Towards Human Understanding of Paraphrase Types in Large Language Models

Dominik Meier[1,2,3], Jan Philip Wahle[1], Terry Ruas[1], Bela Gipp[1]
[1]University of Göttingen, Germany
[2]LKA NRW, Germany
[3]meier@gipplab.org

Abstract

Paraphrases represent a human's intuitive ability to understand expressions presented in various different ways. Current paraphrase evaluations of language models primarily use binary approaches, offering limited interpretability of specific text changes. Atomic paraphrase types (APT) decompose paraphrases into different linguistic changes and offer a granular view of the flexibility in linguistic expression (e.g., a shift in syntax or vocabulary used). In this study, we assess the human preferences towards ChatGPT in generating English paraphrases with ten APTs and five prompting techniques. We introduce APTY (Atomic Paraphrase TYpes), a dataset of 800 sentence-level and word-level annotations by 15 annotators. The dataset also provides a human preference ranking of paraphrases with different types that can be used to fine-tune models with RLHF and DPO methods. Our results reveal that ChatGPT and a DPO-trained LLama 7B model can generate simple APTs, such as additions and deletions, but struggle with complex structures (e.g., subordination changes). This study contributes to understanding which aspects of paraphrasing language models have already succeeded at understanding and what remains elusive. In addition, we show how our curated datasets can be used to develop language models with specific linguistic capabilities.

1 Introduction

Paraphrases are changes in a text's wording or structure, resulting in a new text with approximately the same meaning (Bhagat and Hovy, 2013; Vila et al., 2014; Wahle et al., 2023). Paraphrase plays a fundamental role in NLP as understanding the variability in linguistic expression is key for various tasks, e.g., prompt engineering, text summarization, and plagiarism detection (Zhou et al., 2022b; El-Kassas et al., 2021; Barrón-Cedeño et al., 2013). Many have assessed whether two texts convey the same meaning through a single similarity score or binary assessment, limiting the granularity of predictions.

Atomic Paraphrase Types (APT) (Barrón-Cedeño et al., 2013; Vila et al., 2014) can be used as a new lens through which the linguistic relationship between two paraphrases can be explained. Generating and detecting APTs over binary categorizing paraphrases has multiple advantages (Wahle et al., 2023). For example, APTs can pinpoint whether a sentence's grammatical structure or the used vocabulary has changed between potential plagiarism cases (Alvi et al., 2021). Understanding how language models understand this variation in linguistic expression gives us insights into how their understanding of language differs from that of humans. It also explains in which language aspects models are proficient, where challenges remain, and how we can make models more robust to a wide array of paraphrase characteristics (e.g., syntactical and lexical changes). There are many ways in which two paraphrases can differ. Consider the following example:

> **Original**: "$\underline{\text{They}}_{\text{(a)}}$ had published an advertisement on the Internet $\underline{\text{on June 10}}_{\text{(b)}}$, offering the $\underline{\text{cargo}}_{\text{(c)}}$ for sale, $\underline{\text{he added}}_{\text{(d)}}$."
> **Paraphrase**: "$\underline{\text{On June 10}}_{\text{(b)}}$, $\underline{\text{the ship's owners}}_{\text{(a)}}$ had published an advertisement on the Internet, offering the $\underline{\text{explosives}}_{\text{(c)}}$ for sale."

Here, the paraphrase contains the following APT changes: (a) and (c) change the lexical unit for another one with the same meaning, (b) re-orders the words in the sentence, and (d) adds lexical and functional units.

So far, how well models generate or detect paraphrases with specific APTs has been largely unknown. In this work, we asked 15 humans to annotate 800 APT generations with various properties such as the perceived difficulty of generation, the model's success at generating a certain type and the reasons behind their failure, its confusion with

Proceedings of the 31st International Conference on Computational Linguistics, pages 6298–6316
January 19–24, 2025. ©2025 Association for Computational Linguistics

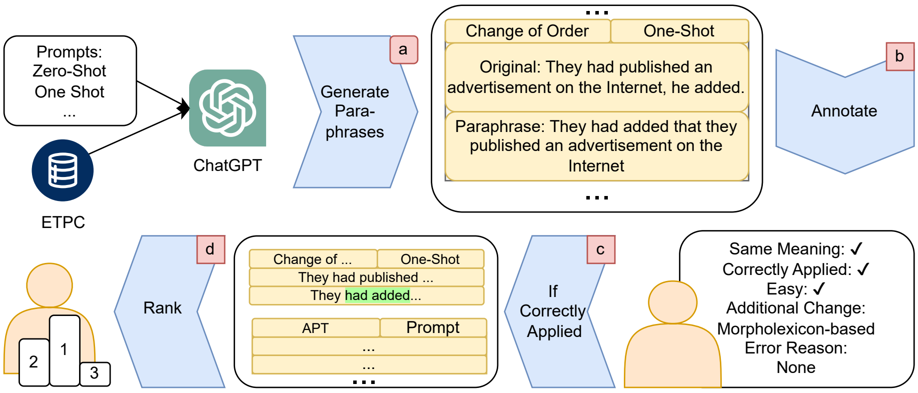


Figure 1: The generation and annotation process. During paraphrase generation (a), we select samples from the ETPC dataset (Kovatchev et al., 2018) and prompt ChatGPT-3.5-turbo-0613 (OpenAI, 2023) using zero-shot, one-shot, few-shot, chain-of-thought, and a fine-tuned model to generate new examples considering selected paraphrase types. For each technique and paraphrase type combination, we sample ten sentences. With five prompting techniques and ten selected APTs, we produce 500 sentence pairs. In (b), paraphrased candidates are annotated by 15 humans, who answer questions and highlight the word spans of the change. In (c), we select the generations in which the APT has been applied correctly, and in (d), the selected generations are ranked from worst to best.

other types, and the similarity of the APT on a sentence and word level. We publish this dataset as **A**tomic **P**araphrase **TY**pes Base (APTY$_{\text{Base}}$)[1]. To further contribute to the future of research on APTs, we extended this new dataset by ranking the different APT examples by human preferences that can be used to optimize paraphrase generation models in the future using human preference methods such as RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023) called APTY$_{\text{Ranked}}$. We demonstrate the usefulness of the preference data by using DPO to train a Llama 2 7B (Touvron et al., 2023) based model and compare the generation performance to a supervised fine-tuned and base version. The whole generation and annotation process is shown in Figure 1 and discussed in Section 3.1.

Our results show that ChatGPT-3.5-turbo-0613 (ChatGPT) is capable of generating *Same Polarity Substitution*, *Semantic-Based Changes*, and *Change of Order* and struggle with generating *Inflectional Changes*, *Synthetic/Analytic Substitutions*, and *Subordination and Nesting Changes*. In general, changes requiring deeper grammatical understanding are difficult to generate. Few-shot and chain-of-thought (CoT) prompting have increased generation success compared to other prompting techniques, especially for *Addition/Deletion* and *Semantic-Based Changes*. Surprisingly, humans often rank CoT generations lower than other methods. We also found that the most common error for ChatGPT when applying APT is applying the wrong kind of APT and that morpholexical changes, i.e., changes that arise at the word or morpheme level, are the most commonly wrongly or additionally applied. Lastly, the DPO-trained LLama outperforms the supervised fine-tuned and base Llama models by a wide margin. Our main contributions are:

1. A human study with 500 annotations of 15 participants on the ability of ChatGPT to generate paraphrase types (Q1);
2. A new dataset (APTY$_{\text{Base}}$) with sentence-pair information on sense-preservation, specific paraphrase type applied, the location of the change, and error reasons over five methods of paraphrase generation (Q2), with different prompt styles of zero-shot, one-shot, few-shot, chain-of-thought, and a fine-tuned model;
3. Analysis of human preferences (Q3) of paraphrase generation and a new dataset APTY$_{\text{Ranked}}$ with human-ranked paraphrase type generations using best-worst scale
4. Investigation of types of errors (Q4) ChatGPT makes when generating APTs and how much ChatGPT confuses APTs (Q5), i.e., confusion between the types;
5. Evaluation of DPO training (Q6) with the obtained human preferences based on the LLama 2 7B model family with 300 annotations

[1] https://github.com/worta/apty

2 Related Work

Approaches for paraphrase generation range from rule- and template-based approaches (Androutsopoulos and Malakasiotis, 2009) to trained transformers generating paraphrased text (Wahle et al., 2022). Rule-based methods rely on parsing the original sentence and applying either hand-crafted (McKeown, 1983) or automatically inferred (Lin and Pantel, 2001) rules to transform the text. Recently, paraphrase generation involves deep learning models, especially LLMs (Zhou and Bhat, 2021). Witteveen and Andrews (2019) use a fine-tuned version of GPT-2 to generate paraphrases and evaluate their semantic similarity. Palivela (2021) fine-tune a T5 model to generate and identify paraphrases. Wahle et al. (2022) explore T5 and GPT-3 regarding qualitative properties of generated paraphrases, access the ability of humans to identify machine-paraphrased text, and suggest that LLMs can generate paraphrases that match human-generated paraphrases in clarity, fluency, and coherence.

As previous paraphrase tasks rely heavily on similarity scores and do not capture the linguistic flexibility of paraphrases, Wahle et al. (2023) proposed two new tasks, i.e., paraphrase type generation and detection, using the ETPC (Kovatchev et al., 2018) dataset. Their findings indicate that current LLMs (e.g., ChatGPT) perform well when generating paraphrases with generic semantic similarity but struggle to generate them with fixed APTs. Additionally, models trained with APTs have improved performance in general paraphrase tasks (i.e., without APTs). While little is known yet in full detail on the paraphrastic mechanisms of all paraphrase types, further work revealed that specific types elicit prompt engineering capabilities over various downstream tasks, e.g., polarity for sentiment, or discourse for summarization (Wahle et al., 2024). An alternative approach to APT to have more control over linguistic properties is generating paraphrases based on syntactic templates (Huang and Chang, 2021). However, changes on a template level are too specific in a generation or detection task for humans to specify desired concepts.

Although Wahle et al. (2023) has sparked interest in more granular paraphrasing, their work relies only on automatic metrics such as BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004), which is limited and lacks the human component in the evaluation process. We explore human preferences in paraphrases by examining whether a paraphrase type was correctly applied by a model, the kind of errors made by ChatGPT at the time of generation, and by ranking paraphrases according to human judgement.

3 Methodology

Our experiments are split into two parts, i.e., generating paraphrase types using ChatGPT and annotating and evaluating their outputs according to multiple criteria with human participants. The process is shown in Figure 1. In the following subsections, we detail the generation and annotation processes.

3.1 Paraphrase Type Generation

We consider a variant of the paraphrase type generation task described in (Wahle et al., 2023). Given an APT $l \in L$, where L is the set of possible paraphrase types, and x a base sentence, we want to generate a paraphrase $\tilde{x}$ incorporating the given change l while maximizing the similarity between x and $\tilde{x}$. We limit ourselves to applying a single paraphrase type for the highest degree of control; we leave research into the complexity of combining different APTs for future work. We do not restrict the position where the change has to be applied; the model must choose where to apply the change.

In our experiments, we use the ETPC dataset (Kovatchev et al., 2018), which contains pairs of original and paraphrased sentences annotated with APTs. We choose the ten APTs with the most examples in the dataset as primary for this study. We excluded *Identity* changes, as those contain no change in the sentence; *Same Polarity Substitution (habitual and named entity)* as it is similar to *Same Polarity Substitution (contextual)*, which could harm the diversity for the chosen paraphrase types. We also exclude *Syntax/Discourse Structure* changes as it would require our annotators to understand all other paraphrase types, even those not considered in the study. Appendix A.1 details a full list of types.

We sample ten paraphrase pairs for each paraphrase type l. We use the first sentence in a paraphrase pair as a base sentence and generate the paraphrase using ChatGPT-3.5-turbo-0613. The prompts are constructed by asking the model to generate a sentence with the same meaning using the APT l. We prompt the model using zero-shot, one-shot, few-shot (Bragg et al., 2021), chain-of-thought (CoT) (Wei et al., 2022), and a fine-tuned model from Wahle et al. (2023), which is based on ChatGPT-3.5-turbo-0613 as well. The prompt contains the name of type l and its definition, taken from (Barrón-Cedeño et al., 2013) with minimal changes to fit the prompt (see Appendix C for exact prompts used.) Similarly to Bragg et al. (2021), five few-shot examples are given; one instance with added reasoning is provided for CoT prompts. Appendix D details the exact APT definitions. We use the same base sentences for evaluating the DPO-tuned LLama model and follow the prompt template for the fine-tuned model. We use Llama 2 7B, a version of LLama 2 7B fine-tuned on the ETPC dataset from Wahle et al. (2023), and a DPO-trained Llama model trained by us on the ranked data from the second phase of the annotation and based on the fine-tuned model.

3.2 Annotation

We recruited 15 annotators to perform ratings in two phases. The annotators were students from computer science, data science, and related programs with self-accessed English language skills of C1 or C2 and an average age of 24. In two one-hour meetings, they were trained with detailed guidelines. Appendices A and E detail the annotation process and the guidelines.

In the first phase, the annotators were shown a pair of original and paraphrased candidates and APTs to be applied. The annotators were asked whether the given sentence pair had approximately the same meaning, to specify if a specific APT was applied correctly, and to determine the groups of any additional APT if multiple changes were made to the original sentence or if the given APT was not applied correctly. Annotators were also asked to highlight the word position of change if the paraphrase type had been applied correctly (i.e., a span of multiple words that can be consecutive or disjoint). Figure 14 in Appendix E shows and example of the system. We also included five manually created gold examples to certify the annotations were carried out carefully and to check for agreement among annotators; no annotator answered more than one gold question incorrectly; the details can be found in Appendix B.1. On average, the median time to complete the annotation of one paraphrase pair was 74 seconds, with the 25th percentile being 47 seconds and the 75th percentile 121 seconds. Each paraphrase type was annotated by one annotator, except for our gold examples, which were given to all annotators. These annotations compose the APTY$_{\text{Base}}$ dataset.

In the second phase, paraphrases with successfully applied paraphrase types were ranked from best to worst (Flynn and Marley, 2014). The list was discarded if less than two generations were successful for a given sentence. In total, 80 of 100 possible lists remained, and the annotators were then asked to rank them according to their preferences. Five annotators gave their preferences for each list, leading to $5 \times 80 = 400$ preference annotations composing the APTY$_{\text{Ranked}}$ dataset.

For the evaluation of the DPO-tuned Llama model, two annotators performed the first phase again, but each annotated every one of the 300 (3 models x 10 APT x 10) generations.

4 Experiments

Q1. *How successful is ChatGPT, on average, at generating specific paraphrase types? Which paraphrase types can it already produce with high success, and which ones do they struggle with?*

Summary *Change of Order*, *Semantic Changes*, and *Same Polarity Substitution* can be generated with decent success rates, while more complicated changes, particularly those involving deeper grammatical changes at multiple points like *Derivational Changes* pose a problem.

Ans. Based on the annotation of the first phase, we measure the success rate of the different prompt and paraphrase type combinations to assess which APTs ChatGPT could already generate well and which APTs are more challenging. The results are shown in Figure 3. We color the different generation strategies and group them by APT on the x-axis.

Humans often judge the generation as successful when prompting ChatGPT to generate *Change of Order* (82%), *Semantic Changes* (82%), and *Same Polarity Substitution* (78%). In contrast, *Derivational Changes* (46%), *Subordination and Nesting Changes* (38%), and *Synthetic/Analytic Subsitution* (34%) show low success rates. While *Change of Order* and *Same Polarity Substitution* require small changes and understanding on the sentence level or of grammatical concepts, *Derivational Changes* require an understanding of the parts-of-speech of the affected lexical unit. *Subordination and Nesting Changes* similarly require a nuanced understanding of sentence structure and grammar. Consider the following example of a paraphrase using both a

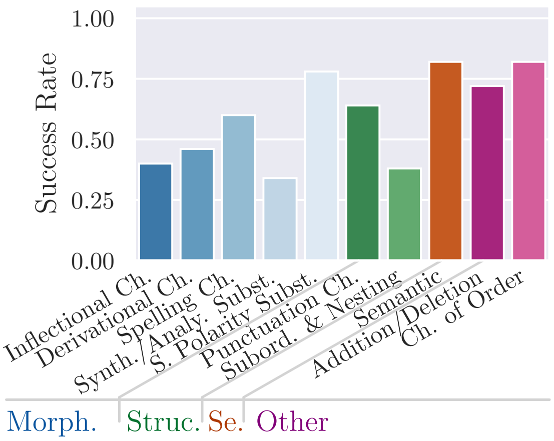


Figure 2: The success rate on the y-axis in generating the specific APT on the x-axis across all tested prompting techniques of ChatGPT.

Derivational change (a) and *Same Polarity Substitution* (b).

> Original: "A smiling$_{(a)}$ senior policeman$_{(b)}$ shook hands with Mr Laczynski"
> Paraphr.: "A senior police officer$_{(b)}$ smiled$_{(a)}$ and shook hands with Mr Laczynski"

Note that for *Same Polarity Substitution*, no change in sentence structure or on word level somewhere else in the sentence is necessary, i.e., the change is locally constrained and does not require understanding beyond synonym relations. However, changing *smiling* to *smiled* makes it necessary to shift the word's position, as it is now used as a verb. It needs to be conjugated to match the rest of the sentence, which is in the simple past tense and third person singular and requires the addition of *and* to keep the sentence correct. The high success rate of 82% for *Semantic Changes* is surprising as a semantic shift is typically complex. But the change also often involves idioms and other common turns of phrases, which might be seen often in training data. Structure-based changes present further difficulty for the model. These changes require a deep grammatical understanding of the text.

In the general paraphrase generation task without APTs, models might seem to generate diverse paraphrases but repeatedly use a small selection of APTs that convince annotators of diversity as a form of reward hacking (Casper et al., 2023). One explanation could be that LLMs are more familiar with some paraphrase types than others, i.e., paraphrasing is also biased towards certain types in the training data and fine-tuning steps (Zhou et al., 2022a). Without diversity of paraphrase types in training, models might be restricted in linguistic diversity, even in cases where underrepresented paraphrase types would be advantageous, similar to how it hinders them in detection tasks (Zhou et al., 2022a).

Q2. *How do different prompting techniques affect the success in generating paraphrase types?*

Summary CoT is the most successful prompting strategy, while one-shot prompting led to the highest error rate.

Ans. Figure 3 decomposes Figure 2 into individual prompting techniques to compare how different prompt techniques influence the success rate of ChatGPT in generating paraphrase types. This allows us to observe how familiar ChatGPT is with the tasks (zero-shot performance) and how much reasoning of ChatGPT improves performance (CoT). For detailed values, refer to Appendix B.1 Table 5. Depending on the APT, prompt and type combinations have notable differences in success rates. For example, while CoT is relatively successful at *Same Polarity Substitution* when zero-shot prompted, ChatGPT struggles with this task.

Human annotators found that the CoT prompt was applied successfully in 69% of the cases, followed by few-shot (63%), zero-shot (61%), and one-shot prompt (55%). The fine-tuned model had the worst performance with only 50%. As the fine-tuned model is trained on applying multiple APTs at once, applying a single change might have compromised its performance. The one-shot prompt might overly rely on the given example and perform worse on APTs where different changes belong to the same APT. For instance, a verb might be changed to an adjective in the one-shot prompt for generating *Derivational Changes*.

Our findings suggest that providing multiple examples of the same type and reasoning about the generation improves APT generation performance in LLMs. For some examples, ChatGPT profits from reasoning about the task at hand, but often, the performance remains poor (e.g., *Synthetic/Analytic Substitution*), suggesting a need to improve model understanding.

Q3. *Are there qualitative differences between prompt methods? Do humans favor the generation of certain prompt styles more than others?*

Summary Annotators favor the generations from few-shot prompting the most, while generations

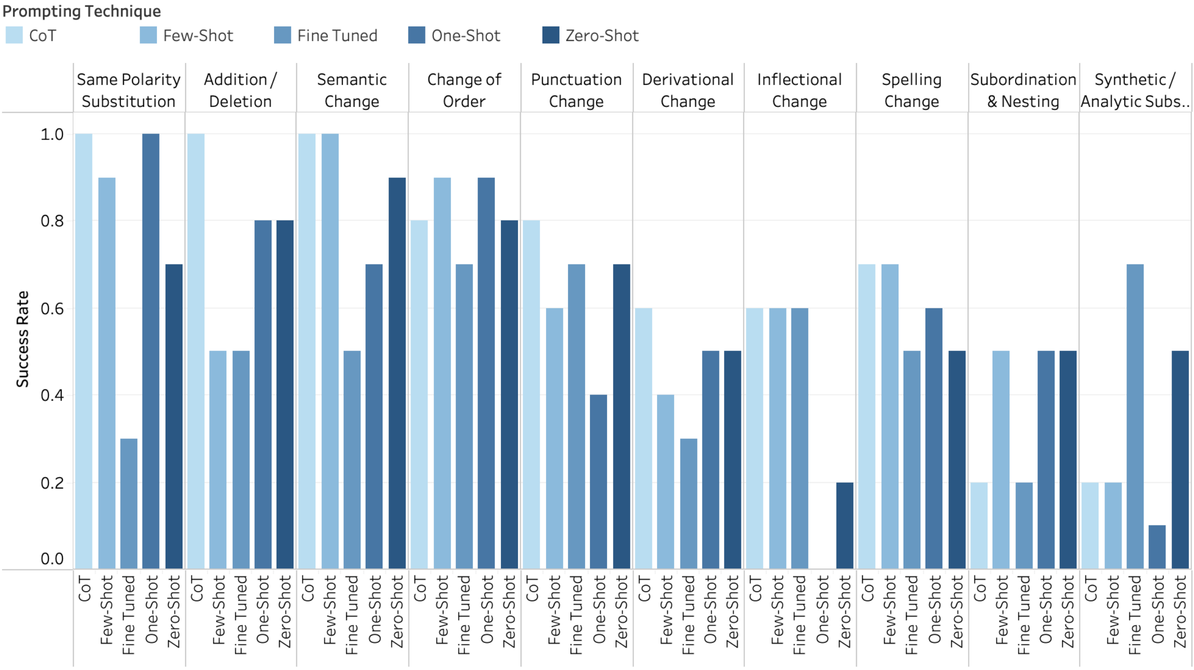


Figure 3: The success rate in generating a specific APT for different prompting techniques of ChatGPT.

from CoT and the Fine-Tuned model are generally preferred less.

Ans. Besides whether a change is applied correctly, the resulting paraphrases have varying degrees of quality. Humans might prefer one paraphrase over another, although they have applied the same type (e.g., some grammatical structures are easier to grasp than others). These preferences can indirectly inform what criteria humans use to judge paraphrases, e.g., at which position a change happens or how creative a change is. We investigate whether different prompt techniques (e.g., CoT, zero-shot) produce paraphrases rated as more or less favorable by humans.

Of all paraphrases where humans have annotated that the APT was correctly applied, we ask annotators to rank these paraphrases from best to worst according to their preferences. The best paraphrase is ranked first, i.e., lower numbers are better. To avoid penalizing prompt methods that produced fewer successful generations, we did not rank unsuccessful generations, and we only looked at the rate at which a generation from one prompt technique was preferred compared to another when both generated an APT successfully. We give the average ranks of the different prompt techniques and the proportion of times one generation is preferred to another in Table 1. Based on the average rank, few-shot and one-shot generated the most preferred paraphrases, zero-shot and CoT seem to be in the middle, and fine-tuned is ranked worst. For the direct comparison, the trend repeats; we see that few-shot is generally favored compared to the other approaches and fine-tuned and CoT are often perceived as worse.

As generations from few-shot prompts are perceived best on average, it suggests that practical examples allow ChatGPT to generate natural-sounding paraphrases more than reasoning. Further, while CoT has one of the highest success rates in generating APTs, it seems to generate paraphrases of lesser quality. One reason for the poor reception of CoT-generated paraphrases might be that more consideration of the formal aspects of a paraphrase leads to less natural outcomes, which are not perceived well by the annotators. The generations of the fine-tuned model were also rated lower than prompting the general model. We noted that the fine-tuned model struggled to generate specific APT changes; this seems to extend to natural-sounding paraphrases with single APTs. That highlights the importance of high-quality datasets tailored to the specific tasks for fine-tuned models. Our preference data can be used to improve model generations via techniques like RLHF or DPO.

Q4. *Which mistakes does ChatGPT make when generating specific paraphrase types? Does it regenerate identical sentences, apply other types than those instructed, or change the meaning?*

Prompt	CoT	Few-Shot	Fine Tuned	One-Shot	Zero-Shot	Avg. Rank
CoT	0.00	0.27	0.63	0.33	0.40	2.27
Few-Shot	0.55	0.00	0.66	**0.44**	0.55	**1.84**
Fine-Tuned	0.34	0.29	0.00	0.28	0.37	2.54
One-Shot	**0.58**	**0.38**	**0.68**	0.00	**0.57**	1.89
Zero-Shot	0.54	0.32	0.59	0.33	0.00	2.21

Table 1: Proportion of times the generation of the row technique is preferred to the column technique.

Summary The most common type of wrong application concerns the model making morpho-lexical changes instead of the requested changes and the boundaries between different types of the same group seem larger than between different groups. Applying the wrong APT is the most frequent reason for error by far.

Ans. To determine what kind of errors the models are most likely to make, we asked the annotators to indicate why a paraphrase type was wrongly applied and if other changes were applied instead. If an annotator judged the generated APT from ChatGPT as a mistake, we provided four possible explanations: the sentences were identical, a different change was applied, the generation was nonsensical, and other reasons. The relative frequencies of these error types are listed in Table 2. Applying the wrong type is the most frequent reason for the error, which happens in 60% of the erroneous cases, while not performing any change is also a common source of error in roughly 20% of the wrong annotations. Only 17% of wrongly applied APTs change the meaning from the original sentence, meaning a paraphrase is usually generated even if ChatGPT fails to use the given APT. An open question is whether giving the model a way to refuse to produce a paraphrase in the prompt if it is unsure, i.e., if it could express uncertainty, would reduce the number of erroneous generations. We leave the investigation of this sub-question to future work.

ChatGPT is good at generating paraphrases with the same meaning but fails to understand the underlying linguistic properties involved. It rarely changes the meaning but often changes the wrong aspect of the sentence or does not change the sentence at all. Since applying a different type than instructed is the most common issue, we investigate which types are most often mistaken or applied in the following question.

Error	Rate in %
Identical sentences	21.3
Wrong type applied	60.0
Nonsense	9.9
Other	9.9

Table 2: Error types and their occurrence in percent given incorrect application of an APT.

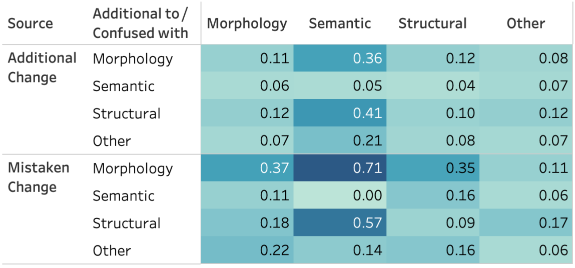
Q5. *Which paraphrase types is ChatGPT confusing most with another? Do we see a correlation between certain paraphrase types?*

Summary We see intra-group errors less often than inter-group errors and morpho-lexical changes are applied erroneously most often.

Ans. The previous answer shows that ChatGPT often applies different APTs if an error occurs. ChatGPT also applies changes besides the desired one in the case of successful applications. In such instances, we can examine the relationship between the different APTs from the perspective of ChatGPT, assessing whether and how they correlate. For these cases, the annotators indicated which kind of APT groups these erroneous or additional changes belong to. We used four major groups for annotators to indicate confusion: morpho-lexical, semantic, structural, and other changes; the definitions can be found in the Appendix E.4.

We show the confusion matrices in Figure 4, where 'Additional Change' indicates cases where the correct APT is applied along with one or more additional APTs. In contrast, 'Mistaken Change' illustrates instances where the model erroneously applies an APT. For the numbers in Figure 4, we count the number of additional erroneous APT applications that belong to the group given at the x-axis, where an APT that is part of the group at the y-axis should have been applied. Then, we divide this count by the total number of paraphrases annotated in the group.

For morpho-lexical changes, the additional changes are evenly spread and happen comparatively rarely. If ChatGPT makes an erroneous change, it is often a different morpho-lexical change. This is also the case for the other groups, e.g., the most common type of wrong application concerns the model making morpho-lexical changes instead of the requested changes. These are probably the most straightforward changes to make without changing the meaning of the sentence



Source	Additional to / Confused with	Morphology	Semantic	Structural	Other
Additional Change	Morphology	0.11	0.36	0.12	0.08
	Semantic	0.06	0.05	0.04	0.07
	Structural	0.12	0.41	0.10	0.12
	Other	0.07	0.21	0.08	0.07
Mistaken Change	Morphology	0.37	0.71	0.35	0.11
	Semantic	0.11	0.00	0.16	0.06
	Structural	0.18	0.57	0.09	0.17
	Other	0.22	0.14	0.16	0.06

Figure 4: Confusion matrices for additional and erroneous changes. The column gives the intended APT, the row the additional or wrongly applied APT.

and also the most common change, e.g., the most common types in the ETPC dataset are morpholexical changes. Generally, the boundaries between different types of the same group seem larger than between different groups. One reason might be that models rarely see isolated changes in the training data (i.e., sentences that only differ by a single APT), leading to a poor understanding and associating them with other changes. We also found that the confusion correlates with the difficulty of the paraphrasing task; the more difficult the task is, the more likely it is that other changes are made either in addition or erroneously. We provide more detailed investigations in the additional research questions in Appendix B.1.

Q6. *Can the human preference data improve the generation of APTs?*

Summary Training a model with the collected preference data improves the performance of the model. We can see similar trends in LLama as we see in ChatGPT.

Ans. We used the collected preferences to train a LLama3 7B model (Dubey et al., 2024), which was fine-tuned on the ETPC dataset from (Wahle et al., 2023), via DPO (Rafailov et al., 2023). We compare the results in generation success across LLama models (base, fine-tuned, DPO-trained); the results for each APT are shown in Figure 5. Across all generations, DPO performs best with a rate of 34% of successful APT applications, followed by the base model with 14% and the fine-tuned model with 11%. We can also see that, while success rates are generally much lower than with the much larger model, the trends from ChatGPT also hold across the tested LLama 7B variants, i.e., success rates across all variants are best for types where changes are local and where the model has freedom, like *Semantic Changes* (37%), and *Addition/Deletion* (43%). At the same time, they are worse for

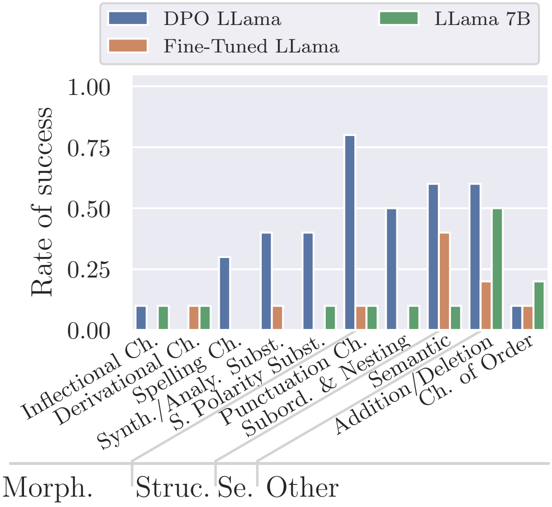


Figure 5: The success rate in generating a specific APT for different LLama 7B Models

changes that require a deeper grammatical understanding like *Derivational Changes* (8%). We see that the DPO training improves generation success by a large margin, and the collected preference data improves the understanding of the model significantly. The success rates for difficult changes like *Synthentic/Analytic substitution* (10% to 40%) or *Subordination and Nesting Changes* (10% to 50%), e.g., changes which we have seen ChatGPT struggles to perform as well, received a big boost via DPO training, suggesting an increased understanding of the model of these concepts. Surprisingly, fine-tuning on the ETPC dataset does not improve generation success in this task. One explanation might be that in ETPC multiple changes are performed at once, and the model does not learn the specific mechanics of a single change due to the notice introduced by other changes.

5 Final Considerations

In this work, we contributed to the understanding of how LLMs can generate variability in linguistic expression through paraphrase types. Using different prompting techniques, we generated 800 paraphrase candidates. We asked 15 participants to annotate these paraphrase candidates according to whether the paraphrase type was correctly applied by ChatGPT , and the kind of errors made by ChatGPT at generation. We made the annotations and human preference rankings available through two new datasets (i.e., $\text{APTY}_{\text{Base}}$and $\text{APTY}_{\text{Ranked}}$).

We found that CoT prompting generally outperforms other methods for generating paraphrases with specific APTs. This suggests that asking ChatGPT to reason about complex paraphrase types (e.g., *Synthetic/Analytic Substitution*) is ineffective, as the required understanding of these APTs is missing. When ChatGPT made mistakes or added additional changes, they most commonly chose APTs from the morpholexical changes. This is likely because these changes are the most common and dominate paraphrasing datasets. We also found that for non-morpholexical changes, ChatGPT rarely confuses changes that can be categorized under the same group (e.g., confusion of one structural change with another instead of a non-structural change). One reason might be that ChatGPT pays increased attention to the properties related to the change so that the retaining of other properties is weighed less. APTY$_{\text{Ranked}}$ opens an interesting avenue for further research in improving the generation and identification quality of APTs using techniques such as DPO or RHLF. Our experiments with Llama 2 7B-based models show a marked advantage when using human preference data from APTY$_{\text{Ranked}}$and DPO and show similar trends to ChatGPT. Alternatively, our dataset can be used as a benchmark to develop metrics that better correlate with human preferences.

Limitations

Although we extend the research in paraphrase types, this study has a few limitations. We consider only a selection of APTs and could not investigate all described types due to the resource-intensive work of human annotation. We focused on the most common ones in the ETPC dataset, which should give a decent proxy for the most relevant types and selected a sample to ensure diversity in the examined types. Additionally, we are limited to one annotation per task for the questionnaire due to effort risking a big variance. However, we used gold examples annotated by each annotator to check the agreement between annotators and to ensure that each annotator performed the annotation attentively. We focused on generations from ChatGPT-3.5-turbo-0613, as it is one of the most used models, and this is the first investigation of APT paraphrase generation combined with instruction-trained LLMs. We evaluated DPO only for a Llama 7B model; while more models could be interesting, the existing evaluation already shows a marked value of using our preference data. Including additional and larger models would make the human evaluation expensive, so we leave this investigation to future work.

References

Faisal Alvi, Mark Stevenson, and Paul Clough. 2021. Paraphrase type identification for plagiarism detection using contexts and word embeddings. *International Journal of Educational Technology in Higher Education*, 18(1).

Ion Androutsopoulos and Prodromos Malakasiotis. 2009. A survey of paraphrasing and textual entailment methods. *CoRR*, abs/0912.3747.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Alberto Barrón-Cedeño, Marta Vila, M. Antònia Martí, and Paolo Rosso. 2013. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics*, 39(4):917–947.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. FLEX: unifying evaluation for few-shot NLP. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 15787–15800.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *ArXiv preprint*, abs/2307.15217.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert systems with applications*, 165:113679.

Terry N Flynn and Anthony AJ Marley. 2014. Best-worst scaling: theory and methods. In *Handbook of choice modelling*, pages 178–201. Edward Elgar Publishing.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 294–297, New York, NY, USA. Association for Computing Machinery.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1022–1033, Online. Association for Computational Linguistics.

Venelin Kovatchev, M. Antònia Martí, and Maria Salamó. 2018. ETPC - a paraphrase identification corpus annotated with extended paraphrase typology and negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

OpenAI. 2023. GPT-3.5-turbo-06. `https://openai.com/blog/gpt-3-5-turbo`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Hemant Palivela. 2021. Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights*, 1(2):100025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Marta Vila, M. Antònia Martí, and Horacio Rodríguez. 2014. Is This a Paraphrase? What Kind? Paraphrase Boundaries and Typology. *Open Journal of Modern Linguistics*, 04(01):205–218.

Jan Wahle, Bela Gipp, and Terry Ruas. 2023. Paraphrase types for generation and detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12148–12164, Singapore. Association for Computational Linguistics.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How large language models are transforming machine-paraphrase plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 952–963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jan Philip Wahle, Terry Ruas, Yang Xu, and Bela Gipp. 2024. Paraphrase types elicit prompt engineering capabilities. *ArXiv preprint*, abs/2406.19898.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.

Sam Witteveen and Martin Andrews. 2019. Paraphrasing with large language models. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 215–220, Hong Kong. Association for Computational Linguistics.

Chao Zhou, Cheng Qiu, and Daniel E. Acuna. 2022a. Paraphrase identification with deep learning: A review of datasets and methods. *Preprint*, arXiv:2212.06933.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. *ArXiv preprint*, abs/2211.01910.

A Additional Annotation Information

The annotators were paid the usual rate for student workers, corresponding to at least minimum wage in Germany. Consent was explicitly requested via email to publish annotation data and aggregated demographic data. No ethics review board was consulted as we judged the collected data as unproblematic. The resulting datasets will be released under CC BY 4.0.

Besides the questions mentioned in 3.2, the annotators also indicated whether additional text was generated, i.e., whether the model generation contained more than just the indicated paraphrase. The annotators were instructed to ignore additional text if it could be clearly separated from the paraphrase candidate. For example, sometimes the generated text might be in quotes, or the model might add a phrase like "The paraphrased sentence is:" in front of the generation. The question was asked to clean up the data and provide a better foundation for future work.

For annotation phase one, we also used gold questions to evaluate whether the annotators understood the assignment and checked their agreement. The gold examples include completely correct applied paraphrase types, a correctly applied paraphrase with additional applied paraphrase types, a non-paraphrase, and a paraphrase with a different applied APT. The Fleiss Kappa for sense-preservation was 0.92, for correct application 0.68, and for failure reasons 0.61. Determining if a type was applied correctly can depend on details that can be missed, similarly failure reasons are not always trivial to see and depend on the previous answers. There was no annotator whose answers to the gold questions suggested a systematic lack of understanding besides normal human errors, e.g., no annotator answered more than one sense-preserving question incorrectly. For the second annotation phase, i.e., the preferences, Kendall's coefficient of concordance is 0.52, indicating a moderate agreement on preferences across annotators. Precise preferences are likely highly individual, but trends exist. The Fleiss Kappa for correct application is 0.80 for the DPO annotation with two annotators, showing high agreement.

A.1 Full List of Considered APT

After filtering, the considered paraphrase types are:

- *Addition/Deletion*,
- *Same Polarity Substitution (cont.)*,
- *Syntheticanalytic Substitution*,
- *Change of Order*,
- *Punctuation Changes*,
- *Spelling Changes*,
- *Inflectional Changes*,
- *Subordination and Nesting Changes*,
- *Semantic-based Changes*,
- and *Derivational Changes*.

A.2 Dataset

The given features in the datasets are given in Table 3 for APTY$_{\text{Base}}$ and in Table 4 for APTY$_{\text{Ranked}}$. For APTY$_{\text{Ranked}}$ we give the ranking pairwise and follow the format of Anthropic for RHLF (Bai et al., 2022) and add information so that the full information can easily be reconstructed.

B Additional Experiments and Details

B.1 Additional Questions

AQ1. *How does ChatGPT perform on examples perceived as easy or hard by humans?*

Ans. The few-shot approach works better in cases where humans evaluate the task as hard, while the CoT-prompted model performs poorly for hard examples

Applying different APTs can be challenging for humans, depending on the precise APT and the base sentence. We explore if the human perceived difficulty of paraphrases also relates to the performance of ChatGPT; that is, are paraphrasing tasks difficult for humans also difficult for ChatGPT?

To answer this question, we asked our annotators, for each presented generation, whether they would find applying the paraphrase type to a given sentence *easy* or *hard*. Then, we computed the rate of successfully generated paraphrases for different approaches depending on the estimated difficulty of the task, as shown in Figure 6. On the x-axis, the tasks are split according to rated difficulty, and the y-axis gives the rate of successful application.

We observe that the few-shot approach works better in cases where humans evaluate the task as hard, seeming to profit from a human perspective more difficult task. The CoT-prompted model performs poorly for hard examples, with 23% points less generation success. One reason might be that human annotators evaluate examples as hard, which

Attribute	Example
meta	
id	106
annotator	5
generation	
APT	AdditionDeletion
Kind	One-Shot
original	They had . . .
paraphrase-text	They had . . .
annotation	
paraphrase	True
applied-correctly	True
correct-format	True
hard	False
failure	
identical	False
other	False
nonsense	False
otherchange	False
additional	
morph	False
struct	False
semantic	False
other	False
mistaken	
morph	False
struct	False
semantic	False
other	False
marked-text	
start	97
end	109
text	additionally

Table 3: Features of the annotation for APTY$_{\text{Base}}$.

Attribute	Example
meta	
id	100
annotators	[8,7,11,12,14]
original	They had . . .
pairwise	
chosen	They had . . .
ranks	[1,1,2,3,1]
id	106
rejected	Adding that . . .
id	104
ranks	[3,4,4,5,4]

Table 4: Features of the annotation for APTY$_{\text{Ranked}}$.

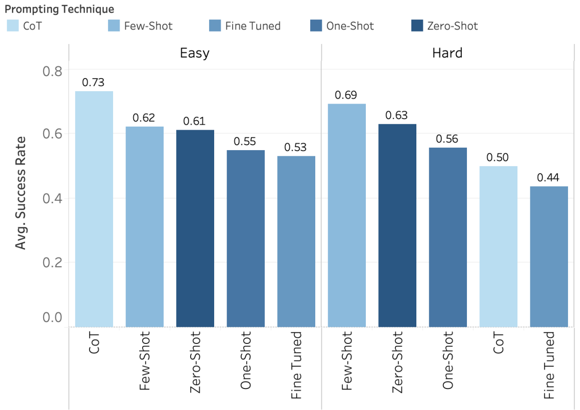
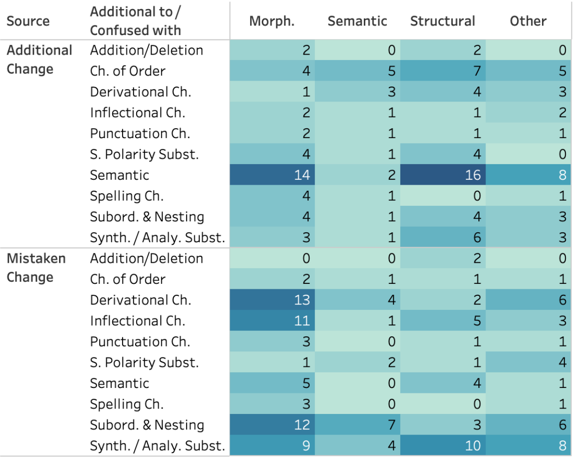


Figure 6: Success rate in generating a specific APT based on the human-perceived difficulty of example.

are difficult to reason about, and this extends to reasoning attempts from LLMs. Similarly, the fine-tuned model also performs worse for hard examples. Surprisingly, the performance of the other prompt paradigms seems largely independent of the difficulty. The results suggest that explicit reasoning about paraphrasing tasks that are hard for humans is also hard for LLMs. If more complex insights are required, as judged by humans, they fail more often as they lack the required understanding of the concepts associated with the APTs. Still, the LLMs exhibit some reasoning capability for APTs, even if they are led astray by it in difficult scenarios.

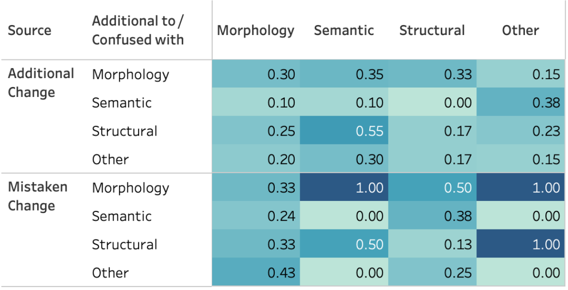
AQ2. *How does the detailed confusion matrix look? Do specific APTs get confused more often than other APTs?*

Ans. To get a more granular view of the confusion, we also looked at the detailed confusion based on single APTs as sources instead of groups. The result is shown in Figure 7. We plotted the absolute numbers of additional/erroneous changes, as here, all groups are the same size, i.e., for each APT, there are 50 generations. We have already noticed that most confusion happens intra-group and not inter-group. There are big differences inside the groups, depending on the APT. For instance, for morpho-lexical changes, *Same Polarity Substitution* is rarely confused, while *Inflectional and Derivational Changes* are often confused for different morpho-lexical APTs. As *Same Polarity Substitution*, i.e., the change of one unit for a synonym, is a very common APT, the model seems to have a very good understanding of it. Still, additional changes to *Same Polarity Sub.* are made at a comparable or even larger rate than other group members. So, even with a good representation in



Source	Additional to / Confused with	Morph.	Semantic	Structural	Other
Additional Change	Addition/Deletion	2	0	2	0
	Ch. of Order	4	5	7	5
	Derivational Ch.	1	3	4	3
	Inflectional Ch.	2	1	1	2
	Punctuation Ch.	2	1	1	1
	S. Polarity Subst.	4	1	4	0
	Semantic	14	2	16	8
	Spelling Ch.	4	1	0	1
	Subord. & Nesting	4	1	4	3
	Synth. / Analy. Subst.	3	1	6	3
Mistaken Change	Addition/Deletion	0	0	2	0
	Ch. of Order	2	1	1	1
	Derivational Ch.	13	4	2	6
	Inflectional Ch.	11	1	5	3
	Punctuation Ch.	3	0	1	1
	S. Polarity Subst.	1	2	1	4
	Semantic	5	0	4	1
	Spelling Ch.	3	0	0	1
	Subord. & Nesting	12	7	3	6
	Synth. / Analy. Subst.	9	4	10	8

Figure 7: Detailed confusion matrix with absolute values, the maximum is 50 in each case. The column gives the intended APT.



Source	Additional to / Confused with	Morphology	Semantic	Structural	Other
Additional Change	Morphology	0.30	0.35	0.33	0.15
	Semantic	0.10	0.10	0.00	0.38
	Structural	0.25	0.55	0.17	0.23
	Other	0.20	0.30	0.17	0.15
Mistaken Change	Morphology	0.33	1.00	0.50	1.00
	Semantic	0.24	0.00	0.38	0.00
	Structural	0.33	0.50	0.13	1.00
	Other	0.43	0.00	0.25	0.00

Figure 8: Confusion matrix only for hard examples. The columns give the intended APT.

the trainings data, it is difficult for the model to generate only the requested type. The big intra-group differences suggest that it makes sense to characterize the common types in as much detail as possible for any paraphrasing-related task, as model performance differs at the APT group and the APT level.

AQ3. *Is there a correlation between perceived hardness of a task and confusion?*

Ans. We have already looked at how difficulty relates to model performance regarding generation success. We also wanted to investigate how it relates to confusion. Therefore, we plotted the same confusion matrix for tasks rated as hard in Figure 8. It follows similar trends to the unrestricted confusion matrix but with remarkably higher confusion rates across the board.

This further supports the assumption that tasks rated hard by humans are also harder for LLMs. If a task is rated hard, ChatGPT is likelier to perform additional, unwanted changes and even more likely to perform the wrong kind of change.

AQ4. *How much do humans and ChatGPT agree on paraphrase type evaluation?*

Ans. Human annotation at scale is costly and difficult to perform. Works like (Huang et al., 2023) have raised the possibility of using LLMs for annotation tasks. Besides potentially posing as an alternative to human annotation, LLM annotation might also give insight into how well LLMs understand paraphrase types from a detection perspective. Therefore, we want to see how well LLM annotation aligns with human annotations. To that end, we used the same annotation document provided to the annotators to prompt ChatGPT and asked it to perform the same tasks the human annotators performed.

We found little agreement between the human annotation and the answers produced by ChatGPT. For evaluating whether the given text pair is a paraphrase, the annotations of humans and ChatGPT are identical in 54 % of the cases. Similarly, to check whether a paraphrase type is applied correctly, humans and ChatGPT only agree in 43% of the cases. In evaluating task difficulty, they only agreed in 50 % of the cases. As these annotation tasks were binary choices, i.e., yes or no and easy or hard, ChatGPT seems to only agree by chance with the human annotation (even with the same instructions).

While an improvement in alignment is probably possible by different prompting techniques and prompts more tailored to the setting, the results still show that ChatGPT has trouble understanding the phenomena from a detection perspective. The models can also not estimate the task difficulty from a human perspective, i.e., ChatGPT can not estimate what tasks humans find complex or easy.

B.2 Result Details

The precise values for the success rates corresponding to Figure 3 of the different prompt methods are given in Table 5.

C Prompts

In the following, we present the prompts used to generate the paraphrases. The placeholder {definition} stands for the definition of an APT, {sentence} stands for the base sentence that should be altered, {example} is replaced by an example application of the given APT, and lastly, {type} is a stand-in for the name of the APT. The examples are constructed from paraphrase pairs in ETPC by manually edit-

Group	APT	CoT	Few-Shot	Fine-Tuned	One-Shot	Zero-Shot
Morph.	Change of Order	0.8	**0.9**	0.7	**0.9**	0.8
	Derivational Change	**0.6**	0.4	0.3	0.5	0.5
	Inflectional Change	**0.6**	**0.6**	**0.6**	0.0	0.2
	Spelling Change	**0.7**	**0.7**	0.5	0.6	0.5
	Same Polarity Subst. C.	**1.0**	0.9	0.3	**1.0**	0.7
Struct.	Punctuation Change	**0.8**	0.6	0.7	0.4	0.7
	Subordination and Nesting	0.2	**0.5**	0.2	**0.5**	**0.5**
Semantic	Semantic-Based Change	**1.0**	**1.0**	0.5	0.7	0.9
Other	Addition/Deletion	**1.0**	0.5	0.5	0.8	0.8
	Synthetic/Analytic Subst.	0.2	0.2	**0.7**	0.1	0.5
Average		**0.69**	0.63	0.5	0.55	0.61

Table 5: The success rate in generating a specific APT for different prompting techniques of ChatGPT. The most successful model(s) are marked in **bold** for each APT.

ing a pair such that only one change of type l is present.

The following figures show the prompt template for zero-shot (Figure 9), one-shot (Figure 10), few-shot (Figure 12) and chain-of-thought (Figure 13) prompts. The fine-tuned models are prompted according to (Wahle et al., 2023) using the sentence and the types. Lastly Figure 11 shows one concrete one-shot prompt and the corresponding model response.

D APT Definitions

Addition/Deletion: Addition/Deletion consists of all additions/deletions of lexical and functional units.

Same Polarity Substitution (contextual): Same Polarity Substitution consists of changing one lexical (or functional) unit for another with approximately the same meaning. Among the linguistic mechanisms of this type, we find synonymy, general/specific substitutions, or exact/approximate alternations.

Syntheticanalytic substitution: Synthetic/analytic substitution consists of changing synthetic structures for analytic structures, and vice versa. This type comprises mechanisms such as compounding/ decomposition, light element, or lexically emptied specifier additions/deletions, or alternations affecting genitives and possessive.

Change of order: Change of order includes any type of change of order from the word level to the sentence level.

Punctuation changes: Punctuation and format changes consist of any change in the punctuation or format of a sentence (not of a lexical unit, cf. lexicon-based changes).

Inflectional Changes: Inflectional changes consist of changing inflectional affixes of words

Spelling changes: Spelling and format changes comprise changes in the spelling and format of lexical (or functional) units, such as case changes, abbreviations, or digit/letter alternations.

Subordination and nesting changes: Subordination and nesting changes consist of changes in which one of the members of the pair contains a subordination or nested element, which is not present, or changes its position and/or form within the other member of the pair.

Semantic based Changes: Semantics-based changes are those that involve a different lexicalization of the same content units. These changes affect more than one lexical unit and a clear-cut division of these units in the mapping between the two members of the paraphrase pair is not possible.

Derivational Changes: Derivational Changes consist of changes of category with or without using derivational affixes. These changes imply a syntactic change in the sentence in which they occur.

E Annotation Guidelines

Dear annotators, In this experiment, we want to explore how humans perceive generated paraphrases. Paraphrases are texts expressing identical meanings that use different words or structures. However, instead of looking at general paraphrases, we want to examine specific paraphrase types. Paraphrase types, also known as atomic paraphrase types, are specific lexical, syntactic, and semantic changes that can be grouped into a hierarchical topology. Each annotator will work on roughly 40 examples.

Your role will be to look at paraphrases and evaluate if these generations were successful.

Disclaimer: No sensitive information will be collected during annotation.

Zero-Shot Prompt Template

In this task you will be given a definition of an alteration and an input sentence in ”’. Your task is to apply the alteration to the given sentence, while preserving the original meaning of the sentence. That means, the new sentence should be a paraphrase of the old sentence. Output the altered sentence and check that the nothing except the alterations you made was changed in the sentence and the alteration given is indeed applied.

Alteration: {definition}

Input: ”’{sentence}”’

Figure 9: The zero-shot template to paraphrase sentences.

One-Shot Prompt Template for APT

In this task you will be given a definition of an alteration, an example of the alteration applied to a sentence and an input sentence in ”’. Your task is to apply the alteration to the given sentence, while preserving the original meaning of the sentence. That means, the new sentence should be a paraphrase of the old sentence. Output the altered sentence and check that the nothing except the alterations you made was changed in the sentence and the alteration given is indeed applied.

Alteration: {definition}

Example: {example}

Input: ”’{sentence}”’

Figure 10: The one-shot template to paraphrase sentences.

E.1 Paraphrases - Background

Paraphrasing is the act of rephrasing or restating a text or idea using different words while retaining the original meaning. For example, consider the following two sentences:

1. Amrozi accused his brother , whom he called "the witness", of deliberately distorting his evidence. 2. Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.

You can see that these two sentences, while using a different order and wording and possibly expressing different nuances, mean the same thing. We are specifically interested in the linguistic building blocks that make up paraphrases. These can take different forms (e.g., lexical, semantic). We can illustrate some of them with the example above. The paraphrase types that appear are:

Change of order : Change of order includes any type of change of order from the word level to the sentence level. Here, the part “Amrozi accused his brother” is moved in the sentence.

Same Polarity Substitution : Same-polarity substitutions change one lexical (or functional) unit for another with approximately the same meaning. In this example, this happens with “whom” and “to him” and “called” and “Referring” respectively.

Addition/Deletion : This type consists of all additions/deletions of lexical and functional units. The word “only” is added.

Please find the full list of atomic paraphrase types we will consider at the end of the document.

In Figure 14, we show an example of how the annotation process will take place.

E.2 Tasks

You will be given a base sentence, an atomic paraphrase type, and an altered sentence. Additionally, you will be given the definition and an example for each atomic paraphrase type you are working on. Please see Figure 1 for an example annotation.

Example One-Shot Prompt

In this task you will be given a definition of an alteration, an example of the alteration applied to a sentence and an input sentence in ”’. Your task is to apply the alteration to the given sentence, while preserving the original meaning of the sentence. That means, the new sentence should be a paraphrase of the old sentence. Output the altered sentence and check that the nothing except the alterations you made was changed in the sentence and the alteration given is indeed applied.

Alteration:
Addition/Deletion: Addition/Deletion consists of all additions/deletions of lexical and functional units.

Example:
Input: ”’Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.”’
Output: Amrozi accused his brother, whom he only called "the witness", of deliberately distorting his evidence.

Input: ”’They had published an advertisement on the Internet, offering the cargo for sale, he added.”’

Model Output: "They had published an advertisement on the Internet on June 10, offering the cargo for sale, he added."

Figure 11: Specific example for the one shot with APT were the highlighted font indicates the model response.

Please read these additional materials carefully to understand what a valid paraphrase of that type would look like.

A Indicate whether the altered sentence is a paraphrase of the base sentence

B Indicate whether the altered sentence contains a correct application of the given atomic paraphrase type.

  1 If yes:
    - Highlight the given change
  2 If no:
    - Please indicate what went wrong with the application of the paraphrase type:
      i. The sentences are identical
      ii. Nonsense
      iii. Different APTs were applied
      iv. Other reason
  3 If additional or different changes were made than the one initially provided, please identify the groups (see item Atomic Paraphrase Type Groups) of change the additional changes belong to. See the end for a reference of possible categories.

C For each example which you annotated, please indicate whether you find applying the paraphrase to the sentence easy or hard.

D Sometimes additional text, besides the paraphrase, might be given (e.g. “Altered sentence” or explanations about the change). Please indicate whether that was the case. Disregard the additional text for the previous tasks.

The annotation interface will support you as far as possible at the annotation and only show the decisions you need to make depending on your prior annotations, i.e., unnecessary questions will not be displayed.

In case of any questions whether about the process or any specific example, please contact me at {author email}. When you are done with all assigned tasks, please also send me a quick email letting me know.

E.3 Atomic Paraphrase Types:

Addition/Deletion consists of all additions/deletions of lexical and functional units. The word “only” was added/removed in the example below.

a Amrozi accused his brother, whom he called

Few-Shot Prompt Template

In this task you will be given a definition of an alteration, examples of the alteration applied and an input sentence in ”’. Your task is to apply the alteration to the given sentence, while preserving the original meaning of the sentence. That means, the new sentence should be a paraphrase of the old sentence. Output the altered sentence and check that the nothing except the alterations you made was changed in the sentence and the alteration given is indeed applied.

Alteration: {definition}

Examples:
{example}
{example}
{example}
{example}
{example}

Input: ”’{sentence}”’

Figure 12: The few-shot template to paraphrase sentences.

"the witness", of deliberately distorting his evidence.

b Amrozi accused his brother, whom he only called "the witness", of deliberately distorting his evidence.

Same Polarity Substitution changes one lexical (or functional) unit for another with approximately the same meaning. Among the linguistic mechanisms of this type, we find synonymy, general/specific substitutions, or exact/approximate alterations.

a Apple noted that half the songs were purchased as part of albums.

b Apple said that half the songs were purchased as part of albums.

Synthetic/analytic substitution consists of changing synthetic structures for analytic structures, and vice versa. That means that the concept of the predicate is already included in the concept of the subject or the additional predicate is removed. This type comprises mechanisms such as compounding/ decomposition, light element, or lexically emptied specifier additions/deletions, or alterations affecting genitives and possessives. a. About 120 potential jurors were being asked to complete a lengthy questionnaire. b. The jurors were being asked to complete a lengthy questionnaire.

Change of order includes any type of change of order from the word level to the sentence level. See the example in the Background section.

a Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence

b Calling him "the witness", Amrozi accused his brother of deliberately distorting his evidence.

Punctuation changes consist of any change in the punctuation or format of a sentence (not of a lexical unit, like doesn't to does not).

a PG&E Corp. shares jumped $1.63, or 8 percent, to close Friday at $21.51 on the New York Stock Exchange.

b PG&E Corp. shares jumped $1.63 or 8 percent to close Friday at $21.51 on the New York Stock Exchange.

Inflectional Changes consist of changing inflectional affixes of words. In the example, a plural/singular alternation (streets/street) can be observed.

a It was with difficulty that the course of streets could be followed

b It was with difficulty that the course of the street could be followed

Chain-of-Thought Prompt Template

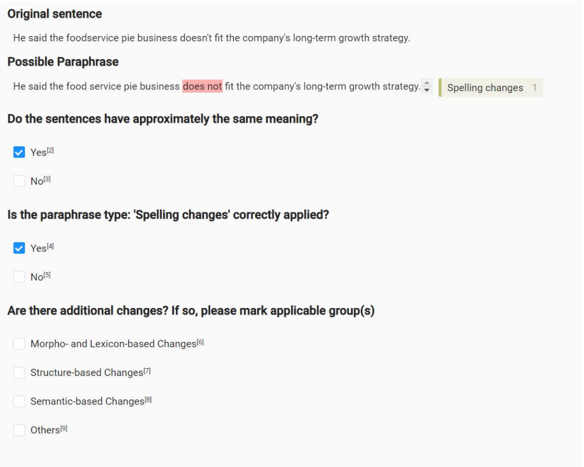
In this task you will be given a definition of an alteration and an input sentence in ”’. Your task is to apply the alteration to the given sentence, while preserving the original meaning of the sentence. That means, the new sentence should be a paraphrase of the old sentence. Think step by step and describe the reason for what part of the sentence you are changing before you do. Output the altered sentence at the end in the format given below, that is with "Output: " in front.

Alteration: {definition}

Example: {example with explanations}

Input: ”’{sentence}”’

Figure 13: The chain-of-thought template to paraphrase sentences.



Original sentence

He said the foodservice pie business doesn't fit the company's long-term growth strategy.

Possible Paraphrase

He said the food service pie business does not fit the company's long-term growth strategy.

Do the sentences have approximately the same meaning?

☑ Yes[2]

□ No[3]

Is the paraphrase type: 'Spelling changes' correctly applied?

☑ Yes[4]

□ No[5]

Are there additional changes? If so, please mark applicable group(s)

□ Morpho- and Lexicon-based Changes[6]

□ Structure-based Changes[7]

□ Semantic-based Changes[8]

□ Others[9]

Figure 14: Annotation example interface.

Spelling changes and format changes comprise changes in the spelling and format of lexical (or functional) units, such as case changes, abbreviations, or digit/letter alterations.

a The DVD-CCA then appealed to the state Supreme Court.

b The DVD CCA then appealed to the state Supreme Court.

Subordination and nesting changes consist of changes in which one of the members of the pair contains a subordination or nested element, which is not present, or changes its position and/or form within the other member of the pair. What is a relative clause in (b) (a spokeswoman for Child) is part of the main clause in Example (a)

a Sheena Young of Child, the national infertility support network, hoped the guidelines would lead to a more "fair and equitable" service for infertility sufferers.

b Sheena Young, a spokesman for Child, the national infertility support network, hoped the guidelines would lead to a more "fair

Semantic based changes are those that involve a different lexicalization of the same content units. These changes affect more than one lexical unit and a clear-cut division of these units in the mapping between the two members of the paraphrase pair is not possible. In the example, the content units referring to increases are present in both sentences, but there is not a clear-cut mapping between the two.

a The largest gains were seen in prices, new orders, inventories and exports.

b Prices, new orders, inventories and exports increased.

Derivational Changes consist of changes of category with or without using derivational affixes. These changes imply a syntactic change in the sentence in which they occur.

a Tyco later said the loan had not been forgiven, and Swartz repaid it in full, with interest, according to his lawyer, Charles Stillman.

b Tyco later said the loan had not been forgiven, and Swartz repaid it fully, with interest, according to his lawyer, Charles Stillman.

E.4 Atomic Paraphrase Type Groups

Morpholexical Changes: These include all changes where a single word or lexical unit is changed. From the paraphrase types you have seen, this includes:

a Inflectional Changes

b Derivational Changes

c Spelling changes and format changes

d Same Polarity Substitution

e Synthetic/analytic substitution

Structure-based Changes: These include all changes that arise from a different structural organization of a sentence. For examples from the paraphrase types you have seen, this includes:

a Subordination and nesting changes

b Punctuation changes

Semantic-based Changes: These include all changes that arise from distributing semantic meaning across different lexical units.

a Semantic Based Changes

Others: Any other changes. For examples from the paraphrase types you have seen, this includes:

a Change of Order

b Addition/Deletion

F Tool Use Acknowledgments

In the conduct of this research project, we used specific artificial intelligence tools and algorithms: ChatGPT, Gemini and DeepL Write to assist with phrasing and editing. While these tools have augmented our capabilities and contributed to our findings, it's pertinent to note that they have inherent limitations. We have made every effort to use AI in a transparent and responsible manner. Any conclusions drawn are a result of combined human and machine insights. This is an automatic report generated with © AI Usage Cards https://ai-cards.org