WinoWhat: A Parallel Corpus of Paraphrased WinoGrande Sentences with Common Sense Categorization

Ine Gevers[1], Victor De Marez[1], Luna De Bruyne[1], Walter Daelemans[1],
[1]CLiPS, University of Antwerp
Correspondence: ine.gevers@uantwerpen.be

Abstract

In this study, we take a closer look at how Winograd schema challenges can be used to evaluate common sense reasoning in LLMs. Specifically, we evaluate generative models of different sizes on the popular WinoGrande benchmark. We release WinoWhat, a new corpus, in which each instance of the WinoGrande validation set is paraphrased. Additionally, we evaluate the performance on the challenge across five common sense knowledge categories, giving more fine-grained insights on what types of knowledge are more challenging for LLMs. Surprisingly, all models perform significantly worse on WinoWhat, implying that LLM reasoning capabilities are overestimated on WinoGrande. To verify whether this is an effect of benchmark memorization, we match benchmark instances to LLM training data and create two test-suites. We observe that memorization has a minimal effect on model performance on WinoGrande.

1 Introduction

While including common sense knowledge in NLP-systems has been a longstanding goal, evaluating this proves to be a non-trivial task. From early on, research used coreference resolution tasks to measure world knowledge and reasoning abilities in machine learning systems. In 2011, the Winograd Schema Challenge was developed, a small test set of 273 instances in which a pronoun has to be disambiguated given two possible antecedents in a short text (Levesque et al., 2012). Where early models failed, transformer-based models quickly achieved remarkable performance on this test. However, researchers objected that this does not prove that models have or use common sense; rather, they could rely on superficial patterns and dataset artifacts (Kocijan et al., 2023). Therefore, a large adversarial benchmark was created: WinoGrande (Sakaguchi et al., 2021). Here, the challenge is to decide which of two options is the correct one in a fill-in-the-blank token '_'. This benchmark is frequently used in combination with other benchmarks to evaluate the performance of new LLMs on common sense reasoning.

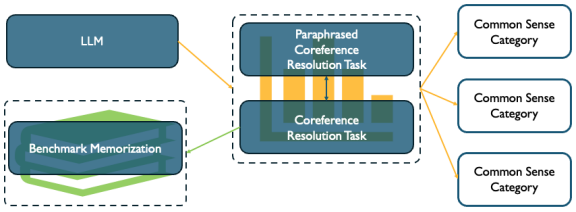


Figure 1: Illustration of the workflow in this study. We evaluate LLMs on WinoGrande, and on its paraphrased variant. We further compare performance per common sense knowledge category, and check for benchmark memorization.

In this study, we evaluate various open-source model families – Gemma 2 (Team et al., 2024), LlaMA 2 (Touvron et al., 2023b), and OPT (Zhang et al., 2022) – on WinoGrande. An overview of the workflow in this study can be found in Figure 1. We present a new parallel corpus of the WinoGrande validation set: WinoWhat, in which we paraphrase each sentence so the '_' token is at the end of the sentence. This transformation makes the task more natural for decoder-only methods and at the same time allows to test whether the performance of LLMs on WinoGrande is robust against paraphrasing (**RQ1**).

While existing works mainly evaluate models on the benchmark as a whole, we define common sense knowledge categories that are crucial to disambiguate the sentence, and evaluate models on each category separately. This allows us to investigate our second research question (**RQ2**): What types of common sense knowledge are more challenging for LLMs? Such an analysis provides insights into the more fine-grained strengths and weaknesses of ML systems on common sense rea-

soning tasks. Instead of creating new benchmarks to focus on one category of common sense knowledge, we suggest using one task setup, which allows us to compare results on different categories without added noise (e.g., different answer formats, different dataset artifacts, etc.).

To verify to what extent data leakage plays a role in LLMs' performance on WinoGrande, we check how many instances of the validation set are included in the pre-training data of LLMs. Further, we create two test-suites; one from which we know that it has been included in LLM pre-training data (i.e., the Winograd Schema Challenge), and one from which we can assume that it has not been seen (i.e., the WinoGrande test set). Comparably to RQ1, we paraphrase each. This answers **RQ3**: What is the role of data memorization in the performance of LLMs on coreference resolution tasks?

The rest of the paper is structured as follows: in Section 2, we summarize relevant literature about disambiguation tasks, common sense categorization, and benchmark memorization. In Section 3, we present in more detail the data, models, evaluation metrics, and the creation of WinoWhat. Further, in Section 4, we present the results of our experiments, and the final Section 5 concludes our research, giving an overview of the findings and suggestions for further research.

2 Related Work

2.1 Coreference resolution and common sense reasoning

Incorporating common sense knowledge into machine learning methods has been a goal since its very beginning (e.g., McCarthy, 1959). However, given the increasing black-box nature of LLMs, it is hard to evaluate whether models have/use common sense knowledge. From early on, sentence disambiguation tasks have been suggested to measure the ability of models to employ common sense knowledge: the assumption being that syntax alone is not enough for the model, and common sense knowledge is needed to determine for instance which noun a pronoun refers to (Browning and LeCun, 2023). An important concept here is bridging, for which the model needs to make inferences about relationships between entities in the world that are not explicitly mentioned in the text (Kobayashi and Ng, 2020; Hou et al., 2018). Since sentence disambiguation and coreference resolution tasks are presented as a proxy to evaluate common sense knowledge, over the years different approaches have been suggested to teach models common sense in order to improve performance on these tasks. In the early 2000s, most coreference resolvers did not include external knowledge sources, relying instead on morpho-syntactic features. The development of large-scale knowledge bases, which were used as features in a baseline resolver, improved results (Rahman and Ng, 2011). Then, with the advent of larger models and more training data, transformer models also relied on external knowledge bases which are generally stored in triplets (Liu et al., 2023).

2.2 The Winograd Schema Challenge

A popular coreference task is the Winograd Schema Challenge (WSC) (Levesque et al., 2012). Based on the work of Winograd (1972), the challenge uses 'schemas' – pairs of twin sentences whose intended meaning can be flipped by changing one word (the 'special word') – to probe ML-methods' ability to reason over natural language[1]. The schemas have three criteria: (1) simple to solve for humans; (2) not solvable by selectional restrictions (i.e., no statistical advantage for one option); (3) google-proof. Over time, newer versions of the challenge were released, which were made in the same format. These datasets are either created by human annotations, or generated by LLMs. For instance, Zahraei and Emami (2024) use Tree-of-Experts to generate new WSC instances, presenting 3,026 LLM-generated sentences. Similarly, Sun and Emami (2024) present EvoGrad, a hybrid method to generate new adversarial WSC instances that feature minor alterations and rewordings using human annotations, ChatGPT, and WordNet. Since WSC and related benchmarks are in English, the challenge was also translated in other languages such as German, Russian, French, Portuguese, and Mandarin Chinese (Emelin and Sennrich, 2021; Amsili and Seminck, 2017; Melo et al., 2019; Bernard and Han, 2020). The task has also been reformulated to evaluate implicit biases in LLMs, with resulting benchmarks such as WinoGender and WinoBias (Rudinger et al., 2018; Zhao et al., 2018).

By 2019, large pre-trained transformer models were reported to achieve over 90% accuracy on WSC (Kocijan et al., 2023). Whereas the initial hypothesis was that systems would need common sense to solve the WSC, there is no proof that this

[1] A classic example is 'The trophy didn't fit in the brown suitcase because *it*'s too [small/big].'

is the case. Indeed, LLMs can rely on superficial pattern recognition and data memorization to solve the task, leading to the conclusion that these results are not indicative of common sense acquisition (Sakaguchi et al., 2021). Furthermore, questions are raised about the quality and implicit biases of WSC, such as lax evaluation, annotation artifacts, and knowledge leakage from training data (Kocijan et al., 2023; Elazar et al., 2021). Trichelair et al. (2018a) also show that the 'google-proof' condition, that stipulates that it should not be solvable via statistics learned from large corpora associating one option to other components in the sentence, is not true for all instances in WSC. In an effort to address these limitations, adversarial variants of the WSC are presented. For instance, Han et al. (2024) adapt the options so that they are more associated with the wrong answer, and Trichelair et al. (2018b) switch the position of the options in the texts where possible. Both report a decrease in model performance. Abdou et al. (2020) show that models are not robust against linguistic perturbations such as changes in tense, gender, or synonym substitution in WSC sentences. Additionally, the WinoGrande benchmark is introduced (Sakaguchi et al., 2021). This benchmark is of a much larger scale (44K instances compared to the 273 in WSC), and employs an algorithm to reduce biases that machines can exploit to solve the task.

2.3 Common sense knowledge categorization

To the best of our knowledge, research on WinoGrande discusses model results holistically (on the entire test or validation set), but we suggest connecting this to common sense knowledge categorization as an effective error analysis of the task. By measuring the performance per category, we can isolate reasoning deficiencies that are obscured by aggregated metrics. There has been much effort on defining semantic categories to structure knowledge for NLP. Schank (1972) describes four main categories in their conceptual dependency theory: objects, actions, location, and time. Jackendoff (1992) suggests common primitives such as entity, property, number, location, state, event, and activity. Other work only uses two high-level categories, such as social and physical (Sap et al., 2020). Yet others define semantic categories within one common sense category; for instance, Wang et al. (2021) include feelings and characteristics, interaction, and norms as sub-categories of social common sense. Additionally, different common sense categories are sometimes evaluated by specific independent benchmarks (e.g., spatial (Xu et al., 2017; Liu et al., 2022), temporal (Zhou et al., 2019; Aroca-Ouellette et al., 2021; Hosokawa et al., 2024; Qin et al., 2021), numerical (Lin et al., 2020), physical (Bisk et al., 2020; Storks et al., 2021), social (Sap et al., 2019), etc.). This can be problematic when comparing one model's ability to reason over various common sense categories, since each benchmark can have a different answer format (i.e., multiple choice, binary choice, open-ended) and structure. Other benchmarks that are more general, do not provide common sense categorizations. Therefore, we annotate the WinoGrande benchmark (a general-purpose benchmark) with which common sense knowledge is relevant when making the decision (i.e., what knowledge is needed when making the bridging inference). In a similar effort, Zhang et al. (2020) proposed 6 common sense categories to evaluate performance on the WSC: property, object, eventuality, spatial, quantity, and others.

2.4 Benchmark memorization and contamination

Xu et al. (2024) define benchmark data contamination (BDC) as LLM exposure to benchmark data during training, leading to inflated evaluation results. They outline contamination severities ranging from exposure to meta information about the benchmark or the task, to the benchmark data itself with labels. One main detection technique is n-gram overlap counting, as used by GPT-3 (Brown et al., 2020) (13-gram) and GPT-4 (Achiam et al., 2023) (40-gram). However, it requires full pretraining data access and can miss rephrasing (Yang et al., 2023). Additionally, Wang et al. (2025) find that factual or lexical tasks are particularly susceptible to memorization, while Carlini et al. (2023) demonstrate that memorization increases with model size, data frequency, and sufficient context.

Since 2012, many WSC sentences have appeared in web text used to train LLMs (Elazar et al., 2021). RedPajama (Weber et al., 2024) contains 58.2% of WSC instances, while other datasets like The Pile (Gao et al., 2020a) contain around 30% (Elazar et al., 2024). Such contamination inflates accuracy scores: Emami et al. (2020) show significant accuracy drops when contamination is minimized.

In contrast, WinoGrande's creators mitigated contamination by keeping the test labels private. Re-

garding the validation set, only 1.1% of this set appears online or in CommonCrawl between December 2020 and October 2023 (Li et al., 2024), and the authors of GPT-4 self-report approximately 0.9% contamination in a sample of 1,000 instances (Achiam et al., 2023). Elazar et al. (2024) demonstrate that large pretraining corpora for LLMs did likely not encounter the WinoGrande test set, but they do not examine contamination of the validation set in these pretraining corpora. Thus, the precise effect of the contamination of the WinoGrande validation set is unknown, but for other benchmark data, it was previously shown that the effect of even minimal contamination can be underestimated (Singh et al., 2024).

3 Methodology

3.1 Data

In this study, we apply models on the WinoGrande benchmark, which was originally presented in 2019 as an adversarial dataset to the Winograd Schema Challenge (WSC) (Sakaguchi et al., 2021). Contrary to WSC, in which the sentence includes a pronoun that must be disambiguated given two candidate antecedents, the WinoGrande benchmark evolved to a fill-in-the-blank token ('_') problem (see Figure 2). Additionally, every instance does not necessarily have a twin sentence. The original paper reports human accuracy of 94%, and model accuracy of 79.1%, which is considerably lower than on WSC (over 90%). The labels of the test set are not publicly available, which has led to research reporting on the validation set (see e.g., Li et al. (2021); Sun and Emami (2024); Elazar et al. (2021)). For that reason, we will also report on the validation set. This split consists of 1,267 instances, with a balanced label distribution. The WinoGrande benchmark is also frequently used to evaluate new LLMs[2]. Recent evaluations include Gemma 2 27B at 83.7% (Team et al., 2024), LlaMA 2 (zero-shot) models ranging from 69.2% (7B) to 80.2% (70B) (Touvron et al., 2023b), GPT-4 (few-shot) achieving 87.5% (Achiam et al., 2023), and Pythia 12B (five-shot) scoring 66.6% (Biderman et al., 2023).

[2]It is unclear whether they report on the validation or test set. We assume these models use prompting techniques instead of partial evaluation (infra), but the reports are unclear on that aspect.

3.2 Models

We focus on recent open-source Large Language Models. Since model size is a known factor in model performance, we select model families that have different sizes available. Specifically, we select Gemma 2 (2B, 9B, and 27B) (Team et al., 2024); LlaMA 2 (7B, 13B, and 70B) (Touvron et al., 2023b), and OPT (1.3B, 6.7B, 13B, and 66B) (Zhang et al., 2022) to evaluate the effect of paraphrasing WinoGrande, and for the evaluation per common sense category. Further, to evaluate benchmark memorization, we include two other models because their pre-training data is publicly available, contrary to the previously mentioned models: Pythia (1B, 1.8B, 6.9B, and 12B) (Biderman et al., 2023) and LlaMA 1 (7B, 13B, 30B, and 65B) (Touvron et al., 2023a).

To evaluate model performance, we use partial evaluation, which calculates the summed log-likelihood for the tokens after each option in the text, selecting the one with the highest score (Trinh and Le, 2018). We choose this metric for three reasons:

1. It is the evaluation metric used in the Language Model Evaluation Harness (Gao et al., 2024), which is the base of the Huggingface Open-LLM Leaderboard[3];
2. Preliminary experiments show that it works better than prompting, and Trinh and Le (2018) show that it works better than full evaluation;
3. It is easily generalizable to different open-source models.

3.3 Paraphrased corpus

To test the robustness of model performance on WinoGrande, we create WinoWhat: a parallel corpus in which we paraphrase the sentences. We follow the fill-in-the-blank convention of WinoGrande because of the naturalness of generation in autoregressive models and known LLM biases for multiple-choice answering (such as in WSC) (Balepur et al., 2024; Cho et al., 2025). Our corpus solves the main limitation of the partial evaluation metric: it relies on the plausibility of the subsequent sequence, rather than directly measuring a model's intrinsic token preference. This can conflate the model's understanding of the antecedent with grammatical or natural continuations. In contrast, in our paraphrased corpus, we position the target token at the end of the sequence, ensuring

[3]WinoGrande was included in the V1 of the leaderboard: https://huggingface.co/docs/leaderboards/en/open_llm_leaderboard/archive

that the decision is based solely on the provided context. This allows for a more transparent evaluation of the model's ability to capture coreference and fill-in-the-blank cues. Contrary to the original partial evaluation that measures the summed log-likelihood on the tokens following the '_' token, our method calculates it on the tokens of the options. An example is given in Figure 2.

We prompt 5 SOTA LLMs (i.e., GPT-4o (Hurst et al., 2024), OpenAI o1-preview (Jaech et al., 2024), Gemini 2.0 Flash Thinking Experimental (DeepMind, 2024), Deepseek R1 (Guo et al., 2025), and LlaMA 3.2 90B Vision (Meta, 2024)) to generate a paraphrased sentence given an input sentence, in which the '_' token is at the end of the sentence. The generated options were manually checked, and the best option was selected for each sentence. However, in many cases ($n = 433$), manual adjustments were still needed. The prompt for this task, and the distribution of which model's output is used, can be found in Appendix A. In this stage, we also evaluate the validity of the sentences in the WinoGrande validation set. We notice that not all instances meet the requirements of WSC (e.g., not 'google-proof', grammatical errors, etc.), which we remove in our paraphrased dataset. In total, we find 82 such cases.[4]

Further, three of the authors annotate a sample of 100 paraphrased instances based on the following criteria: (1) Is the new sentence grammatical?; (2) Is the fill-in-the-blank token at the end of the sentence?
85% of the texts are rated by all annotators as acceptable, 97% by at least two annotators. Given that the annotations are highly skewed (the majority of the ratings is 'acceptable'), we calculated Gwet's AC_1 for the inter-annotator agreement: 0.88 indicates a high agreement (Gwet, 2001).

3.4 Common sense knowledge categorization

We categorized the coreference resolution instances according to the common sense knowledge type that is necessary to make the bridging inference. This categorization can function as data for an error analysis to detect what knowledge types are easier or harder for LLMs to solve. Similarly to Zhang et al. (2020), we select categories that have a broad coverage and are clearly distinguished from each other. We examine which categories are identified in existing benchmarks that evaluate common sense reasoning in NLP[5], which leads to five categories: physical, social, numerical, temporal, and spatial[6]. We use LLMs to categorize the validation set. To identify the relevant common sense type, we prompt GPT-4o-mini to generate reasoning steps to solve the task. We then provide the input text and the generated reasoning steps to GPT-4o, which assigns one or more common sense categories to each instance. The prompts for these tasks are available in Appendix B. Annotation reliability is assessed by one author manually labeling 100 instances and comparing them with GPT-4o's labels, yielding a Kappa score of 0.64, which is a substantial agreement[7] (Cohen, 1960). Across all samples and labels, the annotator and GPT-4o agree in 83% of the cases. When applying our method on the entire validation set, we note a class imbalance; the physical and social categories are considerably larger than the other three, see Figure 3.

4 Results

4.1 Paraphrased corpus

We report on the models' performance on WinoWhat. This allows us to compare the performance on the original texts to the paraphrased texts. If models truly generalize on the Winograd schemas, the performance should remain consistent; after all, the same information is conveyed, in the same task setup, only paraphrased. Additionally, we report on the performance per common sense category presented in Section 2.3. In Table 1 and Table 2, subcolumn 'orig' refers to the original texts in WinoGrande, 'transf' to the paraphrased texts.
Considering the result on the WinoGrande validation set, we see that larger models generally perform better than their smaller variants, with LLaMA 2 70B performing the best. The error analysis comparing the performance of the same model across common sense categories shows that there is no one category that is impossible to be learned by a model, but there are fluctuations. Interestingly, we see that the category with the best results varies

[4] There are an additional 22 instances for which one annotator was not convinced of the quality. These instances were left out in the experiments, but for completeness are added in the released dataset.

[5] e.g., see https://cs.nyu.edu/~davise/Benchmarks/Text.html

[6] Originally, we included causal as label, but removed this category: all instances in WinoGrande had this label, which was also noted by Zhang et al. (2020).

[7] The kappa scores per category: physical 0.63; social 0.68; numerical 0.58; temporal 0.72; spatial 0.59.

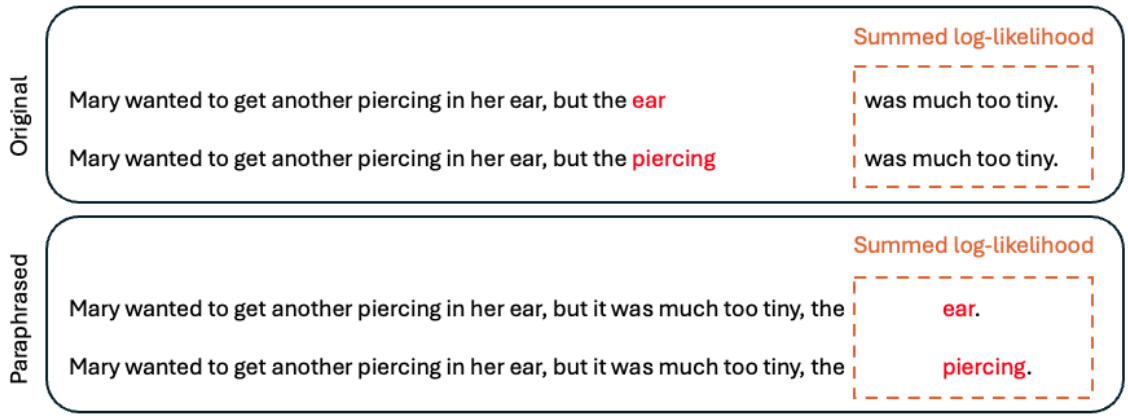


Figure 2: An illustration of the paraphrasing and evaluation method. The option that is filled in the '_'-token is in red. In the original example, the summed log-likelihood is calculated on the tokens after the option. In the paraphrased example, the option is at the end of the sentence, and the summed log-likelihood is calculated on the tokens inside the option.

	LlaMA 2						**Gemma 2**					
	7B		13B		70B		2B		9B		27B	
	orig	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**
TOTAL	0.69	0.58	0.72	0.62	0.78	0.70	0.68	0.59	0.74	0.68	0.66	0.56
Physical	0.71	0.61	0.73	0.63	0.77	0.73	0.71	0.60	0.74	0.68	0.74	0.59
Social	0.68	0.56	0.72	0.61	0.79	0.68	0.68	0.57	0.73	0.67	0.60	0.54
Numerical	0.69	0.53	0.70	0.61	0.79	0.69	0.63	0.62	0.75	0.62	0.69	0.51
Spatial	0.71	0.61	0.76	0.65	0.75	0.70	0.70	0.61	0.78	0.69	0.78	0.62
Temporal	0.76	0.67	0.70	0.69	0.79	0.74	0.65	0.62	0.74	0.71	0.67	0.54

Table 1: LlaMA 2 and Gemma 2 results on WinoGrande validation. The 'orig' columns report the results on the original instances, the 'transf' columns on the paraphrased instances.

	OPT 1.3B		OPT 6.7B		OPT 13B		OPT 66B	
	orig	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**
TOTAL	0.60	0.53	0.66	0.54	0.65	0.56	0.69	0.58
Physical	0.62	0.57	0.72	0.57	0.67	0.60	0.73	0.61
Social	0.59	0.50	0.63	0.50	0.65	0.52	0.66	0.55
Numerical	0.57	0.49	0.62	0.58	0.63	0.54	0.68	0.57
Spatial	0.56	0.61	0.65	0.61	0.63	0.61	0.67	0.61
Temporal	0.50	0.55	0.57	0.58	0.61	0.53	0.66	0.57

Table 2: OPT results on WinoGrande validation. The 'orig' columns report the results on the original instances, the 'transf' columns on the paraphrased instances.

across model families: for LlaMA 2, there is no category that is consistently easier, while for Gemma 2 spatial is best, and for OPT physical. Temporal is consistently the worst category for OPT.

However, when comparing the original to the paraphrased task, we conclude that all models perform worse on the paraphrased corpus, and there is no common sense category that is robust against this transformation.

Our results challenge the assumption that LLMs apply reasoning when solving the WinoGrande task, suggesting they instead rely on dataset artifacts and/or memorization. While Sakaguchi et al. (2021) implemented an algorithm to automatically reduce machine-exploitable bias in their corpus, our results demonstrate that this might not be effective anymore in the LLM era.

We publicly release WinoWhat, consisting of the original WinoGrande validation set with the paraphrased counterparts and common sense catego-

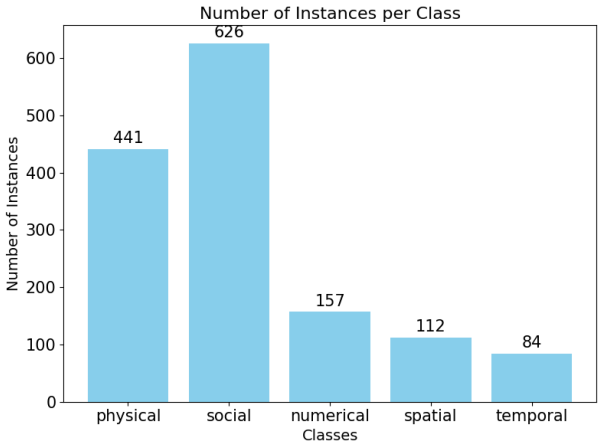


Figure 3: Data distribution across common sense categories on the WinoGrande validation set.

rizations[8].

4.2 Memorization

Model	WG val		WG test		WSC	
	orig	transf	orig	transf	orig	transf
LlaMA 2 7B	0.69	0.58	0.74	0.54	0.86	0.54
LlaMA 2 13B	0.72	0.62	0.73	0.65	0.83	0.63
LlaMA 2 70B	0.78	0.70	0.79	0.70	0.88	0.66
Gemma 2 2B	0.68	0.59	0.73	0.61	0.83	0.64
Gemma 2 9B	0.74	0.68	0.73	0.64	0.86	0.58
Gemma 2 27B	0.66	0.56	0.58	0.57	0.76	0.51
OPT 1.3B	0.60	0.53	0.58	0.50	0.72	0.54
OPT 6.7B	0.66	0.54	0.52	0.56	0.82	0.56
OPT 13B	0.65	0.56	0.68	0.56	0.81	0.56
OPT 66B	0.69	0.58	0.71	0.52	0.82	0.58

Table 3: Accuracy on WinoGrande (WG) validation, WG test, and WSC for LLaMA 2, Gemma 2, and OPT.

Model	WG val		WG test		WSC	
	orig	transf	orig	transf	orig	transf
LlaMA 1 7B	0.70	0.58	0.74	0.59	0.85	0.61
LlaMA 1 13B	0.72	0.60	0.75	0.64	0.88	0.66
LlaMA 1 30B	0.76	0.64	0.74	0.62	0.92	0.62
LlaMA 1 65B	0.77	0.67	0.79	0.69	0.91	0.68
Pythia 1B	0.54	0.53	0.57	0.54	0.71	0.50
Pythia 2.8B	0.60	0.52	0.59	0.53	0.76	0.55
Pythia 6.9B	0.61	0.52	0.58	0.56	0.77	0.52
Pythia 12B	0.63	0.52	0.61	0.60	0.79	0.49

Table 4: Accuracy on WinoGrande (WG) validation, WG test, and WSC for LLaMA 1 and Pythia.

Given the surprising drop in performance comparing WinoGrande to WinoWhat, we investigate further what could cause this. While Elazar et al. (2024) show that the test set of WinoGrande has probably not been seen by LLMs, this is not tested for the validation set. This is problematic, because research often reports on this split because of the absence of the test labels. Therefore, it is crucial to verify how many instances of the WinoGrande validation set have been included in datasets used to pre-train LLMs. Specifically, we count how many instances appear entirely in the pre-training corpora.

Since the pre-training data for Gemma 2, LlaMA 2, and OPT models remains either undisclosed or inaccessible, we examine two LLMs with publicly available pre-training data: LlaMA 1 and Pythia, whose results are presented in Table 4. These models were trained on RedPajama v1 (Computer, 2023) and The Pile's training set (Gao et al., 2020b) respectively.[9]

While we found that The Pile contains no contaminated instances, an interesting pattern emerges: as model size of Pythia increases, the performance gap between WinoGrande and WinoWhat widens, with WinoWhat accuracy remaining stable while WinoGrande scores improve (see column 'WG val' in Table 4).

An analysis of RedPajama v1 reveals 22 contaminated instances (1.7% of the dataset), each appearing once and sourced from academic papers. To investigate potential memorization effects, we conduct a one-sided Mann-Whitney U test between performance on contaminated and non-contaminated instances across LlaMA 1 models (7B, 13B, 30B, and 65B). The results (see Table 6 in Appendix C), with p-values ranging from 0.054 to 0.267, show no significant evidence that LLaMA 1 models give preferential treatment to previously seen WinoGrande instances. However, similarly to Pythia, LlaMA 1 displays a consistent accuracy gap between WinoGrande and WinoWhat. Since this pattern is observed in all other models as well (Table 1 and Table 2), it suggests that factors beyond simple memorization may be driving these performance differences.

To verify the role of contamination in later and more modern models with unknown pre-training data, we create two test-suites. Specifically, we take a sample ($n = 100$) from the WSC dataset (of which we can assume that a substantial part has been memorized by LLMs (Elazar et al., 2024)), and paraphrase those; and we take a sample ($n =$

[8]The full dataset is available on Zenodo (Gevers and De Marez, 2025).

[9]Details about our method to check memorization can be found in Appendix C.

100) from the test set of WinoGrande (of which we can assume that it has not been memorized by LLMs due to its private labels), which we label manually and paraphrase as well.[10] We hypothesize that LLMs perform well on datasets that are polluted, but less so on unseen datasets. Therefore, we expect models to perform well on WSC, but below par on WSC paraphrased and WinoGrande test (both original and paraphrased). We summarize the results in Table 3. As expected, all models perform best on the original WSC benchmark. Paraphrasing almost always causes a drop in performance, regardless of the original source. The difference is biggest for the WSC benchmark, which is in line with our hypothesis given the pollution by this benchmark in LLMs' training data. We still see a drop in performance for the WinoGrande test set, which is not included in the LLM training data, when comparing the original sentences to the paraphrased ones. Together with our findings on Pythia and LLaMA 1, this indicates that there are other factors causing models to struggle with the paraphrased benchmark. We hypothesize that our evaluation metric better captures the model's performance on coreference resolution compared to the original partial evaluation (see Figure 2), which could explain the drop in performance. Additionally, for larger and recent models, even though benchmark instances might not appear directly in the pre-training data, this does not exclude the possibility that it has been used during RLHF or instruction tuning, thereby compromising the validity of their performance on WinoGrande.

5 Conclusion

In this study, we take a closer look at how Winograd schema challenges can be used to evaluate common sense reasoning in LLMs. For this purpose, we focus on WinoGrande, a large adversarial benchmark created in 2019, frequently used to evaluate common sense in new LLMs. We select different generative model families, comparing models of the same family of different sizes. Specifically, we focus on Gemma 2, LlaMA 2, and OPT. To evaluate the models, we employ the partial evaluation metric. To address the limitations of the partial evaluation metric as outlined in Section 3.3, we create a parallel corpus to the WinoGrande validation set in which we paraphrase each text so the fill-in-the-blank token is at the end of the sentence (**RQ1**). In addition, we propose a new method to inspect performance on various common sense knowledge categories within the same task (**RQ2**). We select five categories: physical, social, numerical, spatial, and temporal. This approach can offer an in-depth error analysis, that sheds light on what types of knowledge are more challenging for LLMs. We publicly release WinoWhat, the parallel corpus to the WinoGrande validation set including the paraphrased sentences and the common sense categorization. Our results show that while models perform well on the original WinoGrande validation set, they all perform worse on the paraphrased corpus, and all common sense categories are affected negatively. This questions the assumption that models apply reasoning, leaving the possibility for dataset artifacts or benchmark memorization.

To verify how much data memorization has an effect on the models' performance on the WinoGrande validation set (**RQ3**), we test whether instances that occur in pre-training data score significantly higher than instances that don't. We observe that the memorization of the validation set is minimal. Interestingly, we see that most contaminated instances come from academic publications citing examples from the benchmark. This again calls attention to the scraping methods to create large-scale pre-training data. Because the pre-training data of later models is unknown, we create two small ($n = 100$) test-suites: one of which has been shown to be included in LLM training sets (i.e., the WSC benchmark) and one that is not seen by LLMs (i.e., the WinoGrande test set). We find that all models perform best on the WSC dataset, and paraphrasing causes a drop in performance. Since this is also the case for the WinoGrande test set, we conclude that there are other factors beside memorization that cause models to fail on the paraphrased task. Similarly to conclusions about the original Winograd Schema Challenge, this implies that we are again overestimating LLMs reasoning capabilities when using WinoGrande. Our new paraphrased corpus can be used to verify model generalization on the WinoGrande validation set.

In further research, we plan to inspect the information that is used by models to solve the task per common sense category using mechanistic interpretability: do models use similar information for each category? Do they rely on spurious

[10] To respect the private nature of the WinoGrande test set, we do not release our annotations of this subset.

correlations, and if so, which ones? Mechanistic interpretability could help us identify a causal connection between the direct and the indirect object, giving insights on why models fail. Since data memorization does not seem to cause the drop in performance comparing the original to the paraphrased instances, we suggest to identify dataset artifacts that could be at the root of this. For instance, as previously done on WSC, do linguistic perturbations affect model performance?

Acknowledgments

This research was made possible with a grant from the Fonds Wetenschappelijk Onderzoek (FWO) project 42/FA030100/9770, and funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

Limitations

While, to the best of our knowledge, this is the first time the WinoGrande validation set has been annotated for common sense knowledge categories, this approach has possible shortcomings. First, the agreement between a human annotator and the labeling by GPT-4o shows a substantial agreement, but there will be cases with incorrect labeling. Therefore, we talk about aggregated results across categories in this study, since we're interested in trends, but for even more fine-grained interpretations this categorization should possibly be corrected manually.

Further, as is unfortunately still a trend in NLP-research, this dataset is in English, excluding lower-resource languages. Further research could translate our benchmark to other languages.

During the process of paraphrasing the original instances, we applied a strict quality check, which excluded 82 instances from the original dataset. While we believe this improves the quality of the resulted paraphrased dataset, this means we cannot make a perfectly aligned comparison to the original dataset.

Since we wanted to mitigate shortcomings of the partial evaluation metric, we paraphrased WinoGrande so the fill-in-the-blank token appears at the end of the sentence. However, the constraint of putting this token at the end of the sentence caused a higher number of cleft-constructions in the corpus. A high inter-annotator agreement shows that the created paraphrases are grammatically correct and qualitative, but in some cases the paraphrased output is less natural than the original. However, even though there might be an 'unnaturalness' about some of the paraphrased instances, this does not change the task (i.e., finding the correct antecedent), and a robust model should be able to overcome these superficial variations.

We argue that this setup is more natural for decoder-only models, and allows the partial evaluation metric to better capture model performance on coreference resolution tasks rather than measuring natural continuations of the sentence. However, by adapting the evaluation method so it calculates the summed log-likelihoods on the tokens in the option rather than on the tokens after the option, this obscures whether the difference in performance is a result of the paraphrasing, or of the evaluation method. To verify this, we aim to construct a third level, in which we paraphrase without the constraint of putting the '_'-token at the end of the sentence, allowing us to use the original partial evaluation method. This would indicate whether the drop in performance is caused by the paraphrasing itself, or by the evaluation metric. This would also alleviate the problem that some paraphrased sentences in WinoWhat are slightly less natural-sounding than the original ones.

Finally, our method of finding data contamination in pre-training data was on the data level only, not taking into account the semantic or information level (Xu et al., 2024). Methods such as ours relying on string matching methods might miss certain instances, such as rephrasings (Xu et al., 2024). Furthermore, such methods are only possible when access to pre-training corpora is public (Yang et al., 2023).

References

Mostafa Abdou, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott, and Anders Søgaard. 2020. The sensitivity of language models and humans to Winograd schema perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7590–7604, Online. Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pascal Amsili and Olga Seminck. 2017. A Google-proof collection of French Winograd schemas. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 24–29, Valencia, Spain. Association for Computational Linguistics.

Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. Prost: Physical reasoning of objects through space and time. *arXiv preprint arXiv:2106.03634*.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024. Artifacts or abduction: How do LLMs answer multiple-choice questions without the question? *Preprint*, arXiv:2402.12483.

Timothée Bernard and Ting Han. 2020. Mandarinograd: A Chinese collection of Winograd schemas. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 21–26, Marseille, France. European Language Resources Association.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

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

Jacob Browning and Yann LeCun. 2023. Language, common sense, and the winograd schema challenge. *Artificial Intelligence*, 325:104031.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Gyeongje Cho, Yeonkyoung So, and Jaejin Lee. 2025. ANPMI: Assessing the true comprehension capabilities of LLMs for multiple choice questions. *Preprint*, arXiv:2502.18798.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Google DeepMind. 2024. Gemini 2.0 flash thinking. 2024.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the winograd schema. *arXiv preprint arXiv:2104.08161*.

Ali Emami, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. An analysis of dataset overlap on Winograd-style tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5855–5865, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Denis Emelin and Rico Sennrich. 2021. Wino-x: Multilingual winograd schemas for commonsense reasoning and coreference resolution. In *2021 Conference on Empirical Methods in Natural Language Processing*, pages 8517–8532. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020a. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020b. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Ine Gevers and Victor De Marez. 2025. Winowhat: A parallel corpus of paraphrased winogrande sentences with common sense categorization.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Kilem Gwet. 2001. Handbook of inter-rater reliability: How to estimate the level of agreement between two or multiple raters. *Gaithersburg, MD: STATAXIS Publishing Company*.

Kaiqiao Han, Tianqing Fang, Zhaowei Wang, Yangqiu Song, and Mark Steedman. 2024. Concept-reversed winograd schema challenge: Evaluating and improving robust reasoning in large language models via abstraction. *arXiv preprint arXiv:2410.12040*.

Taishi Hosokawa, Adam Jatowt, and Kazunari Sugiyama. 2024. Temporal validity reassessment: commonsense reasoning about information obsoleteness. *Discover Computing*, 27(1):4.

Yufang Hou, Katja Markert, and Michael Strube. 2018. Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ray S Jackendoff. 1992. *Semantic structures*, volume 18. MIT press.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Hideo Kobayashi and Vincent Ng. 2020. Bridging resolution: A survey of the state of the art. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3708–3721.

Vid Kocijan, Ernest Davis, Thomas Lukasiewicz, Gary Marcus, and Leora Morgenstern. 2023. The defeat of the winograd schema challenge. *Artificial Intelligence*, 325:103971.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. *KR*, 2012:13th.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2021. A systematic investigation of commonsense knowledge in large language models. *arXiv preprint arXiv:2111.00607*.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 528–541, Miami, Florida, USA. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *arXiv preprint arXiv:2005.00683*.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*.

Ruicheng Liu, Rui Mao, Anh Tuan Luu, and Erik Cambria. 2023. A brief survey on recent advances in coreference resolution. *Artificial Intelligence Review*, 56(12):14439–14481.

Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. 2022. Things not written in text: Exploring spatial commonsense from visual signals. *arXiv preprint arXiv:2203.08075*.

John McCarthy. 1959. Programs with common sense.

Gabriela Melo, Vinicius Imaizumi, and Fábio Cozman. 2019. Winograd schemas in portuguese. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pages 787–798, Porto Alegre, RS, Brasil. SBC.

Meta. 2024. Meta llama3.2. https://www.llama.com/.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. Timedial: Temporal commonsense reasoning in dialog. *arXiv preprint arXiv:2106.04571*.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 814–824.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th annual meeting of the association for*

computational linguistics: Tutorial abstracts, pages 27–33.

Roger C. Schank. 1972. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):552–631.

Aaditya K. Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *Preprint*, arXiv:2411.03923.

Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding. *arXiv preprint arXiv:2109.04947*.

Jing Han Sun and Ali Emami. 2024. EvoGrad: A dynamic take on the Winograd schema challenge with human adversaries. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6701–6716, Torino, Italia. ELRA and ICCL.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Paul Trichelair, Ali Emami, Jackie Chi Kit Cheung, Adam Trischler, Kaheer Suleman, and Fernando Diaz. 2018a. On the evaluation of common-sense reasoning in natural language understanding. *arXiv preprint arXiv:1811.01778*, 20180.

Paul Trichelair, Ali Emami, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2018b. How reasonable are common-sense reasoning tasks: A case-study on the winograd schema challenge and swag. *arXiv preprint arXiv:1811.01778*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Gengyu Wang, Xiaochen Hou, Diyi Yang, Kathleen McKeown, and Jing Huang. 2021. Semantic categorization of social knowledge for commonsense question answering. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 79–85, Virtual. Association for Computational Linguistics.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2025. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy S Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 116462–116492. Curran Associates, Inc.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

Cheng Xu, Shuhao Guan, Derek Greene, and M. Tahar Kechadi. 2024. Benchmark data contamination of large language models: A survey. *CoRR*, abs/2406.04244.

Frank F Xu, Bill Yuchen Lin, and Kenny Q Zhu. 2017. Automatic extraction of commonsense locatednear knowledge. *arXiv preprint arXiv:1711.04204*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *CoRR*, abs/2311.04850.

Pardis Sadat Zahraei and Ali Emami. 2024. WSC+: Enhancing the Winograd schema challenge using tree-of-experts. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1671, St. Julian's, Malta. Association for Computational Linguistics.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. WinoWhy: A deep diagnosis of essential commonsense knowledge for answering Winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference*

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. " going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. *arXiv preprint arXiv:1909.03065*.

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

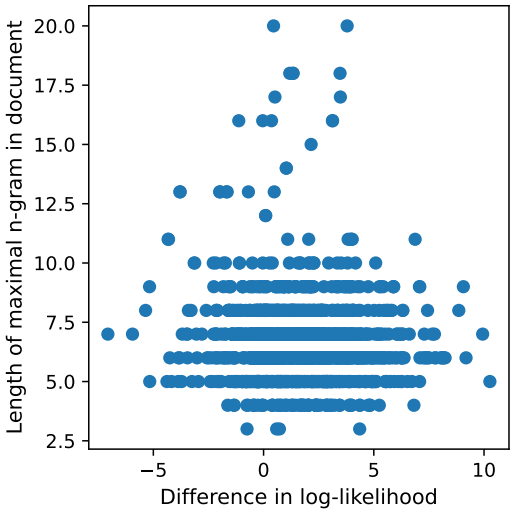
We conducted two visual correlation analyses for all instances in the WinoGrande validation set against the logit difference between ground truth correct and incorrect answers. The first analysis examined the correlation with the length of the longest n-gram sequence that appears at least once in the 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

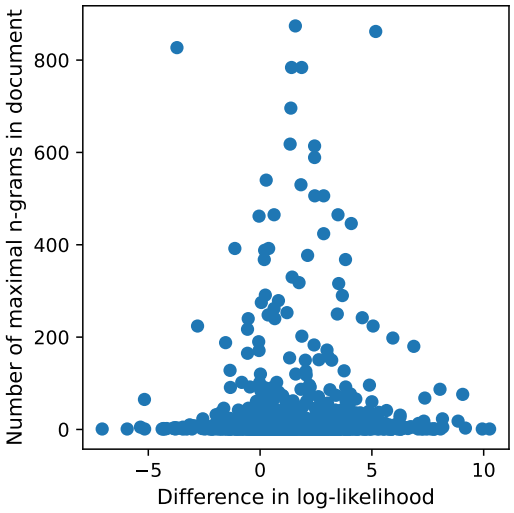
```
Given is a sentence. Below that is an n-
    gram that occurs in the sentence.
    Below that are some numbered
    documents, with the number between
    parentheses, that all contain the n-
    gram.

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.