A Generating paraphrased sentences

Below, we include the prompt used to generate the paraphrased instances:

```
Your task is to restructure the given
    sentence so that the word to fill in
     (_, the blank) is now the last word
     in your restructured sentence that
    was also a word in the original
    sentence.
This means, you can add words after the
    blank, but only if they were not in
    the original sentence. This is so
    that the sentence doesn't
    inadvertently imply a specific
    answer (e.g., by making one option
    grammatically or contextually more
    likely than the other).
Do NOT fill in the _.
In no case should you change anything
    about the meaning the sentence is
    conveying, that is, do not add new
    content to the story that was not in
     the spirit of the original story.
You can only have one blank in the
    sentence.
When easily possible, make the sentence
    sound fluent, while abide by the
    rules above.

Example 1:
Input: I wanted to build a bathroom on
    the third floor of the house but I
    couldn't because the _ would be too
    full.
Possible tokens to fill in (just for
    reference): bathroom, floor
Output: I wanted to build a bathroom on
    the third floor of the house but
    because it would be too full, that _
    , I couldn't.
(Notice that this one violates the rules
     of the last word, but "I couldn't"
    is a vital part of the story that
    determines whether _ should be "
    bathroom" or "floor".)

Example 2:
Input: Jill was on a budget so she only
    bought a new dress for the ceremony
    and wore an old hat. She figured the
     _ would be less noticeable.
Possible tokens to fill in: dress, hat
Output: Jill was on a budget so she only
     bought a new dress for the ceremony
     and wore an old hat, figuring that
    the more noticeable item would be
    the _.

Example 3:
Input: To make frosting I needed pudding
     that was at a store 15 minutes away
     but pre-made frosting was at a
    store 5 minutes away.  The _ was
    closer.
Possible tokens to fill in: pudding,
    frosting
Output: To make frosting I needed
    pudding that was at a store 15
    minutes away but pre-made frosting
    was at a store 5 minutes away, so
    the closer choice was the _.

Example 4:
Input: The home that my parents had when
     I was in school was a lot nicer
    than my house now because the _ was
    sophisticated.
Possible tokens to fill in: home, house
Output: The home that my parents had
    when I was in school was a lot nicer
     than my house now because of how
    sophisticated the _ was.

Your task:
Input: [sentence]
Some possibilities that can replace the
    token are "[option 1]", or "[option
    2]", but either should be able to
    fill the blank (grammatically
    speaking).

Reason about how to make this happen,
    then after thinking, only give the
    restructured sentence.
Draw inspiration from all of the
    examples above. What worked
    previously are eg. cleft sentences,
    passive voice, relative clauses,
    appositives, inversions,
    prepositional phrases, etc.
```

The distribution of which model's results are used can be found in Table 5.

Model	Freq
GPT-4o	336
OpenAI o1-preview	76
Gemini 2.0 Flash Thinking Experimental	105
Deepseek R1	109
LlaMA 3.2 90B Vision	95
Manual	433
Original (unchanged)	31

Table 5: Number of times each model's output was chosen in the paraphrasing process of the 1,185 retained instances. The bottom two lines contain the number of times a manual adjustment was necessary, and the number of times the original sentence was already in the required paraphrased format.

B Prompt used to categorize sentences

The first prompt is used to generate reasoning steps to solve the task. As in-context examples, we use instances from the Winograd Schema Challenge. For this step, we use the OpenAI API to prompt gpt-4o-mini-2024-07-18.

```
You are a helpful assistant. Read the
    instructions carefully.

**INSTRUCTIONS**
Read the Input Text. The Input Text is a
     text from the WinoGrande benchmark.
     You get the text, and the two
    possible options to fill in the _ in
     the text.
Think long and hard, and identify the
    reasoning steps you need to make to
    decide which option is the correct
    answer of the Input Text of the TASK
    .
Provide the reasoning steps concisely.
    Then, return the correct option.

*IMPORTANT:*
- Your response **must** be in JSON
    format with the following structure:
  {
      "reasoning": "Your detailed
          reasoning here.",
      "output": "the correct option to
          fill in the blank, chosen
          between Option1 and Option2"
  }
- Do NOT include any additional text
    outside the JSON object.
- Ensure that the JSON kes are exactly "
    reasoning" and "output".
- Make sure your reasoning and output
    relate to the Input Text of the TASK
    .


**EXAMPLES**
Example Text 1: "The trophy doesn't fit
    into the brown suitcase because _ is
     too large. Option 1: The trophy.
    Option 2: the suitcase."
Example Reasoning 1 : "The object has to
     be smaller than the container in
    order to fit inside of it. If the
    trophy is too large, it does not fit
     in the suitcase."
Example Output 1 : "The trophy"

Example Text 2: "Joan made sure to thank
     Susan for all the help _ had
    recieved. Option 1: Joan. Option 2:
    Susan."
Example Reasoning 2 : "In social
    settings, you thank the person that
    gave you help. Since Joan received
    the help from Susan, Joan thanked
    Susan for the help that Joan
    received."
Example Output 2 : "Joan"

Example Text 3: "The large ball crashed
    right through the table because _
    was made of steel. Option 1: The
    large ball. Option 2: the table."
Example Reasoning 3 : "We know the ball
    is large. A large ball made of steel
    , which is heavy, is more likable to
     crash through a table."
Example Output 3 : "The large ball"

**TASK**
Input Text: "INPUT_TEXT. Option 1:
    OPTION1. Option 2: OPTION2."
```

The second prompt is given the input text, and the generated reasoning steps from the previous step, to label the instances of one of the five common sense categories. For this step, we use the OpenAI API to prompt gpt-4o-2024-08-06.

```
You are a helpful assistant. Read the
    instructions carefully.

** INSTRUCTIONS**
Your task is to decide which common
    sense knowledge categories are
    present in a text. In the Input Text
    , you get an example from the
    WinoGrande benchmark, and the two
    possible options to fill in the _ in
     the text. Then, you get the
    reasoning steps that specify the
    thought processes.
Read the Input Text and Reasoning Steps
    carefully, and select one or more
    common sense knowledge categories in
     which the Reasoning Steps fit. In
    other words, which knowledge types
    are used in the Reasoning Steps?
You can only use categories that are
    part of the list below. Return the
    index of the relevant category,
    following the example below. If
    multiple categories apply, list all
    relevant indices separated by commas
    .
Output only the indices without any
    additional text or explanations.

**COMMON SENSE CATEGORIES**
1. Physical: Pertains to physical
    attributes and properties of objects
     that are relevant to solve the task
    .
    Examples:
            "The apple is red."
            "The bottle is empty."

2. Social: Involves social norms, roles,
     and interactions you need to
    understand to solve the task.
    Examples:
            "She greeted her neighbor."
            "They followed the protocol
                ."

3. Numerical: Relates to numbers and
    quantities; differences in number or
     quantity between entities.
    Examples:
            "There are many books on the
                 shelf."
            "He ran 10 miles."

4. Temporal: Concerns time, temporal
    relations, and eventualities related
     to important entities of the task (
    important: NOT about temperature).
    Examples:
```

```
            "She arrived before noon."
            "They will meet tomorrow."

5. Spatial: Involves spatial relations (
    e.g., higher - lower), locations (e.
    g., north - south), or positions (e.
    g., behind - in front) that are
    important to understand to solve the
     task.
    Examples:
            "The cat is under the table
                ."
            "He walked into the room."


**EXAMPLES**
    Example Reasoning 1 : "The object
        has to be smaller than the
        container in order to fit inside
         of it. If the trophy is too
        large, it does not fit in the
        suitcase."
    Relevant Categories 1 : Numerical
        (3), Spatial (5)
    Output 1: 3, 5

    Example Reasoning 2 : "In social
        settings, you thank the person
        that gave you help. Since Joan
        received the help from Susan,
        Joan thanked Susan for the help
        that Joan received."
    Relevant Categories 2: Social (2),
        Temporal (4)
    Output 2: 2, 4

    Example Reasoning 3 : "We know the
        ball is large. A large ball made
         of steel, which is heavy, is
        more likable to crash through a
        table."
    Relevant Categories 3: Physical (1),
         Numerical (3)
    Output: 1, 3

Input Text: "INPUT_TEXT. Option 1:
    OPTION1. Option 2: OPTION2.
    Reasoning: REASONING"
```

C Detailed statistics on memorization checking

C.1 Counting the number of contaminated instances

From each of the 1,267 WinoGrande validation instances, we extract the longest n-gram that appears at least once in the corpus (i.e. The Pile or RedPajama v1) using the infini-gram API (Liu et al., 2024). For each instance, we find all occurrences of this n-gram in documents and extract 100-grams centered on it, ignoring instances with over 100 occurrences. We then prompt OpenAI's o1 to verify if the full sentence appears in any of these extracted 100-grams. The prompt can be found in Appendix D. This method allows us to handle inserted characters like LaTeX line breaks, functioning similarly to k-skip n-grams, though infini-gram doesn't offer the latter capability. This also ensures that contamination can be found even if there are subtle differences in text segments, a criticism that does apply to more naive n-gram overlap (Xu et al., 2024)

The output is positive if the entire instance of the validation set is found in the pre-training dataset.

C.2 Categorization of contaminated instances

In Table 6, the category distribution of the contaminated instances of RedPajama v1 is shown. The category distribution of the WinoGrande validation set can be found in Figure 3. A two-sample Kolmogorov-Smirnov test on the distributions rejects the zero hypothesis that the distribution of the leaked instances and the true category distribution are drawn from the same underlying distribution ($p = 0.79\%$). Hence, the contaminated instances LLaMA-1 encountered during pre-training are a skewed representation of the true distribution of the WinoGrande validation set.

C.3 Statistical analysis of contaminated instances

We analyze whether contamination in the WinoGrande validation set affects model performance using two statistical approaches: (1) comparing logprob differences between truly correct and incorrect answers, and (2) examining binary classification rates. The classification rate represents the proportion of correct predictions made by the model: a value of 1 means the model correctly identified the answer, while 0 indicates an incorrect prediction. For both approaches, we test contaminated instances from RedPajama v1 against non-contaminated instances across Llama-1 models using one-sided tests (Mann-Whitney U for logprobs and Fisher's exact for classification rates). We formulate the following hypotheses:

- H_0: There is no difference in performance (logprob differences/classification accuracy) between contaminated and non-contaminated instances.
- H_a: Performance is greater for contaminated instances than for non-contaminated instances.

The p-values for both tests can be found in Table 7. Neither test showed statistical significance, and thus the null hypotheses cannot be rejected.

Category	RedPajama v1
Social	11
Physical	9
Spatial	2
Numerical	4
Temporal	1

Table 6: The distribution of the contaminated instances in RedPajama v1 according to their categories.

Model	Mann-Whitney U p-value	Fisher's exact p-value
Llama 7B	0.0539	0.054
Llama 13B	0.2665	0.267
Llama 30B	0.0945	0.095
Llama 65B	0.2573	0.257

Table 7: Statistical test p-values checking the effect of contamination in the WinoGrande validation set (RedPajama v1) on Llama 1 models' performance, measured by logprob differences (Mann-Whitney U) and binary classification rates (Fisher's exact).

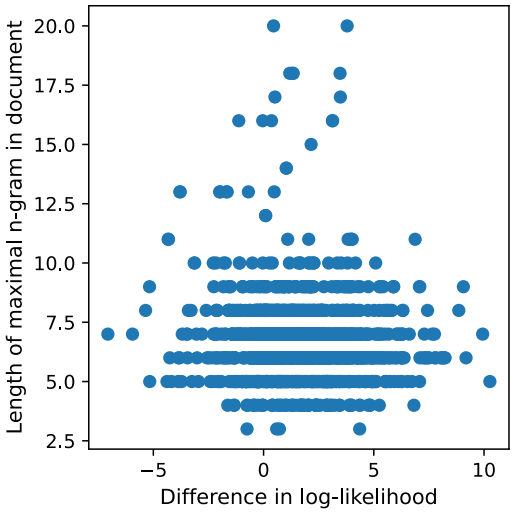
C.4 Correlation between n-gram length/count and performance on contaminated instances

We conducted two visual correlation analyses for all instances in the WinoGrande validation set against the logit difference between ground truth correct and incorrect answers. The first analysis examined the correlation with the length of the longest n-gram sequence that appears at least once in the pre-training data, as detailed in Appendix C.1. The second analysis focused on n-gram frequency, measuring how often the longest n-gram occurs in the pre-training data. Figure 1 displays scatter plots for both analyses, and neither reveals any visible correlation.

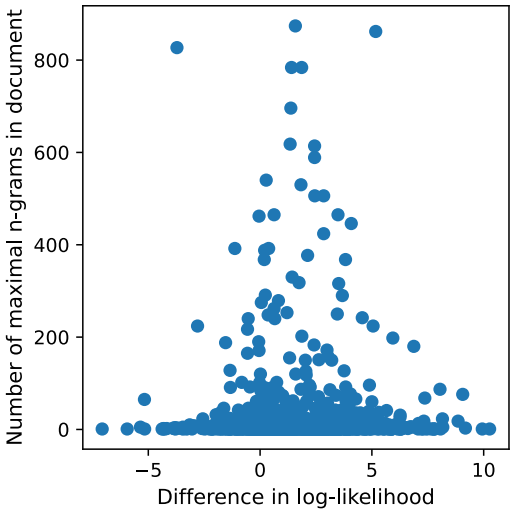
D Prompt used to verify n-grams in a pre-training dataset

```
Given is a sentence. Below that is an n-
    gram that occurs in the sentence.
    Below that are some numbered
    documents, with the number between
    parentheses, that all contain the n-
    gram.

Does at least one of the documents also
    contain the entire sentence,
    regardless of the occurrence of the
    the n-gram? If no, respond only "no
    ". If yes, give one document number
    and respond only "yes: <document
    number>".
Do not respond anything else.

Sentence: "[sentence]"
N-gram: "[n-gram]"
Documents:
```

````
```
(1) [document 1 excerpt]
(2) [document 2 excerpt]
...
```
````



(a) Correlation with n-gram length



(b) Correlation with n-gram count

Figure 4: Scatter plots showing correlation for all instances in the WinoGrande validation set against the logit difference between ground truth correct and incorrect answers. (a) Correlation with n-gram length. (b) Correlation with n-gram count.