, MAC-T5 shows sharper gains from data augmentation, suggesting strong potential for scaling and educational impact.

2 Related Work

2.1 Grammatical Error Correction Feedback

Grammatical error correction (GEC) has seen significant progress, evolving from rule-based systems to data-driven neural models. Early neural approaches typically framed GEC as a sequence-to-sequence translation task, mapping ungrammatical learner input to a corrected version using Transformer-based encoder-decoder models (Chollampatt and Ng, 2018; Kaneko et al., 2020). While these models achieved strong performance on standard benchmarks such as CoNLL-2014 and BEA-2019, they often lacked transparency—an essential feature in educational applications, where learners

benefit not only from corrections but from understanding the underlying rules.

Recent work has increasingly framed grammatical error correction (GEC) as a sequence-tagging problem, offering a more efficient and interpretable alternative to traditional sequence-to-sequence architectures. One of the most prominent models, GECToR, employs a Transformer encoder to predict token-level edit operations from a predefined tag set, including both surface-level actions (e.g., `REPLACE`, `DELETE`) and linguistically informed categories (e.g., `VERB:FORM`, `NOUN:NUM`), thus enabling direct and interpretable error correction (Omelianchuk et al., 2020). Building upon this foundation, subsequent work has introduced enhancements such as character-level edit encoders for non-autoregressive inference (Straka et al., 2021) and ensemble methods with knowledge distillation to improve robustness and coverage (Tarnavskyi et al., 2022). Together, these models demonstrate the scalability, efficiency, and pedagogical suitability of tagging-based approaches to GEC.

While grammatical error correction has been widely explored, we found that very few studies focus specifically on generating learner-oriented feedback that explicitly explains both what the error is and why it is incorrect. Although some related approaches exist in domains like medical reasoning or large language model interpretability, pedagogically motivated grammatical feedback remains underexplored. This study addresses that gap by proposing a reasoning-based framework designed to generate structured, instructive feedback tailored to second language learners.

2.2 Neural Reasoning and Chain-of-Thought Feedback

The growing interest in model interpretability and educational NLP has spurred new approaches to generating structured, logic-based justifications alongside predictions. One promising direction is the use of neural reasoning networks, which combine symbolic interpretability with neural flexibility. For example, Neural Reasoning Networks (Carrow et al., 2025) implement logical operations using weighted fuzzy logic, enabling both accurate classification and natural language explanations. Although originally developed for tabular classification tasks, these architectures are highly relevant for GEC, where many corrections correspond to explicit grammar rules that can be modeled compositionally.

Complementary to this, chain-of-thought (CoT) prompting has emerged as an effective method for eliciting reasoning in large language models. By including verbalized reasoning paths in few-shot prompts, models such as GPT-3 and PaLM are able to break down complex tasks into interpretable intermediate steps (Wei et al., 2023). While originally applied to arithmetic and commonsense tasks, CoT prompting has since been adapted to domains such as Compositional generalization (Zhou et al., 2023), and provides a compelling model for producing reasoning-aware grammatical feedback.

Another complementary line of work comes from compositional attention networks, particularly the MAC (Memory, Attention, Control) architecture (Hudson and Manning, 2018). MAC networks maintain a multi-hop control state that guides structured reasoning over input sequences, making them well-suited to step-by-step explanation tasks. Extensions of MAC have been explored in neuro-symbolic concept learning (Mao et al., 2019) and VQA tasks with disentangled reasoning (Johnson et al., 2017), and their architecture closely aligns with our model's goal of separating error detection, grammatical rule inference, and correction generation.

Our work draws on these traditions by combining MAC-style reasoning with CoT feedback generation, enabling the system to simulate the structure and pedagogical clarity of instructor-like grammatical explanations.

3 Methodology

3.1 Datasets

This study uses the dataset provided by the SLaTE 2025 Shared Task, which introduces a pipeline for assessing and improving spoken learner English. As illustrated in Figure 1, the pipeline begins with automatic speech recognition (ASR), which transcribes learner audio into text. From there, two branches emerge: one leading to second language acquisition (SLA) scoring and the other to grammatical error correction (GEC) followed by grammar explanation feedback (GEF). While SLA is a critical component for learner assessment, it is not used as input to downstream tasks. This study does not involve SLA scoring, but instead focuses exclusively on the GEC and GEF components, which operate directly on the ASR outputs. Although we use the ASR-generated transcriptions as input, no

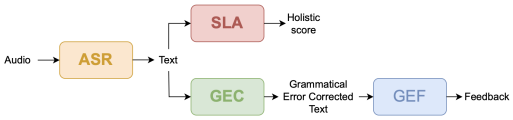

Figure 1: The pipeline of the tasks in SGECF (Spoken Grammatical Error Correction Feedback).

ASR fine-tuning or modifications were performed in our setup.

We use the Whisper-transcribed output as the text input for downstream components. The file contains metadata such as speaker ID, word-level timing, duration, and token labels, making it suitable for aligning speech with text for grammatical error processing.

The grammatical error correction (GEC) module was implemented through three stages. First, a disfluency detection component was used to remove hesitations, filled pauses, and self-corrections from the raw ASR transcripts. Next, token-level alignment and sentence reconstruction were performed to align corrected text with the original transcript, enabling precise detection and localization of grammatical errors. Finally, a rule-based tagging and serialization procedure was applied to classify each identified error and convert it into a structured format suitable for feedback generation. This process resulted in a dataset containing 11,655 annotated error instances.

Each entry in the dataset includes a source sentence (the learner's original utterance), the corresponding corrected version, an error tag representing the general type of grammatical error (e.g., VERB:TENSE, DET:ART), the error phrase that was incorrect, and the correction phrase that resolves it. Since a single sentence can contain multiple errors, it may generate multiple instances in the dataset.

To support different feedback models, two types of feedback were generated using OpenAI's API. The first type is reference feedback, a short paragraph offering a clear, learner-friendly explanation of the grammatical issue, the rule it violates, and how to correct it. These are intended for use with baseline transformer models such as T5. The second type is chain-of-thought (CoT) feedback, consisting of step-by-step reasoning about the error and its correction. This feedback is exclusive to the MAC-T5 hybrid model, where the intermediate logical structure is essential for enabling reasoning-driven generation.

The dataset was split into training, development, and test subsets. For training, we selected either 50 or 100 instances from the full set of 11,655 entries. These configurations are referred to as one-shot and two-shot settings, respectively, with each error type represented by one or two instances. While five- or six-shot learning per error type was feasible, we restricted our experiments to one- and two-shot settings due to limited annotation capacity. Both the development and test sets consist of 50 instances, with one example per error type. These evaluation sets remain fixed across all training configurations to ensure consistency in model comparisons.

3.2 Evaluation

To assess the quality of generated grammatical feedback, we employed both language-model-based qualitative evaluation and automated similarity metrics. Our objective was not only to evaluate whether feedback was grammatically correct, but also to determine how clearly and effectively it communicated the reasoning behind corrections—an essential component in educational feedback.

LLM-Based Evaluation

We implemented a rubric-guided evaluation framework using large language models (LLMs) to compare the explanatory feedback produced by two models: a Transformer-based baseline and our MAC-T5 reasoning model. For each grammatical error instance, the LLM received the following contextual information: the error tag (e.g., `VERB:TENSE`), the original erroneous sentence (`source`), the corrected sentence (`corrected`), and two anonymized feedback explanations—one from each model.

The LLM was then prompted to select the more pedagogically helpful explanation based on a rubric that emphasized clarity, grammatical insight, and learner-oriented reasoning: "Explains not only what is wrong, but also why it is wrong, making feedback more educational and learner-friendly." Originally framed as a binary comparison between the two outputs, the task was later expanded to include a "Neither" option to account for cases where both feedback explanations lacked sufficient quality or interpretability.

This method leverages the reasoning capabilities of LLMs to evaluate the educational value of feedback rather than relying solely on surface-level textual similarity. By aligning evaluation with human instructional criteria, this approach provides a

more nuanced and learner-relevant assessment of model performance.

BERTScore Evaluation

In parallel with LLM-based judgments, we calculated BERTScore to obtain a quantitative measure of semantic similarity between generated feedback and reference explanations. BERTScore computes token-level similarity using contextualized embeddings from pretrained language models, making it more flexible than traditional metrics like BLEU, which rely on exact n-gram matches.

While BERTScore does not evaluate whether the explanation is pedagogically useful, it serves as a complementary metric by quantifying how closely the generated feedback aligns in meaning with reference sentences. In this way, BERTScore offers insight into the degree of semantic faithfulness, even if it cannot capture instructional value or reasoning quality.

BLEU, on the other hand, was considered but ultimately deemed less appropriate due to its emphasis on surface form. Since effective feedback may vary greatly in structure and phrasing while still conveying correct grammatical reasoning, BLEU's rigid token-level matching often penalizes valid variation, leading to misleading or artificially low scores.

Taken together, our dual evaluation strategy—LLM-based comparison for qualitative feedback quality and BERTScore for semantic alignment—enables a more comprehensive assessment of the models' abilities to produce clear, informative, and instructionally meaningful feedback.

4 A MAC-T5 Hybrid Model

The MAC-T5 hybrid model is designed to generate pedagogically grounded grammatical feedback through explicit reasoning. It combines the representational strength of T5 with the structured, stepwise attention mechanism of the MAC (Memory, Attention, Control) network. The overall architecture consists of three components: a T5 encoder, a MAC reasoning loop, and a T5 decoder, each adapted for the feedback generation task.

4.1 T5 Encoder with Prompt Injection

The input to the model is a structured instance describing a grammatical error, including fields such as the error tag, error phrase, source sentence, correction, and corrected sentence. Instead of feeding these as raw key-value pairs or metadata-like structures, we apply prompt injection to convert the instance into a human-readable natural language prompt. This transformation is motivated by the fact that T5 was pretrained on natural language tasks such as summarization and translation; therefore, encoding the input in a naturalistic form aligns better with the model's inductive biases and improves representation learning. For example, the key-value tuple "error_tag": "VERB:TENSE" is converted into a readable form such as "The error involves verb tense.", making the input more interpretable and consistent with T5's pretraining objectives.

4.2 MAC Reasoning Loop

After encoding, the contextualized input representations are passed through a MAC-style multi-hop reasoning loop, which is responsible for guiding the model through a structured interpretive process. The MAC loop operates across four fixed reasoning steps, each step attending to a distinct aspect of the grammatical feedback process. Unlike attention in standard Transformers, which is uniformly soft and jointly distributed, the MAC reasoning loop applies hard masking to selectively isolate different input components at each reasoning step. Specifically:

Step 1: Error Attention

The model focuses exclusively on understanding the nature of the error. At this stage, fields related to the correction—such as the corrected sentence and minimal fix—are masked to ensure that the model reasons solely about what kind of error is present, guided by the error tag and erroneous phrase.

Step 2: Correction Attention

The model then attends to the correction information. Certain error-related metadata are masked to shift attention toward understanding how the correction resolves the problem introduced in the original sentence.

Step 3: Reasoning

In this step, the model integrates prior steps and begins constructing a high-level explanation of why the error is grammatically problematic. All inputs are visible, but emphasis is controlled via MAC's attention gating to refine the internal memory.

Step 4: Reviewing

The final reasoning step simulates a "review" phase where the model verifies and consolidates

its explanation, resolving ambiguities and preparing the output for generation. This mirrors how a human teacher might internally validate their explanation before providing feedback.

Each of these four reasoning steps corresponds directly to a subsection of the CoT (Chain-of-Thought) feedback, which is structured into four pedagogically motivated segments: (1) identifying the grammatical issue, (2) locating the specific error, (3) explaining the rule that was violated, and (4) describing the correction. During training, we align each reasoning step with the corresponding segment of the CoT explanation. That is, in epoch step 1, the model sees only the first segment of the CoT feedback ("what kind of grammatical issue is involved"); in step 2, it sees the second segment, and so on. This mapping not only enforces modularity in the learning process but also allows the model to learn compositional reasoning in a controlled and interpretable fashion.

4.3 T5 Decoder

After completing all reasoning steps, the final memory state and contextual representations are passed to the T5 decoder, which generates the full feedback explanation in natural language. Unlike standard grammatical error correction models that rewrite the sentence, our model outputs a fluent and instructive explanation of the correction. This explanation is learner-facing and designed to be pedagogically helpful, mimicking the type of reasoning a human instructor might provide in a classroom.

This hybrid architecture enforces a separation between "reasoning" and "explanation". While traditional Transformers implicitly mix these two processes, MAC-T5 disentangles them: the MAC loop performs structured internal reasoning, while the T5 decoder handles linguistic generation. This architectural choice is critical in enabling the model to produce clear, interpretable, and educationally meaningful feedback.

5 Results

LLM-Based Evaluation

Figure 2, illustrated in the horizontal bar chart below, reveal several key observations.

The T5 model consistently outperforms the MAC-T5 hybrid model in both one-shot (50-instance) and two-shot (100-instance) settings. Specifically, T5 receives 23 votes in the one-shot condition and 19 votes in the two-shot, while MAC-

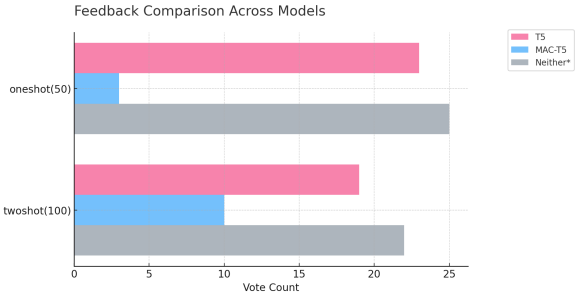

Figure 2: The number of votes on the test set when training with n-shot.

T5 earns 3 and 10 votes respectively. This outcome may appear counter-intuitive, as MAC-T5 is designed to follow rule-based reasoning via Chain-of-Thought (CoT) style feedback, which theoretically should make it more robust in few-shot learning contexts. Unlike T5, which learns feedback generation through pattern recognition, MAC-T5 explicitly reasons about grammatical rules—hence it is assumed to rely less heavily on large training sets. However, the gain in MAC-T5's performance from 3 to 10 votes between one-shot and two-shot settings does suggest that the model benefits from data augmentation and is potentially capable of outperforming T5 with additional supervision.

What stands out most in this analysis is the high proportion of "Neither" votes in both cases, dominating the other two choices. This suggests that the outputs generated by both models often lack clarity or relevance from the perspective of feedback quality. One likely interpretation is that both models are struggling to grasp the intended task, either due to limited data or because the prompt–output mapping is too complex to learn effectively in the current configuration.

The qualitative results complement these findings. In the one-shot scenario, feedback is frequently incoherent or meaningless. Many responses are empty or consist of repeated denials ("no," "no," "no"), while some attempt to identify the error but end up repeating the same phrase multiple times without offering a useful explanation (e.g., "the correct form is 'people', not 'people'"). Even when the error is identified, the explanation tends to lack precision or relevance, as seen in "The correct form is 'would' when describing a task or a task."

When the training data is doubled (two-shot), the generated feedback exhibits clearer signs of understanding the assigned task. For instance, the

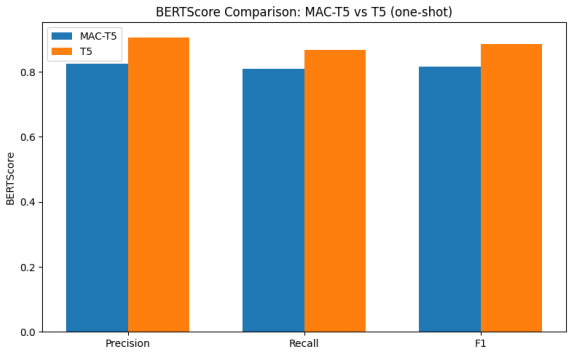

Figure 3: BERTScore comparison between MAC-T5 and T5 (one-shot setting).

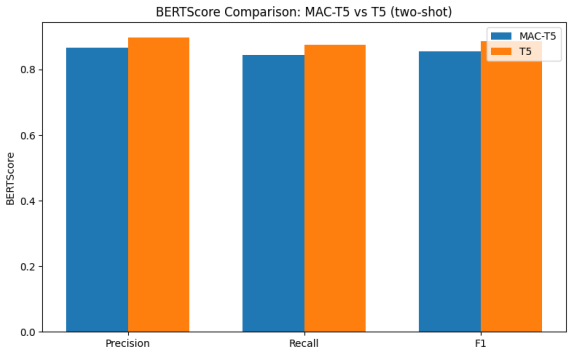

Figure 4: BERTScore comparison between MAC-T5 and T5 (two-shot setting).

Model	Precision	Recall	F1 Score
MAC-T5 (one-shot)	0.8256	0.8085	0.8168
T5 (one-shot)	0.9067	0.8676	0.8866

Table 1: BERTScore results for MAC-T5 and T5 models in the one-shot (50-instance) setting.

Model	Precision	Recall	F1 Score
MAC-T5 (two-shot)	0.8672	0.8436	0.8551
T5 (two-shot)	0.8986	0.8760	0.8870

Table 2: BERTScore results for MAC-T5 and T5 models in the two-shot (100-instance) setting.

model begins to explicitly identify the nature of the error—"the incorrect use of the verb 'would' when referring to a particular subject"—and connects it to the learner's sentence in a relatively intelligible manner. Still, there are remnants of copying the input text into the feedback itself, which compromises its usefulness. This copying behavior, where the original erroneous sentence is simply pasted into the feedback, signals that the model is still relying heavily on surface-level patterns rather than fully internalizing the feedback generation goal.

Together, these findings indicate that while both models face considerable challenges, the MAC-T5 hybrid model shows potential for improved grammatical feedback generation with additional training data and better representation of reasoning steps. Moreover, the high "Neither" vote count and qualitative errors stress the importance of refining task definitions, input formatting, and training strategies for this type of educational feedback generation.

BERTScore-Based Evaluation

To complement the rubric-guided LLM-based evaluation, we also assessed the semantic similarity between generated feedback and reference explanations using BERTScore. While BERTScore does not measure pedagogical quality or instructional clarity, it offers a robust quantitative metric for evaluating how semantically aligned the generated feedback is with reference feedback.

As shown in Figure 3 and Figure 4, the T5 model outperforms the MAC-T5 hybrid model across all metrics (Precision, Recall, and F1) in both one-shot and two-shot settings. In the one-shot scenario, T5 achieves a BERTScore F1 of 0.8866, compared to MAC-T5's 0.8168. Similarly, in the two-shot setting, T5 yields an F1 of 0.8870, while MAC-T5 reaches 0.8551.

This trend mirrors the LLM-based evaluation results, reinforcing the conclusion that T5 is more adept at producing outputs semantically close to the reference feedback. However, the performance gap narrows as the training data increases, suggesting that MAC-T5 benefits from additional supervision and may eventually rival T5 in semantic similarity metrics.

Although BERTScore effectively captures content similarity, it does not account for the educational value or logical coherence of feedback. For example, feedback that simply repeats the original sentence or includes irrelevant information may still yield high BERTScore if the reference contains similar wording. Therefore, while informative, these scores must be interpreted with caution and in conjunction with qualitative assessments and human-centered evaluation methods.

6 Discussion

While the results section has demonstrated performance gaps and progress between the T5 and MAC-T5 models under few-shot settings, several key challenges emerged from both empirical outcomes and architectural decisions that warrant further investigation.

Optimizing Chain-of-Thought (CoT) Feedback Design

Our current approach generates CoT feedback in four steps by prompting the OpenAI API with four distinct guiding questions. Although this setup aligns with the MAC-T5's multi-step reasoning structure, it remains unclear whether four is the optimal number. Increasing the number of reasoning steps may yield richer and more fine-grained feedback, potentially enhancing interpretability and correctness. Conversely, reducing the number of steps could generalize reasoning across error types, aiding few-shot generalization. A promising future direction would be to implement a dynamic framework that determines the optimal number of reasoning steps per instance during training—adjusting granularity based on error complexity and context.

Input Masking Strategy and Reasoning Focus

The input masking strategy plays a pivotal role in guiding the model's attention throughout each reasoning step. In this study, we synchronized the structure of masking with the four-step CoT format: (1) error identification, (2) correction focus, (3) justification, and (4) holistic review. However, alternative configurations could better leverage the model's reasoning capabilities. For example, allowing the reasoning element to appear earlier (e.g., alongside error identification) may help reinforce pedagogical clarity and encourage more abstract reasoning. Since masking and CoT feedback are interdependent, future work should explore joint optimization strategies that adaptively coordinate both structures to maximize learning efficiency.

Architectural Considerations: Balancing Reasoning and Explanation

The MAC-T5 hybrid model aspires to unify the strengths of symbolic reasoning and fluent explanation. The MAC reasoning loop enforces stepwise attention and logic, while the T5 components translate reasoning into natural language. However, over-reliance on CoT inputs can overload the decoder with summarization demands, potentially weakening overall output coherence. Pretraining MAC-T5 on structured feedback data or simulated CoT outputs may help align expectations across components and reduce this burden. This could better prepare the decoder for producing fluent, structured feedback, especially when data is scarce.

Data Augmentation and Model Scalability

The quantitative and qualitative results show clear signs that MAC-T5 benefits more from data augmentation than the baseline Transformer model. This supports the hypothesis that rule-based hybrid systems require more data to stabilize internal memory and reasoning paths. While MAC-T5 initially underperforms, its faster improvement rate implies strong potential if scaled with sufficient annotated data. However, as annotation cost is high, future work must explore efficient bootstrapping strategies—such as semi-supervised learning or active learning frameworks—to expand data coverage while reducing manual effort.

7 Conclusion

This study introduced a MAC-T5 hybrid architecture for generating pedagogically meaningful grammatical feedback through structured reasoning. By integrating a MAC-style multi-hop reasoning loop with a T5 encoder-decoder, the model aims to provide not just corrections but instructive explanations aligned with how language learners benefit from feedback. We leveraged Chain-of-Thought (CoT) feedback structured in four steps and designed a masking strategy to align input attention with reasoning phases.

Our experiments under one-shot and two-shot learning conditions revealed that while the baseline T5 model consistently outperforms MAC-T5 in both LLM-based and BERTScore evaluations, MAC-T5 shows promising gains with data augmentation. The results indicate that MAC-T5 can benefit substantially from increased supervision and more refined reasoning inputs. However, the high proportion of ambiguous outputs and "Neither" votes in LLM assessments highlights the complexity of reasoning-aware feedback generation.

In addition to quantitative evaluations, our study surfaced architectural and procedural challenges—particularly in CoT feedback design, input masking, and balancing the cognitive burden between reasoning and explanation components. Addressing these issues will be critical to realizing the full potential of reasoning-augmented models in educational NLP.

Future work will explore dynamic CoT step selection, adaptive masking strategies, and pretraining strategies tailored to reasoning tasks. Given the annotation bottleneck, scalable feedback generation also calls for leveraging data-efficient learning

techniques. Through these improvements, we hope to move closer to building models that not only correct learners' grammar but also teach them the underlying rules in a transparent and cognitively aligned manner.

References

Stephen Carrow, Kyle Erwin, Olga Vilenskaia, Parikshit Ram, Tim Klinger, Naweed Khan, Ndivhuwo Makondo, and Alexander G. Gray. 2025. Neural reasoning networks: Efficient interpretable neural networks with automatic textual explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(15):15669–15677.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. *Preprint*, arXiv:1801.08831.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. *Preprint*, arXiv:1803.03067.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. Inferring and executing programs for visual reasoning. *Preprint*, arXiv:1705.03633.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Milan Straka, Jakub Náplava, and Jana Straková. 2021. Character transformations for non-autoregressive GEC tagging. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 417–422, Online. Association for Computational Linguistics.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Appendix

1. Prompt for Reference Feedback

You are a grammar feedback assistant. Your task is to write a short but clear sentence that explains what was grammatically wrong in the learner's sentence and how to fix it.

Your explanation must follow these rules:

- **Explain the general grammar rule** that was broken. - **Do NOT just point out the error or repeat the correction.** - Make the feedback useful for learners so they can apply the rule in other situations. - Use clear and simple language — avoid technical jargon if possible. - Keep your response under 80 words. - Just focus on the part of the sentence that the `error_phrase` and `correction` refer to. **Do NOT mention other errors** that are not part of the specified correction.

Here is an example of high-quality feedback that follows these guidelines:

Example:

```
{
  "error_tag": "U:DET",
  "error_phrase": "the",
  "correction": "",
  "source": "having the fitness classes is a very critical thing...",
  "corrected": "having fitness classes is a very critical thing..."
}
```

Feedback: The error in your sentence is the unnecessary use of the article `"the"` before the plural noun phrase `"fitness classes"`. In English, plural or non-specific nouns used in a general sense do not typically require a definite article. You should say `"having fitness classes,"` not `"having the fitness classes."`

Now follow the same format to give feedback for the input below:

```
{
  "error_tag": "{item['error_tag']}",
  "error_phrase": "{item['error_phrase']}",
  "correction": "{item['correction']}",
  "source": "{item['source']}",
  "corrected": "{item['corrected']}"
}
```

2. Prompt for Chain-of-Thought (CoT) Feedback

You are a reasoning assistant helping generate step-by-step grammatical explanations for a grammar correction model.

Given the following structured input, output a 4-step explanation that describes: 1. What kind of grammatical issue is involved (based on the error tag) 2. What exactly is wrong in the sentence 3. Why it is wrong — explain the grammatical rule that was broken so the learner can avoid the mistake in the future. Do not say just that it's not appropriate or not correct. 4. How it is corrected

Important: - Just focus on the corrected part. Do not mention errors that are not highlighted. - Do not simply restate the correction. Ground your explanation in general grammar rules. - Avoid technical jargon, but be clear and instructive.

Here is a sample feedback written by a teacher that explains the same correction:

```
"""{ref_feedback}"""
```

Input:

```
{
  "error_tag": "{item['error_tag']}",
```

```
  "error_phrase": "{item['error_phrase']}",
  "correction": "{item['correction']}",
  "source": "{item['source']}",
  "corrected": "{item['corrected']}"
}
```

Output:

3. Prompt for LLM-Based Evaluation

You are evaluating two different pieces of feedback that correct a student's English sentence.

Each feedback aims to explain: - What is wrong - Why it is wrong - And help the student learn the grammar better.

Here is the information for the case:

- Error Tag: `t_instance['error_tag']`
- Error Phrase: `t_instance['error_phrase']`
- Correction: `t_instance['correction']`
- Source Sentence: `t_instance['source']`
- Corrected Sentence: `t_instance['corrected']`
- Reference Explanation: `t_instance['ref']`

Now, here are two feedback explanations to compare:

Feedback A (Transformer-based):

```
{t_instance['generated_feedback']}
```

Feedback B (MAC-T5-based):

```
{m_instance['generated_feedback']}
```

According to the following rubric:

> "Explains not only what is wrong, but also why it is wrong, making feedback more educational and learner-friendly."

Which feedback is better? Answer only with one word: `"A"` or `"B"`.

4. MAC-T5 Hyperparameter Ranges

	Component	Details
MAC-T5 Configuration	Embedding Dimension	512 (MAC and T5)
	Reasoning Steps	4
	Encoder-Decoder Backbone	`t5-small`
	Tokenizer	`T5Tokenizer.from_pretrained`
	Epochs	10 (one-shot), 15 (two-shot)
	Batch Size	2 (one-shot), 4 (two-shot)
	Learning Rate	5e-5
	Optimizer	AdamW
	Max Input Length	512
	Max Decoded Length	150

T5 Baseline Configuration

Component	Details
Pretrained Model	`t5-small`
Tokenizer	`T5Tokenizer.from_pretrained`
Epochs	10 (one-shot), 15 (two-shot)
Batch Size	2 (one-shot), 4 (two-shot)
Learning Rate	5e-5
Optimizer	AdamW
Max Input Length	512
Max Decoded Length	150

5. Input/Output Examples

Due to length, full training and evaluation examples from both MAC-T5 and T5 models under one-shot and two-shot settings are not included here. You can view the details in the project repository: github.com/shuhashi0352/Structured-Reasoning-for-Grammatical-Feedback–A-MAC-T5-Hybrid-Approach-.