Task: Rewrite your initial response to a user query according to the factuality corrections. Your rewritten response should keep internal consistency while making minimal edits.

You will be given:
1. Query: The user query.
2. Initial Response: The original text generated by an LLM to answer the user query.
3. Non-Factual Spans and Replacements: A list of text segments from the initial response that were potentially incorrect, along with their intended factual replacements.

Instructions:
1. Carefully review the query and the initial response.
2. Examine each non-factual span and its corresponding factual replacement.
3. Rewrite the initial response to incorporate the factual replacements while ensuring the response remains coherent and consistent.
4. Make minimal edits to the original text, altering what is necessary to correct factual inaccuracies as well as any incoherent or inconsistent content.
5. Output the rewritten response.

Test example:
1. Query: `[QUERY]`
2. Initial Response: `[INITIAL RESPONSE]`
3. Non-Factual Spans and Replacements:
Span 1:
- Span: `[SPAN #1]`
- Revision: `[REVISION #1]`
...
Span N:
- Span: `[SPAN #N]`
- Revision: `[REVISION #N]`
4. Your Output (Rewritten Response):

Table 11: Prompt template used to generate factuality rewrites. The `[QUERY]` and `[INITIAL RESPONSE]` placeholders will be replaced by a given user query and its initial response, and the `[SPAN #1 → N]` and `[REVISION #1 → N]` placeholders will be replaced by each of the N spans and revisions from the SAFE (Wei et al., 2024) outputs.

Task: Rewrite a source text according to the comment.

You will be given:
1. Source: The original text from a public corpus.
2. Comment: An open-ended rewrite requirement, such as formalize, paraphrase, shorten, elaborate, etc.

Instructions:
1. Carefully review the source text and the comment.
2. Understand the intent of the comment and how it should influence the rewrite.
3. Rewrite the source text to align with the comment's requirement, ensuring that the text remains internally consistent and coherent.
4. Ensure the rewritten text maintains the original meaning and context as much as possible.
5. Output the rewritten text.

Test example:
1. Source: `[SOURCE]`
2. Comment: `[REWRITE INSTRUCTION]`
3. Your Output (Rewritten Response):

Table 12: Prompt template used to generate stylistic rewrites. The `[SOURCE]` and `[REWRITE INSTRUCTION]` placeholders will be replaced by a given source text and rewrite instruction.

Task: Rewrite an email based on the given instruction. The rewritten email should maintain internal consistency and align with the provided instruction.

You will be given:
1. Natural Prompt: The original prompt used to generate the initial email.
2. Raw Generated Email: The initial email generated by the LLM based on the natural prompt.
3. Rewrite Instruction: The instruction for how to improve or modify the raw generated email.

Instructions:
1. Carefully read the 'Natural Prompt' and 'Raw Generated Email'.
2. Analyze the 'Rewrite Instruction' to understand the required modifications.
3. Rewrite the 'Raw Generated Email' according to the 'Rewrite Instruction'.

Test example:
1. Natural Prompt: `[NATURAL PROMPT]`
2. Raw Generated Email: `[EMAIL]`
3. Rewrite Instruction: `[REWRITE INSTRUCTION]`
4. Your Output (Rewritten Email):

Table 13: Prompt template used to generate conversation rewrites. The `[NATURAL PROMPT]`, `[EMAIL]`, and `[REWRITE INSTRUCTION]` placeholders will be replaced by a given natural prompt, its raw generated email, and rewrite instruction.

Task: Generate a rewrite instruction for a raw generated email based on the natural prompt. The instruction should guide the rewriting process to improve clarity, tone, structure, or other aspects.

You will be given:
1. Natural Prompt: The original prompt given to the LLM.
2. Raw Generated Email: The initial email generated by the LLM based on the natural prompt.
3. Demonstrations: Examples of existing emails and their rewrite instructions.

Instructions:
1. Read the 'Natural Prompt' and 'Raw Generated Email' carefully.
2. Analyze the 'Raw Generated Email' and identify areas for improvement based on the 'Natural Prompt'.
3. Refer to the provided 'Demonstrations' to understand different types of rewrite instructions and their contexts.
4. Generate a concise and clear rewrite instruction for the 'Raw Generated Email'.

Demonstrations:
Example 1:
- Email: `[EMAIL #1]`
- Instruction: `[REWRITE INSTRUCTION #1]`
...
Example N:
- Email: `[EMAIL #N]`
- Instruction: `[REWRITE INSTRUCTION #N]`

Test example:
1. Natural Prompt: `[NATURAL PROMPT]`
2. Raw Generated Email: `[EMAIL]`
3. Your Output (Rewrite Instruction):

Table 14: Prompt template used to generate raw conversation rewrite instructions. The `[NATURAL PROMPT]` and `[EMAIL]` placeholders will be replaced by a natural prompt and its generated raw email, and the `[EMAIL #1` $\rightarrow$ `N]` and `[REWRITE INSTRUCTION #1` $\rightarrow$ `N]` placeholders will be replaced by each of the N (email, instruction) pairs.

Explicit input segmentation: The prompt clearly separates the natural prompt, the raw generated email, and the rewrite instruction, ensuring that the model understands the original context before making modifications. (ii) Focus on instruction adherence: Unlike factuality and stylistic rewrites, which prioritize correctness and rewording, this task emphasizes cohesively integrating the instruction into the existing email while maintaining internal consistency. (iii) Preserving conversational flow: The instructions explicitly require the model to analyze the provided email and apply the requested changes while ensuring the rewritten email remains natural and engaging.

By structuring the prompt in a way that guides but does not over-restrict the model, this template ensures that email rewrites maintain fluency, correctness, and engagement while following specific improvement instructions. The step-by-step breakdown aids the model in handling more challenging contextual refinements, such as enhancing tone, incorporating enthusiasm, or making the message more concise.

Conversational rewrite instruction generation. Table 14 outlines the prompt template for generating rewrite instructions for conversational emails. This step is crucial in structuring the dataset, as it determines the type and specificity of modifications the model will learn to perform. Unlike factuality or stylistic rewrites, conversational rewrites often involve subtle refinements in tone, structure, and content balance.

Table 15 illustrates the diversity of generated instructions, ranging from simple grammatical refinements (e.g., "use complete sentences") to substantial structural modifications (e.g., "add structure and boilerplate to make the email more professional"). The ability to generate rich, context-aware instructions ensures that conversational rewrites cover various real-world use cases, enhancing the model's generalization ability.

Conversational rewrite instruction refinement. Table 16 presents the prompt template for refining generic rewrite instructions by making them more specific and actionable. This step is crucial in ensuring that rewrite instructions provide clear, detailed guidance, reducing ambiguity for the rewriting model. As seen in Table 17, refining instructions significantly improves task clarity and execution. Instead of broad instructions (e.g., "Make it more persuasive"), the specified versions provide concrete actionable changes (e.g., "Highlight increased brand visibility, direct customer engagement, and networking opportunities"). This ensures that the rewriting model receives precise, contextually relevant directives, ultimately leading to higher-quality, more controlled rewrites.

Table 18 presents the prompt template for making rewrite instructions more natural and linguistically diverse. This step is essential in ensuring that rewrite instructions feel human-like, conversational, and engaging, while still preserving clarity and specificity. Table 19 showcases how structured rewrite instructions are transformed into more intuitive, engaging requests. Instead of formal, mechanical directives (e.g., "Make the email more specific by mentioning the product name and key features."), the modified instructions use more natural phrasing (e.g., "This email is too generic—specify the product name and key features, and highlight why these updates matter for media professionals.").

D AutoRater Prompting

The LLM-as-a-judge framework systematically evaluates rewrites across agreement, coherence, and pairwise comparisons (AutoSxS) to ensure fine-grained, objective assessments. Below, we describes its design rationales.

Agreement evaluation across tasks. Agreement evaluation measures how well the rewritten response adheres to the given instruction across different tasks:

- **Factuality rewrite**: Table 20 illustrates a span-based approach, checking whether all identified factual inaccuracies in the initial response have been corrected. Each non-factual span is individually assessed, ensuring precise, granular feedback.
- **Stylistic rewrite**: Table 21 deconstructs the instruction into multiple transformation requirements (e.g., formalization + conciseness) and evaluate whether each aspect is incorporated while preserving meaning.
- **Conversational rewrite**: Table 22 introduces a context-aware evaluation, ensuring that modifications align with the natural prompt while accurately implementing tone, structure, or engagement-related changes.

Coherence evaluation. Coherence evaluation in AutoRater is designed to assess whether a rewritten response maintains internal logical consistency while ensuring fluency and readability. Unlike