DIMSIM: Distilled Multilingual Critics for Indic Text Simplification

Sneha Mondal[1], **Ashish Agrawal**[3*], **Ritika**[2*], **Preethi Jyothi**[3], **Aravindan Raghuveer**[1]
[1]Google DeepMind [2]Google [3]IIT Bombay
{snehamondal, ritikagoyal, araghuveer}@google.com
{ashishagrawal, pjyothi}@cse.iitb.ac.in

Abstract

Self-correction techniques have recently emerged as a promising framework to improve the quality of responses generated by large language models (LLMs). Few-shot prompted LLMs act as critics to produce feedback for an input, which is further fed to a refiner (also an LLM) to produce an output. However, these critique-refine steps require multiple expensive LLM calls. To circumvent this large inference cost, we borrow inspiration from prior work on knowledge distillation and propose the use of critique distillation to train critic models. These are smaller sequence-to-sequence models that are trained on input-critique pairs generated by an LLM. We focus on the problem of text simplification for three Indian languages: Hindi, Bengali and Marathi. This task is a good fit for self-correction style techniques. It also has not been systematically explored for Indian languages before. We train two separate critics that focus on lexical and structure complexity, and show that it is surprisingly more effective than using an LLM directly as a critic in both 0-shot and few-shot settings. We also show the benefits of training multilingual critics, as opposed to monolingual critics. Extensive human evaluations show that on average, raters find 80% of DIMSIM's output to be simple and easy to read.

1 Introduction

Large language models (LLMs) are well-equipped to generate high-quality responses using only a few task-specific examples in its input prompts (Brown et al., 2020; Chowdhery et al., 2023; Wei et al., 2022a; Min et al., 2022). However, these generations are prone to vulnerabilities such as "hallucinations" (Huang et al., 2023b; Mündler et al., 2023), biased content (Kotek et al., 2023; Wan et al., 2023) and incorrect reasoning (Zhang et al., 2023a; Huang and Chang, 2023), to name a few. A promising approach to fix inconsistencies in LLM outputs is *self-correction* specifically the critique-refine model (Bai et al., 2022; Nathani et al., 2023; Pan et al., 2023), where the LLM is given feedback or a critique about its outputs – either generated by the LLM itself or by humans – and asked to revise its outputs in accordance with the given feedback.

While self-correction can be effective (Huang et al., 2023a; Tang et al., 2019a), we incur substantial costs when invoking LLMs with over 500B parameters (Chowdhery et al., 2023). Each LLM call requires significant memory and compute utilizing at least 350GB GPU memory to serve a 175B LLM, which is far beyond the scope of what is affordable for most product developers and service providers. To address these computational constraints, knowledge distillation has been used in prior work (Tang et al., 2019a; Hsieh et al., 2023) where smaller models are trained with labels from a larger LLM. However, distillation within the critique-refine paradigm and more so for tasks using multilingual LLMs has not been explored yet.

In this work, we propose the use of *distilled trained critics* within the critique-refine framework for the problem of text simplification for three Indian languages. We chose the task of text simplification for two main reasons: 1. It is a form of textual style transfer and hence well-suited to the critique-refine paradigm. 2. There is no existing work on text simplification for Indian languages. In fact, multilingual text simplification has not been sufficiently well-explored in prior work (Ryan et al., 2023). Our work is an attempt to bridge this gap. Expanding the task to Indian languages like Hindi, Bengali and Marathi, which are not as well-represented as English in the LLM, also leads us to train multilingual critics for better generalization.

We assume a few-shot setting for Hindi with

[*] Equal contribution

Findings of the Association for Computational Linguistics ACL 2024, pages 16093–16109
August 11-16, 2024 ©2024 Association for Computational Linguistics

access to a small set of expert-written examples of text simplification in Hindi. We adopt the more challenging zero-shot setting for Bengali and Marathi where we assume no access to text simplification instances in these two languages and only make use of the Hindi examples.

We decouple text simplification along two natural axes -- structural and lexical complexity -- and design two separate critic modules that provide critiques specific to each of these two dimensions (§3.1). Each distilled critic is a smaller sequence-to-sequence model (e.g., mT5 (Xue et al., 2021)), trained using synthetic data created via few-shot prompting of the LLM followed by careful data filtering (§3.2).

Apart from critique generation, we also show the merits of both eager and lazy refinement where the LLM (used as a refiner) can either individually refine the outputs based on each trained critic's feedback or refine the output based on combined feedback from both critic modules, respectively (§3.3).

We evaluate our results using the standard SARI metric (Xu et al., 2016) for text simplification and also undertake detailed human evaluations. We find that trained critics, especially multilingual critics that are trained on data pooled together from Hindi, Bengali and Marathi, offer improved cross-lingual transfer to Bengali and Marathi. The trained critics result in significant performance improvements of up to 2.4 absolute SARI points for Hindi and up to 2 absolute points for Bengali and Marathi in the zero-shot setting, compared to using LLMs at test time. We call our model DIMSIM: **Di**stilled **m**ultilingual critics for text **sim**plification.

2 Related Work

2.1 Critique Refinement

Despite demonstrating impressive task performance (Guo et al., 2023; Suzgun et al., 2022) and reasoning abilities (Wei et al., 2022b; Kojima et al., 2022), LLMs are observed to exhibit undesired behavior such as hallucinations (Lin et al., 2021; Zhang et al., 2023b), unfaithful reasoning (Golovneva et al., 2022; Wu et al., 2023) and generating inappropriate or harmful content (Shaikh et al., 2022; Gehman et al., 2020; Levy et al., 2022). Self correction via critique feedback generated by the LLM (Schick et al., 2022; Madaan et al., 2023) is a very promising direction. (Pan et al., 2023) presents a comprehensive survey of approaches in this body of work. (Huang et al., 2022) uses chain of thought and self consistency to produce clean task training data that is to further fine tune the LLM. Self-Refine(Madaan et al., 2023) use the LLM iteratively as the generator, refiner and the critic for 7 tasks across dialog generation, coding, reasoning and constrained generation. (Lahoti et al., 2023) argue that instead of sequential iterations, diverse sampling of critiques, refinement followed by voting produces better results. Multi-Aspect Feedback (Nathani et al., 2023) argues that breaking down critiquing into specialized feedback modules provides better performance than relying on one single generic feedback source. Our approach is similar in that we use specialized critics for various aspects of text simplification. REFINER (Paul et al., 2023) shows that critiquing the intermediate reasoning steps in the chain of thought can improve performance.

2.2 LLM Distillation

One of the main practical limitations of LLMs is the compute and latency cost incurred during inference by virtue of their complex architectures and vast parameter space. Knowledge distillation (Hinton et al., 2015; Tang et al., 2019b; Liang et al., 2020; Fu et al., 2023; West et al., 2021; Li et al., 2022). reduces the serving cost concerns by learning a small “student” model to mimic the behaviour of a larger “teacher" model, an LLM in our case. LLMs have been shown to produce very detailed reasoning and rationale steps for the outputs they produce (Wei et al., 2022b; Kojima et al., 2022). A recent direction of research has started to show that distilling using teacher generated rationales can be very effective (Ho et al., 2022; Wang et al., 2022; Magister et al., 2022; Li et al., 2023). The proposed approach in this paper shares the same spirit. However to the best of our knowledge, we are the first to propose using rationale distillation as a way of training critic models. We show, similar to very recent work (Hsieh et al., 2023), that the distilled critic is able to surpass the performance of the teacher LLM while an order of magnitude smaller.

2.3 Text Simplification

Text simplification is a well-studied NLP problem (Zhu et al., 2010; Xu et al., 2015; Alva-Manchego et al., 2020; Al-Thanyyan and Azmi, 2021) that deals with reducing the complexity of text without compromising on its meaning. This task has been commonly modeled either as a super-

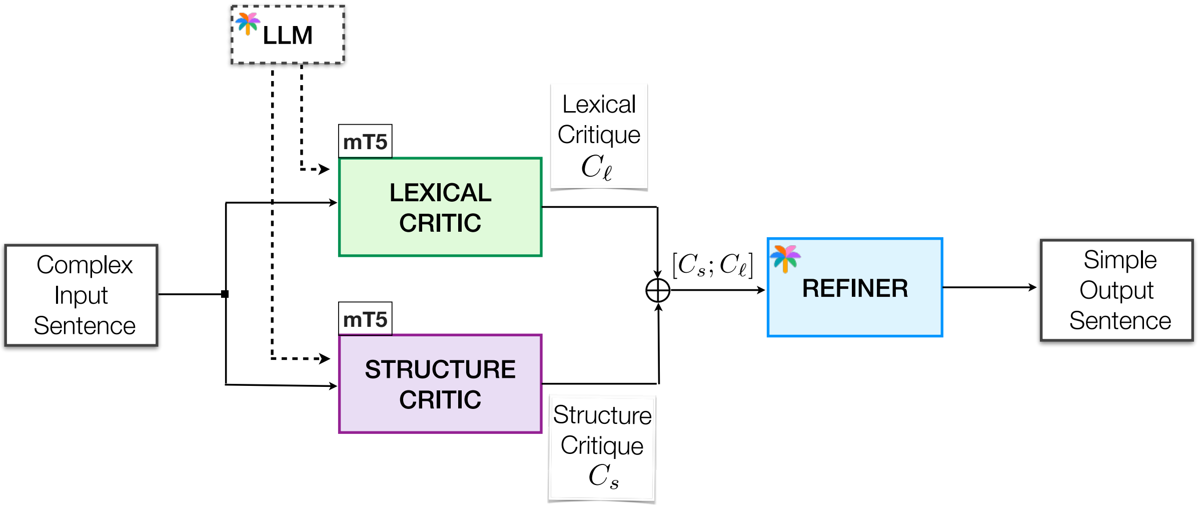


Figure 1: Main components of DIMSIM in the lazy mode. PALM 2 is used to generate synthetic data to train lexical and structure critic modules. Critiques C_ℓ and C_s from separate modules are concatenated and passed to the refiner LLM (also PALM 2), as part of its prompt to produce the final output.

vised sequence-to-sequence problem (Nisioi et al., 2017; Zhang and Lapata, 2017; Mallinson et al., 2020) or in an unsupervised setting with no access to parallel complex-simple sentence pairs (Surya et al., 2019; Kumar et al., 2020). Recent work on the BLESS benchmark (Kew et al., 2023) has comprehensively evaluated state-of-the-art LLMs for the task of text simplification and found them to be comparable with existing state-of-the-art baselines. This evaluation, however, was entirely on English datasets. Multilingual text simplification is relatively far less explored. As highlighted in (Ryan et al., 2023), text simplification research has become exceedingly English-centric in the last few years; the authors introduce a multilingual benchmark that covers text simplification for many diverse languages. With our work, we add to this existing multilingual benchmark for text simplification with datasets for three Indian languages, Hindi, Bengali and Marathi.

3 DIMSIM: Methodology

In this work, we present a critique-refine framework for text simplification (TS) using multilingual LLMs. Instead of a single critic that offers overall feedback for simplification, we decompose the simplification problem along two dimensions -- structural complexity and lexical complexity -- and we designate one critic to each dimension. The structure critic offers suggestions on how to alter the structure of a sentence, either by paraphrasing or splitting, to make it simpler. The lexical critic is more targeted in its critique by identifying archaic or difficult words in the text and offering simpler alternatives. Henceforth, we will collectively refer to these two modules as sub-critics. Finally, a refiner module processes the combined critique from the sub-critics to generate a simplified text output (Nathani et al., 2023).

To prime the sub-critics to offer informative critiques specific to each dimension, we need carefully designed few-shot examples. We create a corpus of 20 manually-curated input-output TS pairs for Hindi. These span a range of simplification operations, and are created by trained language experts. This method is expensive, tedious, and requires access to trained annotators. It does not readily scale to new languages. Therefore, we adapt the critique-refine framework to work in a zero-shot cross-lingual setting for Bengali and Marathi, using Hindi few-shot examples.

Figure 1 offers a schematic illustration of DIMSIM. Section 3.1 describes the key design choices we make to build the sub-critics. Section 3.2 further elaborates on generating synthetic data to train sub-critics. And, Section 3.3 describes how the refiner module processes these critiques to produce the final simplified text.

3.1 Design of Critic Modules

Critique generation is the first step of DIMSIM. As mentioned earlier, we choose two dimensions along which a text might be complex: structural complexity and lexical complexity. Decoupling the problem in this manner has multiple advantages.

Firstly, each sub-critic can now focus exclusively on a specific aspect of the problem and offer targeted critiques. Secondly, sub-critics can now be combined in different ways with the refiner, i.e., either sequentially after each critique is generated (*eager* refinement); or both the critiques are concatenated and a collective critique is sent to the refiner (*lazy* refinement). Lastly, decoupling critics allows for the effective use of multilingual data to learn language-agnostic properties of TS and enable better cross-lingual transfer.

The **structure critic** provides specific suggestions on simplifying the structure of the text, either by paraphrasing, removing repeated information, sentence-splitting, or a combination of the three. The **lexical critic** provides specific suggestions on simplifying words used in the text. It provides easier alternatives for words that are too difficult, archaic, or uncommon in spoken language. The alternative might be a simpler synonym, a well-understood English loanword, or a phrase that explains the meaning of the difficult word. The option of allowing popular English loanwords as replacements yields *code-mixed* sentences that are conversational, less formal, and typically simpler to comprehend than their monolingual counterparts.

In order to build strong sub-critics, we elaborate on two important design choices below.

1. Few-shot prompted vs. trained sub-critics. A natural choice for a critic is to prompt an LLM with few-shot examples. We provide examples of difficult sentences in the LLM's input prompt, along with structural or lexical feedback, and rely on its in-context learning ability to generalize to unseen instances at test time.

Alternatively, we can *train* models to generate structural or lexical feedback given a complex input sentence. This approach creates the need for a reasonably high-quality corpus consisting of sentence-critique pairs to train the sub-critics. We generate synthetic data via few-shot prompting of the LLM, followed by rigorous data filtering strategies to remove low-quality instances. The synthetic data is used to fine-tune a pre-trained multilingual model like mT5 (Xue et al., 2021) to create a trained sub-critic, which can be used at test-time to generate critiques for a test instance. Details about synthetic data creation filtering mechanisms are in Section 3.2. While there is a one-time overhead of training the sub-critics, this additional step is fully justified since we find trained sub-critics to produce better simplifications, especially in the cross-lingual setting. Using smaller trained sub-critics instead of a much larger LLM, also reduces the latency to generate critiques at test-time.

2. Multilingual vs. Single-language Critics. When training critics, we can either train language-specific sub-critics or pool all the data for a single dimension and train a multilingual sub-critic. There are trade-offs involved in each option. Language-specific sub-critics can capture nuances specific to a language, but can also overfit especially when there is limited diversity in the synthetic data. Multilingual sub-critics can learn language-agnostic TS patterns from data pooled across languages that might help cross-lingual generalization, but they are also prone to forgetting due to language interference. We find that multilingual critics outperform their single-language counterparts. This is also a more efficient solution, since we get performance gains from a single model per critic, instead of a model for every critic, language pair.

3.2 Data Pipeline to Train Critics

3.2.1 Synthetic Data Generation

For all three languages, we start with a monolingual corpus and use a few-shot prompted LLM to generate structural and lexical feedback. The few-shot prompt includes a brief task description, followed by k in-context examples. For Hindi, we set k to 7; for Bengali and Marathi, k is 4. For all three languages, the in-context examples are sampled from our manually curated set of 20 Hindi examples. For Bengali and Marathi, the prompt includes an explicit instruction that Hindi examples are illustrative only, and the LLM should generalize to the target language at inference. Since our critic modules are instructed to provide feedback about text complexity, the prompt also includes examples where the input text is already simple and does not require any modifications.

The few-shot prompt for the lexical critic is shown in Appendix C. The lexical critic itself contains two sub-modules. The first is a *classification* module that classifies every word in the input text as *easy* or *difficult*. The second is a *replacement* module that suggests a simpler alternative for difficult words identified by the classification module. The output is structured with one suggested replacement per line. While it is possible to merge the two components in a single prompt (i.e. identify difficult words and suggest alternatives), we find that breaking the task into separate components is specially crucial in the cross-lingual setting.

The few-shot prompt for the structure critic is shown in Appendix C. The prompt contains examples of inputs that require paraphrasing, deletion, or splitting to simplify. Examples contain a chain-of-thought style reasoning, followed by a suggested re-phrasing of the input. To prevent over-triggering, we also provide in-context examples where the input text is simple enough and requires no modifications.

3.2.2 Data Filtering

After generating synthetic data using few-shot prompts as outlined in Section 3.2.1, we adopt the following data filtering steps to maintain the quality of training instances:

- Remove empty or ill-formatted LLM generations.
- Ignore suggestions for lexical replacements when the suggested replacement is nearly identical to the original. This can happen when the LLM suggested replacement is just a different inflectional form of the original word instead of being a simpler substitute.
- Ignore suggestions for lexical replacements when the suggested replacement is for an English word. Since Indian languages have incorporated popular English loanwords as part of colloquial usage, we do not want to replace English words in the original text. We run a token-level language identification pass on the inputs, and remove suggestions where the original word is identified as English.
- Remove structural feedback if incorporating the feedback may lead to meaning loss or incoherent text. Since we do not have a way to directly evaluate the quality of the structure critique, we do it heuristically. The structural feedback is fed to the refiner module to obtain a likely simplified text. If the resulting simplified text has a compression ratio of more than 1.2 or less than 0.7, the corresponding structure critique is deemed to be of poor quality and removed from the training dataset.

3.3 Refiner Module

We re-use the LLM as a refiner, that takes the original complex text along with feedback from the critics as input and produces simplified text as output. We observe that existing LLMs (such as PALM 2) are proficient at incorporating feedback and simplifying the text, while also ensuring that the generated output is fluent, coherent and semantically similar to the input.

Given the presence of sub-critics in our framework, we have the option of two refinement strategies that we will refer to as *eager* and *lazy* refinement. In the eager refinement strategy, the text first undergoes structure simplification, followed by lexical simplification (or vice-versa). The refiner acts on a single critic's feedback one-at-a-time, thereby reducing the complexity of the refinement task. This can be especially important in the cross-lingual setting where the LLM is already burdened with generalizing to a language different from that of the in-context examples. In the lazy refinement strategy, we concatenate feedback from the structure and lexical sub-critics to create a single piece of composite feedback, which is then passed to the refiner. While this increases the complexity of the refinement task, it is more efficient than eager refinement because we incur only a single LLM inference call for refinement.

4 Experimental Setup

4.1 Datasets

Human-annotated test set. To the best of our knowledge, there are no existing TS datasets for Indian languages. As part of this work, we create and publish the first human-annotated TS datasets for Hindi, Bengali, and Marathi. All simplification metrics are reported on these datasets. We hope to be able to contribute these datasets to the multilingual benchmark MULTISIM (Ryan et al., 2023), that currently does not support any of these languages.

For Hindi, our dataset comes from two sources - WMT 2013, and scraped Wikipedia articles. News articles and Wikipedia pages are a natural source of complex and formal sentences, making them suitable for a TS task. We extract sentences and filter them for length between 5 and 30 tokens. We select 500 sentences from the WMT newscrawl corpus, and 541 sentences from Wikipedia articles. Each sentence is simplified and reviewed by trained Hindi language experts. For Bengali and Marathi, our dataset contains 500 sentences each, derived from the WMT 2013 and WMT 2018 newscrawl respectively.

Training dataset for critic modules. To train the lexical and structure critics, we start with 140,000 sentences from the WMT 2020 newscrawl for each language. We filter out sentences that are too long or too short, and retain sentences that contain between 5 and 30 tokens. Starting with this monolingual corpus, the training datasets are created as described in Section 3.2.

4.2 Metrics

To assess how well a model performs on TS, we evaluate it using a set of automatic metrics. Model outputs are evaluated along two dimensions - 1) simplification, and 2) semantic similarity. As is standard in prior work, simplification is measured using SARI (Xu et al., 2016). This is an n-gram based metric that compares model outputs against human references, and explicitly measures the goodness of words that are added, retained, and deleted by the model. Semantic similarity with the input sentence is measured using BERTScore (Zhang* et al., 2020). This metric matches words in the input and output text by computing cosine similarities between their contextual embeddings.

Both SARI and BERTScore are computed using the EASSE package (Alva-Manchego et al., 2019), and a higher score indicates better performance. However, previous work suggests that BERTScore has certain limitations: despite relying on contextual embeddings, it is biased towards rewarding sentences with higher lexical overlap. To contextualize and compare BERTScore between models, we establish a *skyline* by computing BERTScore between input sentences and human references. Since human references are expected to be largely meaning-preserving, the skyline establishes a fairly high bar. An ideal TS model should be able to simplify the input (=high SARI), without altering the meaning of the original text (=low deviation from BERTScore skyline).

Additionally, we report the percentage of model outputs that are identical to the input. For this metric, a lower score indicates better performance, since we want the model to simplify the input text instead of merely copying it.

It is worth noting that there is an inherent tradeoff between these metrics. For instance, models that tend to copy the input (i.e. higher %age of identical outputs) would naturally do a good job of meaning preservation, resulting in a higher BERTScore. However, copying would simultaneously result in a low SARI because the input text has not been simplified. In isolation, none of these metrics is a good indicator of model performance; they are meant to be assessed together.

Two prevalent TS metrics, FKGL (Kincaid et al., 1975) (a readability metric) and LENS (Maddela et al., 2023) (a learned metric), are not reported in our work because they are available only for English.

4.3 Baselines

In this work, we focus on comparing our method with existing approaches to self-refinement. Below we list baselines for each language.

4.3.1 Hindi

Few-shot: Input prompt includes a brief task description followed by 10 human-written in-context examples of {input, simplified output} pairs.

Chain-of-Thought (CoT): Input prompt includes a brief task description followed by 10 human-written in-context examples of {input, analysis, simplified output} tuples. The analysis provides feedback about lexical and structural complexity of the input. For instance, it highlights difficult words or repetitive phrases and offers simpler alternatives.

Critique-Revision: We use a "critique request" and "revision request" prompt, both implemented using standard few-shot prompting. The critic request prompt contains human-written examples of {input, analysis} pairs, whereas the revision request prompt contains human-written examples of {input, analysis, simplified output} tuples.

Collective-Critiques and Self-Voting (CCSV): This is a self-refinement strategy proposed by (Lahoti et al., 2023). It starts with an initial generation, followed by sampling multiple critiques and revision drafts from the LLM. Once multiple candidate revisions are available, authors prompt the LLM to self-select and vote on the best response. We extend this method for TS and design appropriate prompts for the task. A crucial difference is that while (Lahoti et al., 2023) propose 0-shot CCSV, we find that 0-shot prompting fails to produce meaningful critiques for TS. We instead implement 1-shot CCSV, with a single human-written example per prompt.

4.3.2 Bengali and Marathi

Cross-lingual few-shot: Same as few-shot prompting, except that the human-written in-context {input, simplified output} pairs are in Hindi instead of being in the target language.

Category	#LLM calls	Approach	SARI	% identical outputs	BERTScore
Few-shot prompting with a single prompt	1	Without chain-of-thought	27.66	94.52	92.71
		With chain-of-thought	40.21	40.82	90.64
Few-shot prompted critic + few-shot prompted refiner	2	Critique revision	38.56	34.58	91.05
Collective-Critiques and Self-Voting (CCSV)	4	-	41.96	23.05	86.3
Few-shot prompted sub-critics	3	Lazy refinement	41.88	40.73	91.35
	4	Eager refinement (lexical first)	41.36	34.10	89.82
	4	Eager refinement (structure first)	41.68	41.21	91.34
Trained sub-critics (DimSim)	1	Lazy refinement	**44.31**	31.41	91.63

Table 1: Results on Hindi TS for different strategies. BERTScore between inputs and human references: (*skyline*) = 92.71, identity SARI = 25.9.

Category	#LLM calls	Method	Bengali			Marathi		
			SARI	%age identical	BERTScore	SARI	%age identical	BERTScore
Few-shot prompting with a single prompt	1	Crosslingual	37.46	3.6	86.09	37.94	9.3	88.6
		Crosslingual + CoT	32.16	53.8	88.43	30.35	76.3	92.6
		Translation	37.96	2.8	85.91	38.69	4.7	87.5
		Translation + CoT	33.31	45.8	88.17	30.12	74.7	92.6
Crosslingual few-shot prompted sub-critics	3	Lazy refinement	40.49	8.2	86.98	39.40	20.3	90.0
	4	Eager refinement (structure first)	40.77	7.4	86.94	40.33	20.3	90.0
		Eager refinement (lexical first)	40.67	9.6	87.26	39.14	20.7	89.9
Trained sub-critics (DimSim)	1	Single-language	42.22	2.0	87.79	40.94	11.3	89.5
		Multilingual	**42.68**	2.1	87.80	**41.48**	9.0	89.6

Table 2: Results on Bengali (BERTScore *skyline* = 89.76, identity sari = 20.47) and Marathi (BERTScore *skyline* = 92.8, identity sari = 21.5)

.

Cross-lingual CoT: Same as Chain-of-Thought, except that the human-written in-context {input, analysis, simplified output} tuples are in Hindi instead of being in the target language.

Translation few-shot: Input prompt includes a brief task-description followed by machine-generated examples of {input, simplified output} pairs. These in-context examples are derived by translating both input text and simplified output from Hindi to Bengali or Marathi, using the Google Translate API.

Translation CoT: Input prompt includes a brief task-description followed by machine-generated examples of {input, analysis, simplified output} tuples. We use translated input and simplified outputs, as above. Hindi phrases and suggested word-replacements in the analysis string are also translated to Bengali or Marathi.

4.4 Implementation

We use the instruction-tuned PaLM 340 billion params model (Chowdhery et al., 2023; Longpre et al., 2023) as the base LLM for all experiments. For training the critic models, we use mT5-xxl as the base model (Xue et al., 2021). Inferences for all models, methods and baselines are performed using top-1 decoding at a sampling temperature of 0.7 and 256 decode steps. All numbers are reported over a single run.

5 Results

Tables 1 and 2 summarize our main results for Hindi, Bengali, and Marathi. For Hindi, DimSim outperforms all other methods, with a near 3 point improvement in SARI scores (44.31) compared to CCSV, the closest competitor (41.96). It does so while making only $\frac{1}{4}$ the number of LLM calls. Lazy refinement performs slightly better than eager refinement (in both directions). Critique refinement with a single critic is not very effective, and falls behind chain-of-thought prompting by roughly 2 SARI points.

Cross-lingual evaluations for Bengali and Marathi are reported in Table 2. As before, DimSim outperforms all other methods, with a gain of close to 2 SARI points over the closest competitor. It is also more efficient, requiring only $\frac{1}{4}$ the number of LLM calls, along with multilingual sub-critics instead of per-language sub-critics.

We observe that with cross-lingual + CoT prompting, the LLM tends to copy inputs more than 50% of the time. This is primarily because it fails to generate a comprehensive chain-of-thought string, and typically regresses to either producing an empty string or irrelevant text in Bengali or Marathi. Using translated examples without CoT slightly boosts SARI scores for both Bengali and Marathi. However, with CoT added, there is a 4 point drop in SARI. Recall that in the translation + CoT approach, Hindi phrases and suggested word-replacements in the analysis string are also translated to Bengali or Marathi. However, there is no guarantee that words/phrases present in the translated analysis string are actually valid sub-strings of the translated input string. As a consequence, the LLM has to learn from noisy in-context examples, resulting in a performance drop.

5.1 Human Evaluations

We perform both side-by-side, as well as single-sided human evaluations to measure the efficacy of our approach. Rater pool demographics and exact task instructions are reported in the Appendix.

Side-by-side: In this eval, human annotators are presented with two texts side by side - the original text, and its model-generated simplification. We ask two questions: 1. Which text is simpler? 2. How similar are they in meaning? Each text pair is evaluated by 3 language experts, and a consensus label is generated via majority vote. Instances without consensus agreement on both questions are removed from analysis.

Table 3 shows results from the side-by-side evaluations of DIMSIM on 200 randomly-sampled sentences from the test set of each language. For Hindi and Bengali, we find that in the majority of instances, raters find the output from DIMSIM to be simpler and also consistent in meaning to the original sentence. Marathi performs a bit worse in comparison, with raters frequently preferring the original input or finding both to be similar. However, DIMSIM outputs are rated simpler than the originals, though not in a majority of cases.

Table 4 compares CCSV (the second-best TS model in terms of SARI) vs. DIMSIM on 200 randomly-sampled Hindi sentences. While the CCSV approach trails slightly in producing simpler outputs, it lags significantly in producing outputs that preserve the meaning of the original text. This is corroborated by its BERTScore (in Table 1) which has regressed more than 6 points from the skyline.

	Which is simpler?			How similar are they in meaning?		
	original input	DIMSIM output	both are similar	very	somewhat	not at all
Hindi	4.1%	77.4%	18.5%	93%	7.0%	0%
Bengali	16.5%	60.5%	23.0%	82.5%	16.0%	1.5%
Marathi	32.5%	43.3%	24.2%	58.6%	37.3%	4.1%

Table 3: SxS human evals for Hindi, Bengali, and Marathi on DIMSIM.

	Which is simpler?			How similar are they in meaning?		
	original input	simplified output	both are similar	very	somewhat	not at all
CCSV	6.0%	76.1%	17.9%	57.6%	38.8%	3.6%
DIMSIM	4.1%	77.4%	18.5%	93%	7.0%	0%

Table 4: SxS human evals for Hindi comparing DIMSIM vs. CCSV.

Hindi		Bengali		Marathi	
original	simplified	original	simplified	original	simplified
45%	83%	53%	75%	64%	85%

Table 5: %age of simple instances in original vs. DIMSIM-simplified corpus, single-sided eval.

Single-sided: In this eval, we randomly sample 100 sentences and ask raters to annotate whether it is objectively simple to read and understand. Each sentence is presented to 2 raters, and is deemed to be simple only if both raters consider it simple. Table 5 shows results from a single-sided eval comparing the original and DIMSIM-simplified test corpus for all 3 languages. We see +38 point improvement in the number of simple sentences for Hindi, and more than +20 point improvement for Bengali and Marathi.

5.2 Impact of Trained Sub-Critics

In this section, we attempt to compare the magnitude of lexical and structure changes brought about by different approaches, with the help of a few easy-to-compute metrics. For both kinds of change, we report the trigger rate of few-shot and trained sub-critics (i.e. DIMSIM). A sub-critic is said to trigger for an input if it produces an actionable suggestion to simplify it. To quantify lexical change, we report the total number of word replacements that were suggested by the lexical critic and incorporated in the simplified version. To quantify structure change, we report two metrics: number of sentence-splitting instances, and number of paraphrasing instances. An instance has undergone splitting if the simplified version contains more sentences than the original. An instance has undergone paraphrasing if the Levenshtein distance between the original and simplified string is greater than 20.

As Table 6 shows, for both lexical and structure sub-critics, the trained variant has a significantly higher trigger rate compared to the few-shot variant. This higher trigger rate, taken together with superior SARI and BERTScores, suggests that the trained critics are making more accurate simplification changes to the input. We further validated this using a small human evaluation exercise. We randomly selected 200 instances where the few-shot sub-critics and DIMSIM produced different refinements of the input text. Human evaluations indicated DIMSIM led to simpler outputs for 32% of the instances, fewshot critics led to simpler outputs in 28% of instances, and raters found both sentences to be at-par for the remaining 40% of instances.

We expect a well-trained critic to trigger less on simpler datasets, and trigger more on a complex dataset. To check whether this holds, we evaluate the trained critic models on 3 different corpora comprising original complex inputs, DIMSIM outputs, and human references. Table 7 shows the trigger rates on all three corpora. As expected, trigger rates are highest for the original inputs, lowest for the human references and trigger rates for DIMSIM outputs are in-between.

These metrics can also help understand differences between DIMSIM and CCSV. Since CCSV does not have explicit sub-critics, it is not straightforward to quantify the magnitude of lexical change. However, it is possible to quantify structure change by comparing the original and simplified version. Recall that CCSV operates by collating multiple sampled critiques from the LLM and revising based on collated critique. Table 6 shows that CCSV is ≈2x more likely to trigger a structure change compared to DIMSIM. The large volume of change is because it incorporates diverse feedback from multiple critiques. However, as shown in Section 5.1, CCSV also changes the meaning of the input text. One possible hypothesis is that while trying to incorporate the volume of change brought on by multiple critiques, the LLM is not able to do it in a meaningful way.

	Lexical change		**Structure change**		
	trigger rate	#word replacements	trigger rate	#sentence splits	#paraphrases
Few-shot sub-critics	57.3%	1454	14.6%	118	64
DIMSIM	74.3%	1197	43.5%	383	183
CCSV	-	-	83.2%	645	771

Table 6: Trigger rates and operations suggested by few-shot and trained critics on the Hindi test corpus containing 1041 sentences.

		Test set inputs	Simplified outputs (DIMSIM)	Simplified outputs (human)
Lexical	Hindi	74.3	39.8	31.5
	Bengali	70.8	61.4	54.0
	Marathi	70.0	62.1	58.5
Structure	Hindi	43.5	31.6	28.3
	Bengali	44.4	30.8	18.0
	Marathi	26.3	20.7	11.3

Table 7: Table showing trigger rates of lexical and structure critics on 3 different corpus.

6 Conclusions

In this work, we present a new framework for multilingual text simplification using LLMs for self-correction. We decouple the simplification problem along two salient dimensions and train individual critics using synthetic data for each critic generated via few-shot prompting of LLMs. We evaluate our framework on text simplification for three Indian languages in a few-shot (for Hindi) and zero-shot (for Bengali and Marathi) settings. While we focus entirely on text simplification in this work, we think our framework is broad enough to be applicable to other generation tasks such as formality transfer, abstractive summarization, etc. and leave this exploration to other tasks as future work.

Acknowledgements

The fourth author would like to gratefully acknowledge a faculty grant from Google Research India supporting her research on multilingual models and is also very thankful for Google Cloud credits that helped support all the compute requirements. We are deeply indebted to the Vani and Verse Solutions team (especially Keshav and Deepika Agarwal) for helping us scale and expedite human evaluations on their platform.

7 Limitations

Our contribution in this work is limited to one particular family of languages, for a single task.

We recognize that the principle of using distilled trained critics is more general, and requires rigorous study on a multitude of tasks and languages. Second, methods that use LLMs to generate data may produce patterns of bias or factual inconsistencies, which are likely to propagate to downstream distilled models. Finally, we recognize that while our progress on Hindi and Bengali TS is significant, overall performance on Marathi is relatively low and requires further inspection.

References

Suha S Al-Thanyyan and Aqil M Azmi. 2021. Automated text simplification: a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. 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 49–54, Hong Kong, China. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1).

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023a. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Tannon Kew, Alison Chi, Laura Vásquez-Rodríguez, Sweta Agrawal, Dennis Aumiller, Fernando Alva-Manchego, and Matthew Shardlow. 2023. BLESS: Benchmarking large language models on sentence simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, pages 12–24.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Preethi Lahoti, Nicholas Blumm, Xiao Ma, Raghavendra Kotikalapudi, Sahitya Potluri, Qijun Tan, Hansa Srinivasan, Ben Packer, Ahmad Beirami, Alex Beutel, et al. 2023. Improving diversity of demographic representation in large language models via collective-critiques and self-voting. *arXiv preprint arXiv:2310.16523*.

Sharon Levy, Emily Allaway, Melanie Subbiah, Lydia Chilton, Desmond Patton, Kathleen McKeown, and William Yang Wang. 2022. Safetext: A benchmark for exploring physical safety in language models. *arXiv preprint arXiv:2210.10045*.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also" think" step-by-step. *arXiv preprint arXiv:2306.14050*.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2020. Mixkd: Towards efficient distillation of large-scale language models. *arXiv preprint arXiv:2011.00593*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2020. Zero-shot crosslingual sentence simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation.

Deepak Nathani, David Wang, Liangming Pan, and William Wang. 2023. MAF: Multi-aspect feedback for improving reasoning in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6591–6616, Singapore. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*.

Michael Ryan, Tarek Naous, and Wei Xu. 2023. Revisiting non-English text simplification: A unified multilingual benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model. *arXiv preprint arXiv:2208.11663*.

Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2022. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. *arXiv preprint arXiv:2212.08061*.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019a. Distilling task-specific knowledge from BERT into simple neural networks. *CoRR*, abs/1903.12136.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019b. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. "kelly is a warm person, joseph is a role model": Gender biases in LLM-generated reference letters. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Peifeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022. Pinto: Faithful language reasoning using prompt-generated rationales. *arXiv preprint arXiv:2211.01562*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022a. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. 2023a. On the paradox of learning to reason from data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3365–3373. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023b. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*.

A Human Evaluations

We hired a professional data labeling service for all human evaluations. This section provides exact annotation instructions, as well as relevant details about the rater pool.

A.1 Side-by-side Evaluation

In the side-by-side eval, raters were shown a pair of sentences (say, A and B) and asked to evaluate them on simplicity and semantic similarity. Each side-by-side pair was rated by 3 trained humans, followed by majority vote aggregation. When rating a dataset of sentence pairs, the order in which items in the pair are presented (i.e. (A, B) or (B, A)) may bias the raters' response over time. To mitigate this, items in the pair were randomly ordered so that raters' do not systematically prefer one side over the other.

A.1.1 Simplicity

For this question, raters were asked to select which one of the two texts is more simple, based on the following definition of simplicity:

A text is simple if it is easy to understand for an average native speaker. It has simple sentences, everyday vocabulary, and clear structure. It is not verbose or repetitive, and it does not use overtly academic or archaic language.

Raters could select one of 3 possible options: A is more simple, B is more simple, Both are similar.

A.1.2 Semantic Similarity

For this question, raters were asked to select how similar are the two texts in their meaning? They could select one of 3 possible options:

Very similar - A and B convey exactly the same information.

Somewhat similar - There is meaning overlap between A and B, but they are not identical. There are some differences in information or context.

Not at all similar - A and B are totally unrelated, there is absolutely no similarity between them.

A.2 Single-sided Evaluation

In this task, raters were shown a text and asked a binary question: *Is this text simple to read and understand?*. The definition of simplicity was identical to that of the side-by-side evaluation.

A.3 Rater Pool

Evaluations were performed by bilingual raters from India, who are fluent in English and one of {Hindi, Marathi, Bengali}. Every rater either held or was working towards a Bachelors degree. Raters were paid USD 0.12 for a completed side-by-side task, and USD 0.10 for a completed single-sided task.

B Sample Hindi Outputs

In Table 8 we show sample outputs from our best performing model, along with edit operations being performed. The examples show different operations like word replacements, sentence splitting, phrase rewrites, and paraphrasing been done by the model.

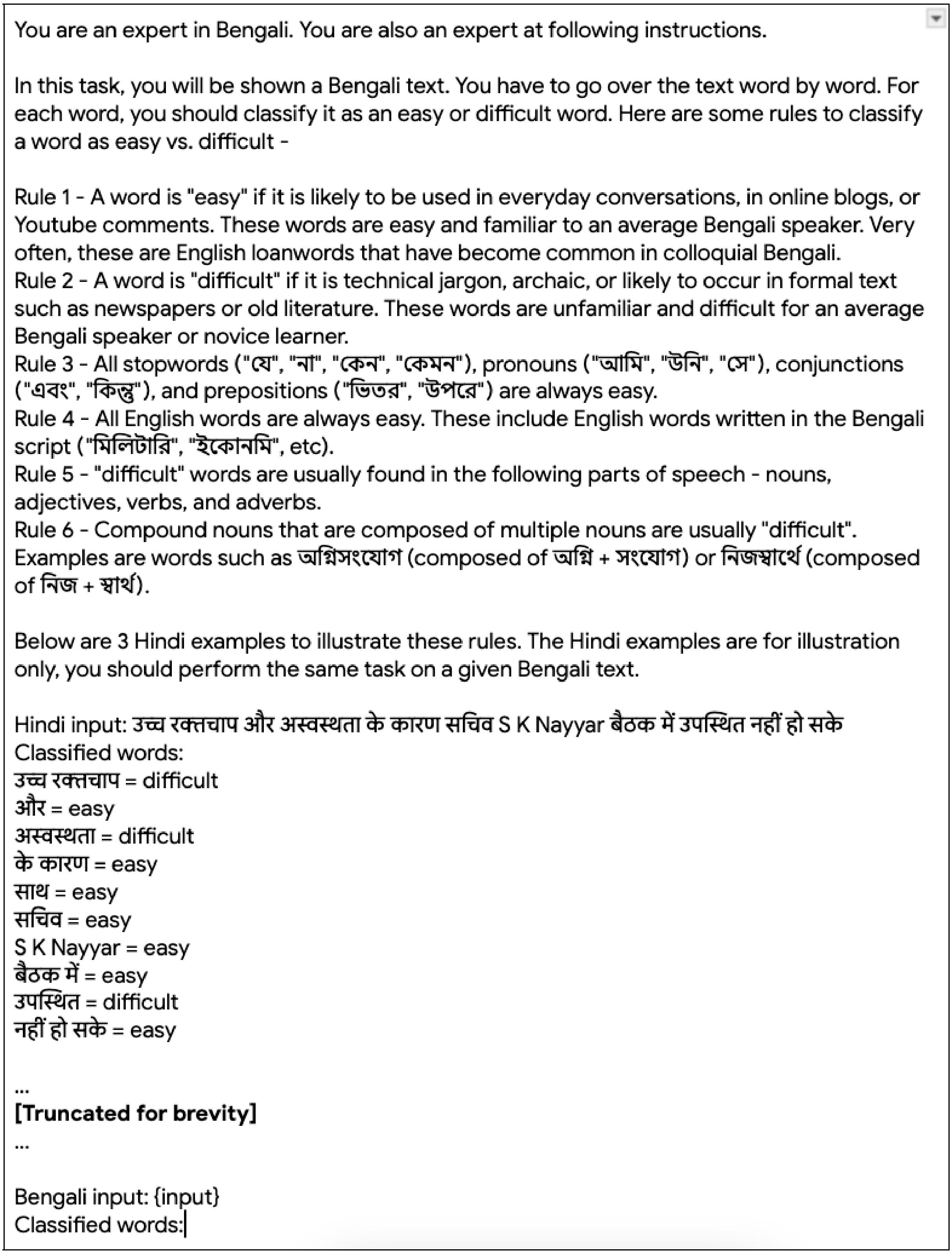
C Critic Prompts

The lexical critic module contains two sub-modules. The first is a *classification* module, the second is a *replacement* module. For brevity, we show the prompts for Bengali in Figure 2 and 3; prompts for Hindi and Marathi are similarly written. As shown in the prompt, the model is tasked to classify each word as easy or difficult in the classification prompt. We provide six set of rules to the model to help it classify the word as easy or difficult. In the subsequent task, the model is provided the replacement prompt to act upon the difficult words earlier identified. Again, we provide three steps in the prompt

Complex sentence	Simplified sentence	Operations
Sashastr sena adaalaton ko naagarik avehalana shakti bhee pradaan kee jae aur yah adaalaten kaanoon mantraalay ke adheen rakhee jaen	Army court ko civil contempt power bhee dee jae aur ye court law ministry ke adheen rakhee jaen.	Word replacements
Sandeep kumaar ke anusaar ghar se nakadee sahit keematee samaan jisakee kul keemat lagabhag dedh laakh banatee hai, kee choree huee hai.	Sandeep kumaar ke anusaar ghar se nakadee sahit keematee saamaan kee choree huee hai. Choree hue saamaan kee kul keemat lagabhag dedh laakh hai.	Sentence splitting
Yah vah anushaasan hai jo aakaash mein avalokit kee ja sakane vaalee tatha unaka samaavesh karane vaalee kriyaon ke aarambh, badalaav aur bhautik tatha raasaayanik gunon ka adhyayan karata hai.	Yah vah vishay hai jo aakaash mein dekhee ja sakane vaalee ghatanaon ka adhyayan karata hai. Yah unake shuruaat, badalaav, aur phizikal tatha kemikal gunon ka bhee adhyayan karata hai.	Paraphrase + Sentence splitting + phrase rewrites

Table 8: Examples of Hindi text simplification with DIMSIM. Original text is in the Devanagari script, it has been romanized for ease of reading.

to help the model to come up with simpler alternatives. The structure critic prompt is presented in Figure 4. Here, the model is prompted to critique the structure of the text, thereby highlighting a modified structure of the provided text that could make it more readable.



You are an expert in Bengali. You are also an expert at following instructions.

In this task, you will be shown a Bengali text. You have to go over the text word by word. For each word, you should classify it as an easy or difficult word. Here are some rules to classify a word as easy vs. difficult -

Rule 1 - A word is "easy" if it is likely to be used in everyday conversations, in online blogs, or Youtube comments. These words are easy and familiar to an average Bengali speaker. Very often, these are English loanwords that have become common in colloquial Bengali.
Rule 2 - A word is "difficult" if it is technical jargon, archaic, or likely to occur in formal text such as newspapers or old literature. These words are unfamiliar and difficult for an average Bengali speaker or novice learner.
Rule 3 - All stopwords ("যে", "না", "কেন", "কেমন"), pronouns ("আমি", "উনি", "সে"), conjunctions ("এবং", "কিন্তু"), and prepositions ("ভিতর", "উপরে") are always easy.
Rule 4 - All English words are always easy. These include English words written in the Bengali script ("মিলিটারি", "ইকোনমি", etc).
Rule 5 - "difficult" words are usually found in the following parts of speech - nouns, adjectives, verbs, and adverbs.
Rule 6 - Compound nouns that are composed of multiple nouns are usually "difficult". Examples are words such as অগ্নিসংযোগ (composed of অগ্নি + সংযোগ) or নিজস্বার্থে (composed of নিজ + স্বার্থ).

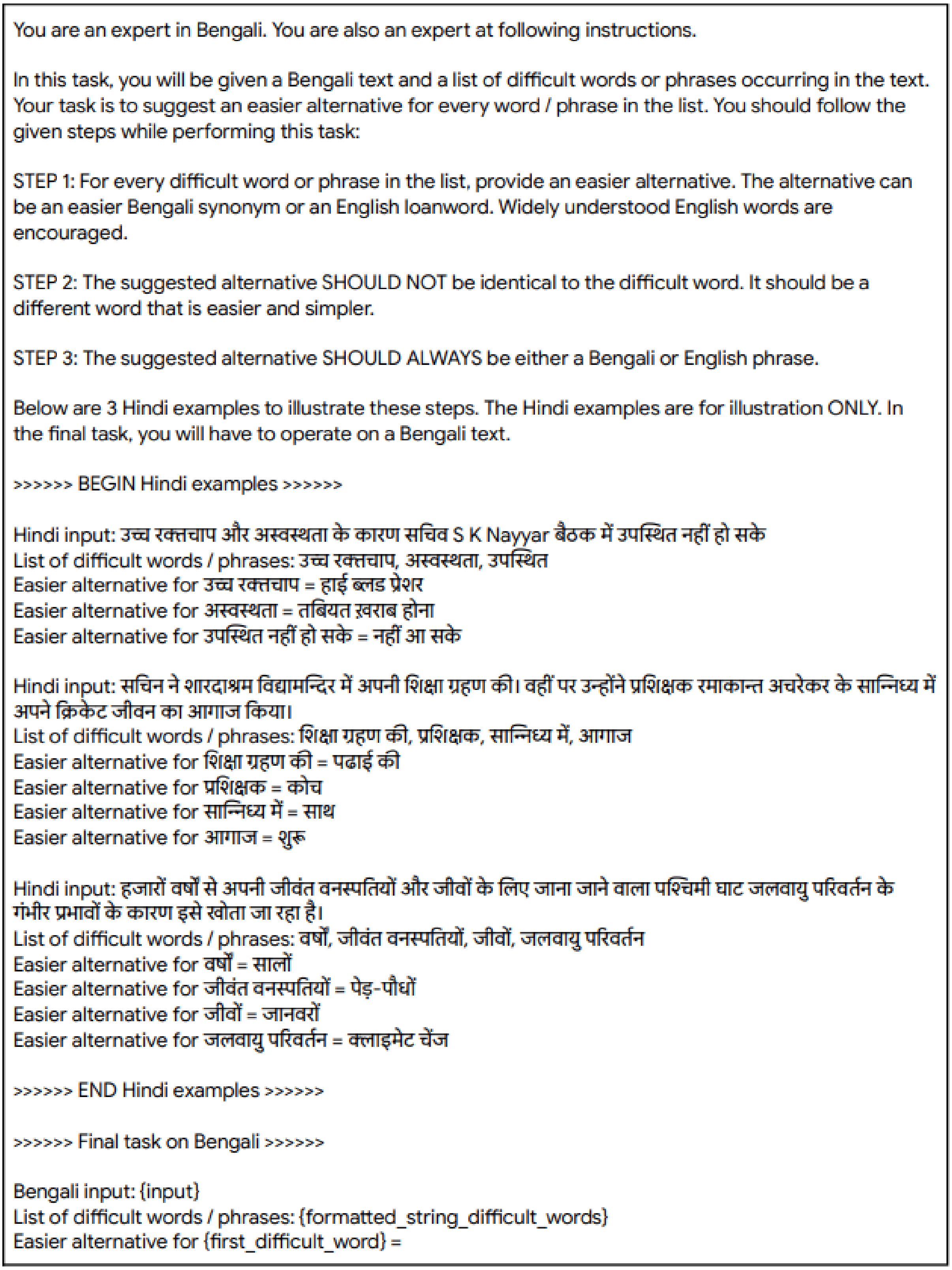
Below are 3 Hindi examples to illustrate these rules. The Hindi examples are for illustration only, you should perform the same task on a given Bengali text.

Hindi input: उच्च रक्तचाप और अस्वस्थता के कारण सचिव S K Nayyar बैठक में उपस्थित नहीं हो सके
Classified words:
उच्च रक्तचाप = difficult
और = easy
अस्वस्थता = difficult
के कारण = easy
साथ = easy
सचिव = easy
S K Nayyar = easy
बैठक में = easy
उपस्थित = difficult
नहीं हो सके = easy

...
[Truncated for brevity]
...

Bengali input: {input}
Classified words:

Figure 2: Lexical classification prompt



You are an expert in Bengali. You are also an expert at following instructions.

In this task, you will be given a Bengali text and a list of difficult words or phrases occurring in the text. Your task is to suggest an easier alternative for every word / phrase in the list. You should follow the given steps while performing this task:

STEP 1: For every difficult word or phrase in the list, provide an easier alternative. The alternative can be an easier Bengali synonym or an English loanword. Widely understood English words are encouraged.

STEP 2: The suggested alternative SHOULD NOT be identical to the difficult word. It should be a different word that is easier and simpler.

STEP 3: The suggested alternative SHOULD ALWAYS be either a Bengali or English phrase.

Below are 3 Hindi examples to illustrate these steps. The Hindi examples are for illustration ONLY. In the final task, you will have to operate on a Bengali text.

>>>>>> BEGIN Hindi examples >>>>>>

Hindi input: उच्च रक्तचाप और अस्वस्थता के कारण सचिव S K Nayyar बैठक में उपस्थित नहीं हो सके
List of difficult words / phrases: उच्च रक्तचाप, अस्वस्थता, उपस्थित
Easier alternative for उच्च रक्तचाप = हाई ब्लड प्रेशर
Easier alternative for अस्वस्थता = तबियत ख़राब होना
Easier alternative for उपस्थित नहीं हो सके = नहीं आ सके

Hindi input: सचिन ने शारदाश्रम विद्यामन्दिर में अपनी शिक्षा ग्रहण की। वहीं पर उन्होंने प्रशिक्षक रमाकान्त अचरेकर के सान्निध्य में अपने क्रिकेट जीवन का आगाज किया।
List of difficult words / phrases: शिक्षा ग्रहण की, प्रशिक्षक, सान्निध्य में, आगाज
Easier alternative for शिक्षा ग्रहण की = पढाई की
Easier alternative for प्रशिक्षक = कोच
Easier alternative for सान्निध्य में = साथ
Easier alternative for आगाज = शुरू

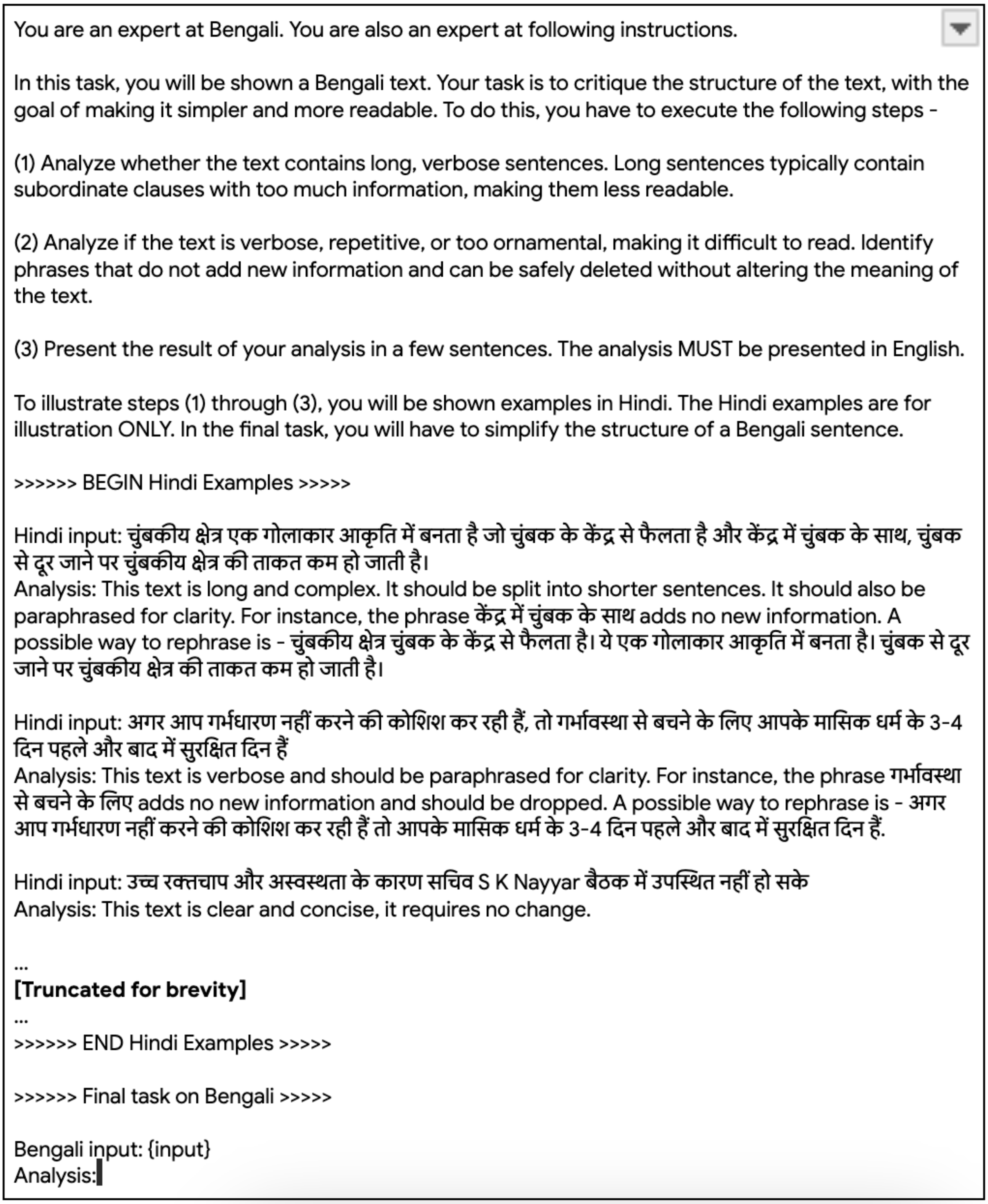
Hindi input: हजारों वर्षों से अपनी जीवंत वनस्पतियों और जीवों के लिए जाना जाने वाला पश्चिमी घाट जलवायु परिवर्तन के गंभीर प्रभावों के कारण इसे खोता जा रहा है।
List of difficult words / phrases: वर्षों, जीवंत वनस्पतियों, जीवों, जलवायु परिवर्तन
Easier alternative for वर्षों = सालों
Easier alternative for जीवंत वनस्पतियों = पेड़-पौधों
Easier alternative for जीवों = जानवरों
Easier alternative for जलवायु परिवर्तन = क्लाइमेट चेंज

>>>>>> END Hindi examples >>>>>>

>>>>>> Final task on Bengali >>>>>>

Bengali input: {input}
List of difficult words / phrases: {formatted_string_difficult_words}
Easier alternative for {first_difficult_word} =

Figure 3: Lexical replacement prompt



You are an expert at Bengali. You are also an expert at following instructions.

In this task, you will be shown a Bengali text. Your task is to critique the structure of the text, with the goal of making it simpler and more readable. To do this, you have to execute the following steps -

(1) Analyze whether the text contains long, verbose sentences. Long sentences typically contain subordinate clauses with too much information, making them less readable.

(2) Analyze if the text is verbose, repetitive, or too ornamental, making it difficult to read. Identify phrases that do not add new information and can be safely deleted without altering the meaning of the text.

(3) Present the result of your analysis in a few sentences. The analysis MUST be presented in English.

To illustrate steps (1) through (3), you will be shown examples in Hindi. The Hindi examples are for illustration ONLY. In the final task, you will have to simplify the structure of a Bengali sentence.

>>>>>> BEGIN Hindi Examples >>>>>

Hindi input: चुंबकीय क्षेत्र एक गोलाकार आकृति में बनता है जो चुंबक के केंद्र से फैलता है और केंद्र में चुंबक के साथ, चुंबक से दूर जाने पर चुंबकीय क्षेत्र की ताकत कम हो जाती है।
Analysis: This text is long and complex. It should be split into shorter sentences. It should also be paraphrased for clarity. For instance, the phrase केंद्र में चुंबक के साथ adds no new information. A possible way to rephrase is - चुंबकीय क्षेत्र चुंबक के केंद्र से फैलता है। ये एक गोलाकार आकृति में बनता है। चुंबक से दूर जाने पर चुंबकीय क्षेत्र की ताकत कम हो जाती है।

Hindi input: अगर आप गर्भधारण नहीं करने की कोशिश कर रही हैं, तो गर्भावस्था से बचने के लिए आपके मासिक धर्म के 3-4 दिन पहले और बाद में सुरक्षित दिन हैं
Analysis: This text is verbose and should be paraphrased for clarity. For instance, the phrase गर्भावस्था से बचने के लिए adds no new information and should be dropped. A possible way to rephrase is - अगर आप गर्भधारण नहीं करने की कोशिश कर रही हैं तो आपके मासिक धर्म के 3-4 दिन पहले और बाद में सुरक्षित दिन हैं.

Hindi input: उच्च रक्तचाप और अस्वस्थता के कारण सचिव S K Nayyar बैठक में उपस्थित नहीं हो सके
Analysis: This text is clear and concise, it requires no change.

...
[Truncated for brevity]
...
>>>>>> END Hindi Examples >>>>>

>>>>>> Final task on Bengali >>>>>

Bengali input: {input}
Analysis:

Figure 4: Structure critic prompt