Acquisition and Application of Novel Knowledge in Large Language Models

Anonymous ACL submission

Abstract

Recent advancements in large language models (LLMs) have demonstrated their impressive generative capabilities, primarily due to their extensive parameterization, which enables them to encode vast knowledge. However, effectively integrating new knowledge into LLMs remains a major challenge. Current research typically first constructs novel knowledge datasets and then injects this knowledge into LLMs through various techniques. However, existing methods for constructing new datasets either rely on timestamps, which lack rigor, or use simple templates for synthesis, which are simplistic and do not accurately reflect real-world. To address this issue, we propose a novel knowledge dataset construction approach that simulates biological evolution using knowledge graphs to generate synthetic entities with diverse attributes, resulting in a dataset **NovelHuman**. We then evaluate existing training strategies and knowledge augmentation methods on NovelHuman. A systematic position-based analysis reveals that the intra-sentence position of knowledge significantly affects the acquisition of knowledge. Therefore, we introduce an intra-sentence permutation to enhance knowledge acquisition. Furthermore, given that potential conflicts exist between autoregressive (AR) training objectives and permutation-based learning, we propose **PermAR**, a permutation-based language modeling framework for AR models. PermAR seamlessly integrates with mainstream AR architectures, endowing them with bidirectional knowledge acquisition capabilities. Extensive experiments and ablation studies demonstrate the superiority of PermAR, outperforming knowledge augmentation methods by 3.3%-38%.

1 Introduction

Recently, LLMs (OpenAI, 2023; Touvron et al., 2023) have gained widespread attention for their training on massive corpora, acquisition of vast factual knowledge, demonstrating remarkable capabilities in generating high-quality text (Gatt and Krahmer, 2018), in-context learning (Brown et al., 2020), and following complex instructions (Ouyang et al., 2022). However, the parametric knowledge of LLMs is constrained to their pre-training corpus, which predominantly comes from public resources like Wikipedia (Lemmerich et al., 2019), Github (GitHub, 2025), and CommonCrawl (Raffel et al., 2020), covering only a specific period. This limitation prevents LLMs from accessing other information beyond their training corpus. Consequently, efficiently integrating continuously updated knowledge into LLMs has emerged as a crucial task (Jiang et al., 2024c; Saito et al., 2024; Allen-Zhu and Li; Shi et al., 2024).

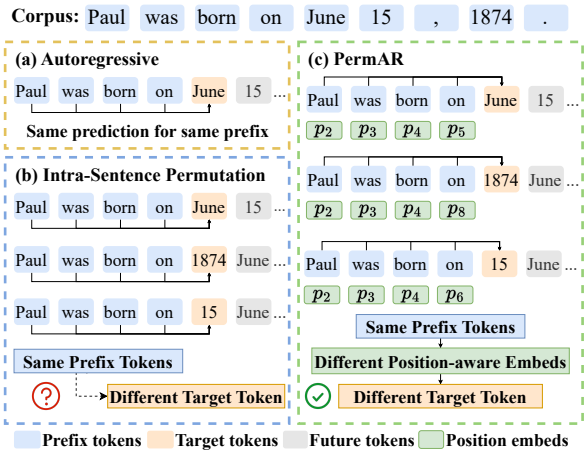

Figure 1: Illustration of the same new knowledge under different training methods: (a) AR models predict left-to-right. (b) Intra-sentence permutation with AR objectives cannot tackle different ground truths for the same prefix. (c) The PermAR framework adds position-aware embeddings for permutations, enabling different ground truth predictions.

Although recent works have attempted to construct novel knowledge datasets that LLMs have not encountered during pre-training and proposed various training strategies to inject unseen knowl-

edge, Jiang et al. found that LLMs can correctly answer nearly 10% of questions about knowledge that appeared on Wikipedia after their pre-training, even without external information. This finding underscores a key limitation in the construction of novel knowledge datasets: **selecting new knowledge based on timestamps is inherently unreliable.** First, there is no guarantee that the collected data did not appear in the LLMs' pre-training corpus. Knowledge on the internet is widely distributed, and the accessible information may simply be a post-processed or organized version of existing content, *i.e.*, this novel knowledge could have been collected in pre-training (Tirumala et al., 2024). This suggests that such datasets may primarily reactivate a model's internal memory rather than facilitate genuine learning or application of novel knowledge. Second, the training data used by different LLMs varies significantly, with inconsistent cut-off dates, thereby requiring substantial human effort to collect novel knowledge. Additionally, existing work (Allen-Zhu and Li) generates synthetic novel knowledge using templated methods, which result in the synthetic dataset being overly simplistic and failing to reflect real-world complexity.

To tackle the above challenges, inspired by the process of biological evolution producing new species, we propose a novel knowledge construction method involving inheritance, mutation, and expansion operations based on existing large-scale knowledge graphs (KGs) (Dong et al., 2014), aiming to generate totally novel knowledge for all LLMs. We synthesize a large-scale dataset with human subjects called **NovelHuman**, which contains 8,507 human subjects and 143K triples. For each subject, we convert its corresponding triples into linguistically fluent natural language text passages with the help of a more advanced LLM, and finally collect more than 144k questions.

Additionally, recent works (Allen-Zhu and Li; Saito et al., 2024) have revealed that LLMs' ability to master knowledge declines as the sentence containing the knowledge appears later in the document, *i.e.*, inter-sentence sensitivity. To address this issue, previous research has explored various knowledge augmentation strategies, such as sentence permutation and rewriting, which have been shown to enhance knowledge acquisition. However, this finding often relies on relatively simple knowledge datasets, where each sentence contains only a single piece of knowledge, limiting the ability to fully assess the generalizability of methods. To this end, we evaluate existing methods on NovelHuman. Beyond prior observations, we find that **LLMs exhibit not only inter-sentence sensitivity but also intra-sentence sensitivity**, meaning their ability to master knowledge varies even within different positions of the same sentence.

A straightforward alternative solution is to adapt existing inter-sentence permutation strategies into intra-sentence permutation. However, we find that intra-sentence permutation inherently conflicts with mainstream autoregressive (AR) modeling. Mainstream LLMs typically rely on an autoregressive (AR) architecture (Radford, 2018), which employs a strict left-to-right processing mechanism during both training and inference to facilitate information memorization and generation. In certain scenarios, simply intra-sentence permutation may not be sufficient to achieve the desired goal within standard AR, as shown in Figure 1. In the standard AR modeling, intra-sentence permutation leads to issue illustrated in Figure 1 (b), where given the same prefix tokens, model generates identical feature representations and predictions, even when the corresponding ground truth differs. Although XLNet (Yang et al., 2019) mitigates this problem by introducing a two-stream attention mechanism, it requires modifying the model architecture and retraining LLMs, making it highly costly.

To address this issue, we build upon the AR structure and propose the permutation language modeling framework for AR, **PermAR**, as shown in Figure 1 (c). PermAR introduces minimal modifications by learning a single position embedding for the next token's original position, enabling accurate prediction of various permutations, even when the prefix remains unchanged. Meanwhile, PermAR maximizes the expected likelihood over all possible inter-sentence and intra-sentence permutations, facilitating bidirectional contextual learning within the AR transformer while enhancing the model's ability to comprehend knowledge across different positions. Additionally, we propose a permutation annealing training strategy, which gradually restores LLMs from fully permutation factorizations to the original natural language sequence order, allowing LLMs to reconstruct fragmented knowledge points into a coherent knowledge representation. Extensive experiments and ablation studies demonstrate the superiority and adaptability of PermAR, outperforming knowledge augmentation methods by 3.3%-38%.

2 Building Dataset for Continual Knowledge Acquisition

Drawing inspiration from the organization of knowledge on Wikipedia, where each page introduces a subject whose distilled knowledge can be abstracted into KGs, the generation of new knowledge can be seen as the expansion of KGs, *i.e.* , the creation of new subjects. To obtain **reasonable** and **diverse** new subjects, we propose three operations that mimic biological evolution: **inheritance**, **mutation**, and **expansion**.

Suppose a specific KG $\mathcal{G} = \{\mathcal{G}_f, \mathcal{G}_o\}$ is given as the benchmark, where $\mathcal{G}_f = \{\mathcal{V}_f, \mathcal{R}_f, \mathcal{T}_f\}$ and $\mathcal{G}_o = \{\mathcal{V}_o, \mathcal{R}_o, \mathcal{T}_o\}$ stand for the instance and ontology sub-graph, while $\mathcal{V}_\sim$, $\mathcal{R}_\sim$, and $\mathcal{T}_\sim$ ($\sim \in \{f, o\}$) denote the set of entity, relation, triple, respectively. Moreover, $\mathcal{T}_f = \{(h_f, r_f, t_f) | h_f, t_t \in \mathcal{V}_f, r_f \in \mathcal{R}_f\}, \mathcal{T}_f = \{(h_o, r_o, t_o) | h_o, t_o \in \mathcal{V}_o, r_o \in \mathcal{R}_o\}$, where $h_\sim$, $r_\sim$, and $t_\sim$ ($\sim \in \{f, o\}$) denote the subject, predicate, object of a triple, respectively. New subjects h_f^{new}, can be categorized into two types: (1) Novel at the instance level while the ontology remains unchanged. (2) Novel both at the instance and ontology levels. To construct a reasonable new subject, we focus solely on the first type, *i.e.*, $h_f^{new} \notin \mathcal{V}_f, h_o^{new} \in \mathcal{V}_o$.

To create a specific new subject within a given ontology, it is necessary to establish connections with existing subjects (**inheritance**). This involves linking the new subject to the existing ontology, ensuring it inherits certain fundamental characteristics. To distinguish this new subject from its predecessors, it is essential to endow it with unique attribute values (**mutation**). Finally, we expand the existing attribute values of the current new subject to achieve diversity (**expansion**).

Inheritance Specifically, we begin by randomly selecting an ontology $h_o^I \in \mathcal{V}_o$, to which the new subject will belong. To mimic sexual reproduction in biological evolution, we randomly select two existing, distinct subjects as the dad h_f^D and the mom h_f^M from $\{h_f \mid h_f \in \mathcal{V}_f \wedge h_o = h_o^I\}$. The new subject h_f^{new} inherits all attributes from h_f^D and h_f^M, and merges the same relation. For instance, in the case of humans, parents have a birth date, and we consolidate these dates, allowing the new subject to have two objects for the birth date.

Mutation Next, we introduce mutation to predetermine the important attributes r_f^I for the new subject, such as the birth date for humans, by selecting a random date between the birth dates of h_f^D and h_f^M.

Expansion Subsequently, to enhance the diversity of attributes for h_f^{new}, we introduce an anchor relation, $r_f^a \in \mathcal{R}_f$, based on prior knowledge. For example, for human entities, r_f^a could represent a relation such as professions. We first obtain two candidate expansion subject sets $n^* = \{h_f^{add} \mid h_f^{add} \in \mathcal{V}_f \text{ and } h_o^{add} = h_o^I \text{ and } (h_f^*, r_f^a, t) \in \mathcal{G}_f \text{ and } (h_f^{add}, r_f^a, t) \in \mathcal{G}_f\}$, where $* \in \{\mathrm{D}, \mathrm{M}\}$, corresponding to the dad subject h_f^D and mom subject h_f^M, respectively. Then, to minimize potential conflicts, it is essential to filter the candidate expansion subject sets by integrating prior knowledge and important relations of the new subject. For example, in the case of humans, a person cannot participate in events that occurred before the birth date. Therefore, we apply this logical constraint to filter two candidate expansion sets, obtaining $n_f^* = \mathrm{RULE}(\mathrm{n}^*)$, where RULE is the constraint function of important relations. Ultimately, the structured triple set of $\mathcal{T}_{h_f^{new}}$ corresponding to h_f^{new} can be obtained through the process of inheriting and mutating attributes from the refined candidate expansion subject sets. Given the rich and complex attributes of human entities, we synthesize a large number of novel human entities. The detailed construction process of these novel triples is presented in Appendix A.1.

Knowledge & general question generation The process of generating new entities and their attributes can be summarized as assigning them as many diverse attributes and values as possible, in a logically consistent manner. However, during the inheritance, mutation, and expansion, we identified triple conflict, *i.e.*, while we construct novel knowledge, the attribute values of the new entity intuitively do not align with the current logic of the real world. To address this, we employ more powerful LLMs, such as GPT-4, to perform consistency checks. Subsequently, to obtain natural language text corresponding to the triple set $\mathcal{T}_{h_f^{new}}$ of h_f^{new}, we harness the language generation capabilities of advanced LLMs (GPT-4). By providing a sophisticated prompt, it can generate text in the style of Wikipedia, effectively creating novel knowledge.

To further assess whether LLMs effectively learn constructed new knowledge, we employ a common question-answering (QA) for evaluation. Since the new knowledge is generated directly from triple

sets, we can also derive evaluation questions from these triples. Specifically, we explain the meaning of each relation in the triples to GPT-4 and instruct it to create various question templates. However, due to the inherent hallucination phenomenon of LLMs, not all triples are accurately reflected in the generated text. To mitigate this, we further filter the triple set to identify which triples are correctly represented in the generated text. These confirmed triples are then used to fill in the question templates, ensuring that the answers to these questions are explicitly present in the generated text. Appendix D includes all the prompt templates used for knowledge and question generation.

2.1 Dataset Summary

Through the above steps, we construct the NovelHuman dataset, which contains 8,507 new human entities covering 435 attributes, with each entity initially associated with an average of 20 attributes. After triple consistency checking and the knowledge generation phase, the average number of attributes per entity was refined to 16. For the generated knowledge, each piece of knowledge contains an average of 410 tokens[1], resulting in a total of 144,221 QA pairs, with questions averaging 15 tokens and corresponding answers averaging 4 tokens. Further detailed dataset statistics, along with the corresponding training and test set splits, can be found in Appendix A.2.

3 Preliminary Experiments

3.1 Background

We provide a brief overview of autoregressive (AR) modeling with a next-token prediction objective. Given a discrete token sequence $x = [x_1, x_2, \cdots, x_T]$, the goal of AR modeling is to maximize the likelihood of the sequence under a forward AR factorization (Kitouni et al., 2024). Specifically, the objective is to maximize the joint probability of predicting the current token x_t based on all preceding tokens $[x_1, x_2, \cdots, x_{t-1}]$:

$$\max_{\theta} \quad p_\theta(\mathbf{x}) = \prod_{t=1}^{T} p_\theta \left(x_t \mid x_1, x_2, \cdots, x_{t-1} \right) \quad (1)$$

where p_θ denotes a token distribution predictor with a model parameterized by θ. Currently, most mainstream LLMs follow the AR pre-training paradigm. Similarly, to align with the representations learned during pre-training, existing AR-based CPT also adheres to Equation 1.

3.2 Knowledge Acquisition Techniques

Building upon prior research (Allen-Zhu and Li; Jiang et al., 2024c; Saito et al., 2024), we evaluated prevalent continued pre-training, instruction-tuning and knowledge augmentation paradigms in NovelHuman based on Llama-2-7B and Llama-3-8B, including standard continued pre-training (CPT), continued pre-training with supervised fine-tuning (CP+SFT), continued pre-training with forgetting-resistant SFT (CP+SFT w/o F), mixed training involving both pre-training and SFT simultaneously (MT) (Allen-Zhu and Li), mixed training involving pre-training and SFT (prompt for loss computation) (Allen-Zhu and Li), human-like learning approaches for acquiring new knowledge (R&A), pre-instruction-tuning (PIT++) (Jiang et al., 2024c), Attn Drop, D-AR and inter-sentence permutation (Allen-Zhu and Li). More introduction and hyperparameters setting of these methods can be seen in Appendix B.

3.3 Evaluation Metrics

In the evaluation process, we follow settings of Jiang et al. 2024c, where LLMs are required to generate answers for given questions using greedy decoding. Given that our questions tend to yield short and precise answers, exact match (EM) is employed as the primary metric to assess whether the answers are completely identical to the ground truth (Kwiatkowski et al., 2019). Furthermore, considering that some answers may be order-independent, we also report the recall rate (R) to measure whether the ground truth appears within LLMs' generated responses. Additionally, ROUGE-L (R-L) is used to evaluate the longest common subsequence between the LLMs' outputs and the ground truth (Lin, 2004). During evaluation, for LLMs that have not been fine-tuned with instructions, five QA pairs are provided as in-context demonstrations that are used to ensure the output follows the specified format.

3.4 Experimental Results

As shown in Table 1, the relatively low knowledge QA performance of the original Llama-2 and Llama-3 (0% EM for all) indicates that almost all knowledge in the test set is not included in the original pre-training corpus. It can be seen that these methods are struggling to acquire novel

[1]No special instructions, the token statistics are based on the Llama3 tokenizer.

Model	Training Pattern	Text			Number			Date			All		
		EM	R	R-L	EM	R	R-L	EM	R	R-L	EM	R	R-L
Llama-2-7B	Original	0.0	1.8	0.9	0.0	0.0	0.9	0.0	2.1	0.9	0.0	1.9	0.9
	CPT	0.0	1.5	1.5	0.0	0.0	1.6	0.0	1.5	1.5	0.0	1.5	1.5
	CPT + SFT	20.7	22.9	26.7	4.6	4.6	4.6	32.5	54.3	63.3	21.2	26.6	29.2
	CPT + SFT(w/o F)	26.8	28.5	32.8	5.4	5.4	5.4	40.3	61.5	69.1	27.3	32.3	35.0
	MT	32.8	33.7	38.2	10.4	10.4	10.4	42.0	65.9	72.6	32.8	37.3	40.2
	MT(prompt)	**37.6**	**38.8**	**43.1**	11.1	11.1	11.1	42.8	66.9	73.5	37.1	**41.7**	**44.5**
	R&A	17.4	20.4	23.6	3.2	3.2	3.2	14.6	40.6	52.5	16.6	22.7	25.4
	Attn Drop	19.3	20.8	24.5	4.0	4.0	4.0	18.9	23.5	24.1	19.6	23.8	26.6
	D-AR	23.1	25.4	28.7	4.9	4.9	4.9	34.5	62.6	70.8	25.3	28.2	31.9
	PIT++	22.1	25.8	27.9	11.5	11.5	11.5	24.4	52.5	61.8	22.0	30.0	30.3
	InterSP	36.5	37.6	41.8	**15.8**	**15.8**	**15.8**	**52.8**	**71.6**	**77.1**	**37.3**	41.5	44.0
Llama-3-8B	Original	0.0	2.2	1.2	0.0	0.0	1.2	0.0	2.0	1.1	0.0	2.1	1.2
	CPT	0.0	2.5	1.5	0.0	0.0	1.5	0.0	2.3	1.3	0.0	2.5	1.5
	CPT + SFT	19.3	21.9	25.1	6.8	6.8	6.8	36.1	55.1	24.3	20.4	26.0	28.0
	CPT + SFT(w/o F)	24.8	25.6	30.2	7.5	7.5	7.5	38.2	57.7	25.8	22.8	27.9	32.4
	MT	28.9	30.9	33.7	29.8	29.8	29.8	58.8	58.6	83.2	31.5	31.9	37.8
	MT(prompt)	28.8	31.2	34.3	**30.6**	**30.6**	**30.6**	59.1	59.1	84.9	31.7	32.6	38.9
	R&A	22.3	26.9	27.4	23.7	23.7	23.7	50.1	63.4	75.2	25.4	30.8	36.7
	Attn Drop	15.4	18.3	23.2	6.3	6.3	6.3	35.7	50.8	23.9	18.4	22.5	26.8
	D-AR	20.6	23.5	26.7	7.3	7.3	7.3	37.2	56.1	24.9	21.5	26.3	30.0
	PIT++	21.2	24.7	26.3	21.0	21.0	21.0	50.7	68.4	74.8	24.0	30.4	30.7
	InterSP	**39.4**	**42.1**	**44.6**	28.3	28.3	28.3	**73.1**	**84.2**	**87.3**	**42.2**	**47.6**	**48.0**

Table 1: Preliminary experimental results. Bold numbers denote the best results. Underline numbers imply the second-best results.

knowledge even though the perplexity has been reduced to 1 during the continued pre-training phase. Compared to only CPT, the EM after SFT increased to 21.2%/20.4% (Llama-2/Llama-3), indicating the effectiveness of the standard paradigm and that LLMs have captured some new knowledge. Among the knowledge augmentation methods, InterSP achieved the best performance, improving EM by 0.2%/10.5% in the knowledge QA, highlighting the importance of considering the position of sentences within a passage. However, these methods are still far from enabling LLMs to truly master knowledge.

4 Intra-sentence Permutation and PermAR Framework

Given the failures of various mainstream CPT and augmentation paradigms in new knowledge scenarios, we aim to further enhance LLM's learning effectiveness of new knowledge. Drawn inspiration from (Golovneva et al., 2024; Guo et al., 2024), we conducted a position-based systematic analysis of CPT+SFT and InterSP to explore the relationship between the effectiveness of its learning by LLMs and the position of knowledge between and within sentences. Statistical results indicate **both the position within and between sentences have a significant influence on prediction accuracy.** Specifically, the average EM of the first three knowledge in the sentences that are positioned differently in passage of the test set is depicted in Figure 2 (a) and (c). It can be observed that for the same position within sentences, the earlier a sentence appears in the passage, the more easily LLMs can learn that knowledge. Additionally, it can be observed that after applying inter-sentence permutation enhances knowledge retention across different sentences. However, when observing Figure 2 (b), a distinct downstair-like pattern emerges within sentences meaning knowledge positioned earlier in a sentence is more easily retained, whereas knowledge appearing later remains significantly harder to grasp. Furthermore, Figure 2 (d) indicates that inter-sentence permutation alone is insufficient to bridge the accuracy gap between knowledge positioned earlier and later within the same sentence.

Therefore, to improve LLM's ability to perceive knowledge at different positions within a sentence, a straightforward approach is to apply intra-sentence permutation alongside inter-sentence permutation (InterSP+IntraSP). Inspired by XLNet (Yang et al., 2019), InterSP+IntraSP can be viewed as a subset of permutation language modeling. Formally, the objective of permutation language modeling can be formulated as follows:

$$\max_{\theta} \quad p_\theta(\mathbf{x}) = \mathbb{E}_{\tau \sim \mathcal{S}_T} \left[\prod_{t=1}^{T} p_\theta \left(x_{\tau_t} \mid x_{\tau_{<t}} \right) \right] \quad (2)$$

where $\mathcal{S}_T$ denotes the set of all possible permutations of the index sequence $[1, 2, \cdots, T]$, and τ

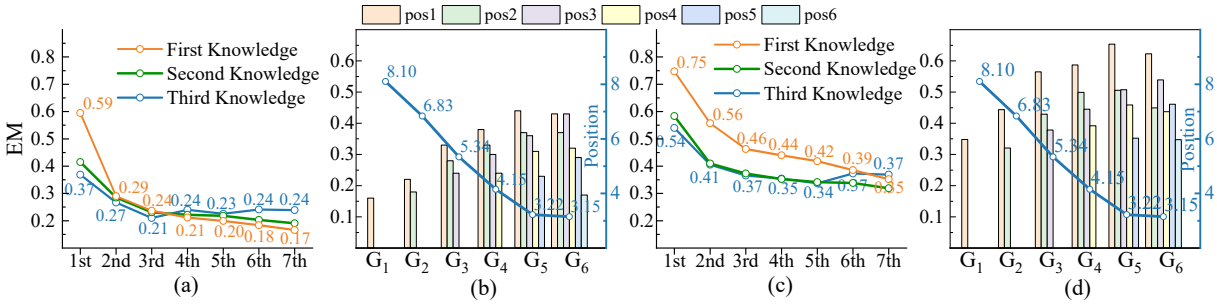

Figure 2: EM of CPT+SFT and InterSP on the test set using Llama-3-8B. (a) and (b) show the results for CPT+SFT, while (c) and (d) correspond to InterSP. (a) and (c) represent the EM scores of the i_{th} (i=1, 2, 3) knowledge from different sentences, with the horizontal axis indicating the sentence index. (b) and (d) categorize sentences based on the number of knowledge points they contain, where the horizontal axis represents the number of knowledge points per group, and the right vertical axis denotes the average position of sentences within the passage for each group.

represents a randomly sampled permutation from $\mathcal{S}_T$. The notation τ_t refers to the t-th element in the permuted sequence, and $\tau < t$ represents all preceding positions to τ_t. Since the model parameters θ are shared across all sampled factorization orders, each token x_t is exposed to every possible context and learns relationships with every other token $x_i, i \neq t$, during training. Moreover, this allows the model to effectively capture bidirectional context while preserving the integrity of the autoregressive formulation.

However, directly applying the permutation language modeling objective from Equation 2 to the standard AR model can fail in certain scenarios. For example, consider a sequence of length T corresponding to two different permutation orders $\tau_a = [1, 2, 3, \cdots, T-1, T]$ and $\tau_b = [1, 2, 3, \cdots, T, T-1]$. When predicting the second-to-last token, the standard AR model would return identical logits, despite the ground truth being different. The fundamental reason for this lies in the fact that the standard AR model, during next-token prediction, cannot incorporate the positional information of the target token.

Position-aware Instruction Embedding To enable a standard AR model to perceive differences in target positions when predicting the next token, we introduce a set of position-aware instruction embeddings to encode positional information of the next predicted token. Formally, we define a position-aware instruction embedding set $\mathbf{p} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_T]$. The position embedding of the next token is then directly added to the current token embedding, resulting in a position-aware token embedding $\mathbf{x}_\tau$:

$$\begin{aligned} \mathbf{x}_\tau^p &= \mathbf{x}_\tau + \mathbf{p}_\tau \qquad (3) \\ &= [\boldsymbol{x}_{\tau_1} + \boldsymbol{p}_{\tau_2}, \boldsymbol{x}_{\tau_2} + \boldsymbol{p}_{\tau_3}, \cdots, \boldsymbol{x}_{\tau_{T-1}} + \boldsymbol{p}_{\tau_T}, \boldsymbol{x}_{\tau_T}] \end{aligned}$$

where $\mathbf{x}_\tau$ and $\mathbf{p}_\tau$ represent the token embeddings of the original sequence after permutation τ, and the position-aware instruction embedding corresponding to the next position in the permuted sequence, respectively. It is noted that the last token in the permuted sequence does not receive a position-aware instruction embedding, as there is no subsequent token to predict.

For obtaining the position-aware instruction embedding, we first learn a single shared embedding for all positions, $\mathbf{e} \in \mathbb{R}^{1 \times dim}$, where dim is embedding dimension, and then rotate using the RoPE-1D (Su et al., 2024) combined with $\mathbf{e}$ and coordinates of next predicted token. The position-aware instruction embedding for token at position t is:

$$\boldsymbol{p}_t = \mathrm{RoPE}(\mathbf{e}, t) \qquad (4)$$

It is worth noting that position-aware instruction embeddings can be implemented using alternative approaches, such as learning a separate dense vector for each position or training a linear fusion layer for Equation 4 instead of using an additive operation. Detailed experiments and discussions on these alternatives are provided in Appendix C.4.

Permutation Annealing Strategy Although the proposed PermAR framework can be seamlessly integrated into existing AR models, enabling models to effectively learn new knowledge even when faced with small-scale knowledge, the number of possible permutations for a token sequence is exceedingly large. For instance, for a token sequence of length 1024, the number of possible permutations is 1024!, which would overwhelm the model and significantly reduce training efficiency. Meanwhile, based on the observation in Figure 2: standard AR models tend to better learn knowledge from earlier sentences of the passage, and even from earlier positions of the sentence.

Model	Method	Text			Number			Date			All		
		EM	R	R-L	EM	R	R-L	EM	R	R-L	EM	R	R-L
Llama-2-7B	InterSP	36.5	37.6	41.8	15.8	15.8	15.8	52.8	71.6	77.1	37.3	41.5	44.0
	IntraSP	55.8	56.5	59.5	36.5	36.5	36.5	67.5	80.6	84.3	56.1	59.2	60.8
	InterSP+IntraSP	71.9	73.7	75.2	45.4	45.4	45.4	**83.2**	91.9	93.4	72.0	75.8	75.8
	PermAR	**75.3**	**76.5**	**78.0**	**54.3**	**54.3**	**54.3**	**83.2**	**92.0**	**93.5**	**75.3**	**78.3**	**78.6**
Llama-3-8B	InterSP	39.4	42.1	44.6	28.3	28.3	28.3	73.1	84.2	87.3	42.2	47.6	48.0
	IntraSP	44.4	57.2	55.2	49.9	49.9	49.9	81.5	93.1	94.5	48.1	62.3	58.9
	InterSP+IntraSP	51.3	61.6	60.2	48.0	48.0	48.0	80.2	92.2	93.8	53.9	65.9	63.2
	PermAR	**61.7**	**69.2**	**68.0**	**57.4**	**57.4**	**57.4**	**84.4**	**93.8**	**95.1**	**63.7**	**72.7**	**70.4**

Table 2: Comparison of QA performance (%) between knowledge augmentation and PermAR.

Furthermore, we propose a permutation annealing strategy designed to help model reconstruct the fragmented knowledge learned during permutation training into more coherent and logically consistent knowledge. Specifically, we introduce a probability r to control the degree of permutation applied to the samples. When $r = 1$, it indicates that tokens in each sample are fully permuted, while $r = 0$ means that the token sequence remains in its original order. Formally, r can be modeled as follows:

$$r = \begin{cases} 1.0, & \text{if } epoch < start, \\ 0.0, & \text{if } epoch > end, \\ 1.0 - \frac{epoch - start}{end - start}, & \text{otherwise} \end{cases} \quad (5)$$

where $epoch$ denotes the current epoch during training, and $start$ and end represent the beginning and ending epochs of the permutation annealing strategy, respectively.

5 Experiments

5.1 Settings

Baselines. We further compare knowledge augmentation methods with PermAR, including intra-sentence permutation (IntraSP) and the combination of inter-sentence and intra-sentence permutation (InterSP+IntraSP). Details of knowledge augmentation are shown in Appendix B.3 and B.4.

5.2 Main Results

The experimental results are shown in Table 2, from which we can draw the following conclusions: **(1)** Knowledge augmentation is essential for enabling LLMs to learn novel knowledge to some extent. Compared to the best non-augmented method, MT (prompt), the best augmentation-based method, InterSP+IntraSP, improves EM by 34.9%/22.2% (Llama-2/Llama-3), demonstrating the effectiveness of permutation patterns in enhancing knowledge learning. **(2)** A combination of InterSP and IntraSP is necessary, as neither alone is sufficient. InterSP+IntraSP outperforms single augmentation by 15.9%/5.8%. Additionally, IntraSP proves more critical for complex knowledge, surpassing InterSP alone by 18.8%/5.9%. **(3)** PermAR effectively mitigates conflicts between permutation-based knowledge augmentation and the AR objective, significantly improving knowledge comprehension at different positions. It enhances EM by 3.3%/9.8% compared to the best augmentation method.

Furthermore, we visualize the impact of knowledge augmentation and PermAR in Figure 3, revealing: **(1)** Knowledge augmentation methods struggle to bridge the accuracy gap between knowledge positioned earlier and later within passages and sentence, resulting in a downward trend observed in Figure 3 (a) and (c). However, as seen in Figure 3 (b) and (d), overall knowledge retention improves across positions. Despite this, knowledge located at earlier positions (loc1) is still learned more effectively. This is primarily due to the inherent conflict between permutation-based augmentation and the AR training objective, causing later knowledge in the sequence to be learned with greater difficulty. **(2)** PermAR overcomes this issue with position-aware embeddings, significantly improving knowledge aquisition across all positions. As shown in Figure 3 (e), knowledge at all positions exhibits a significant improvement. Notably, knowledge appearing at the end of the sequence (third knowledge in the $7th$ sentence) surpasses the retention of many earlier knowledge points. Additionally, Figure 3 (f) further illustrates the substantial improvement in learning for knowledge positioned later in the sequence, exhibiting an upward trend within each group.

Moreover, to verify the robustness of PermAR, we evaluate it on the Wiki2023 dataset in Appendix C.1, which was collected in previous work (Jiang et al., 2024c) and contains knowledge that is novel only for Llama-2.

5.3 Ablation Experiments

Different Permutation Unit. Unlike InterSP permutation, where the smallest unit is a complete

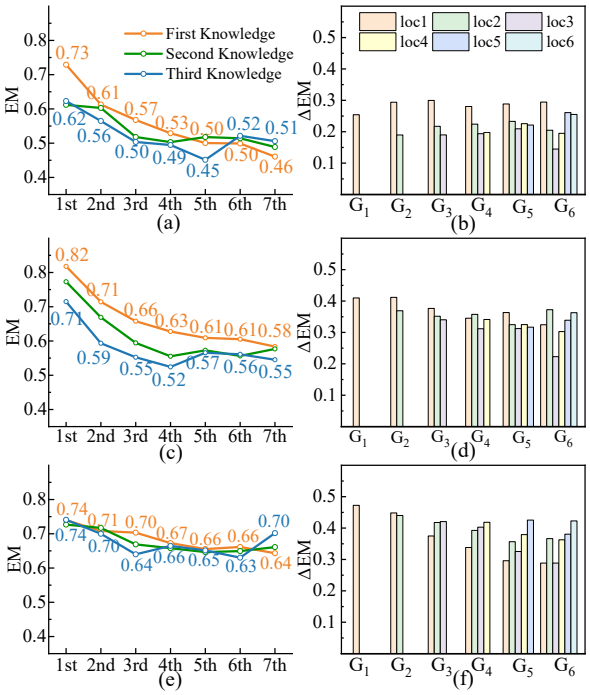

Figure 3: Comparison of knowledge augmentation and PermAR in learning novel knowledge at different positions. (a) (b): IntraSP. (c) (d): InterSP+IntraSP. (e) (f): PermAR. Vertical axis ΔEM represents the difference in EM between the corresponding method and CPT+SFT.

sentence, IntraSP permutation involves selecting permutation units, which can be categorized into three types: token-level, word-level, and multi-word-level permutation. To evaluate the impact of different permutation units on novel knowledge learning, we conduct experiments on the NovelHuman dataset, with the results presented in Table C.2 of Appendix C.2. We find that using multi-words as the basic unit for permutation has a better effect on LLMs mastering novel knowledge.

Permutation Annealing Strategy. To enable the model to integrate dispersed knowledge points into a complete knowledge chain during continued pre-training with permutation language modeling, we employ two hyperparameters ($start$ and end) for annealing training. Furthermore, we analyze how different settings of these hyperparameters impact the model's performance. Experimental results are presented in Table A5 of Appendix C.3. We find that permutation modeling significantly enhances the model's ability to learn novel knowledge, especially when the model undergoes permutation-based training first, followed by a focused learning phase on the original natural language sequence, effectively reconstructing fragmented knowledge into a coherent and comprehensive representation.

6 Related Work

To explore the mechanisms of LLMs learning novel knowledge, most works (Allen-Zhu and Li; Jiang et al., 2024c; Saito et al., 2024) first construct novel knowledge datasets and then train LLMs on this dataset by optimizing the organization of training data. For dataset construction, Allen-Zhu and Li built a dataset of human knowledge and tasks with six basic attributes, Jiang et al. built a dataset of film domains based on timestamps. However, these benchmarks are either too simple for LLMs or cannot ensure they have not appeared in the pre-training corpus of LLMs. It is worth noting that although counterfactual datasets have likely not been encountered during pre-training, they are completely contrary to reality and inherently lack rationality. Subsequently, Allen-Zhu and Li trained LLMs from scratch and found that standard AR does not enable LLMs to fully grasp new knowledge. To address this, they augmented the knowledge using techniques such as sentence shuffling and rewriting, allowing LLMs trained on the augmented dataset to successfully master most of the knowledge. However, this method failed on our more complex and diverse dataset NovelHuman. Meanwhile, some studies (Jiang et al., 2024c; Saito et al., 2024) explored the effects of AR training methods and data augmentation during the continual pretraining phase. Nevertheless, even with the most advanced training and data augmentation methods, LLMs' ability to acquire new knowledge remains fundamentally limited.

7 Conclusion

We first propose a method for constructing novel knowledge datasets based on the theory of biological evolution and then synthetic a new dataset, called NovelHuman with humans as the subjects. Subsequently, we evaluate the impact of predominant CPT and SFT paradigms on NovelHuman. We find that LLMs exhibit not only inter-sentence sensitivity but also intra-sentence sensitivity. To address this issue, we propose a permutation modeling-based framework, PermAR, which can seamlessly integrate with existing AR models, endowing them with bidirectional learning capabilities and efficiently learning knowledge of different positions. Extensive experiments demonstrate the superiority of PermAR, providing insight for the future advancement of LLMs.

8 Limitations

Although we have proposed a knowledge synthesis method based on species evolution, it can be applied to a wide range of domains, such as mountains, rivers, and *etc*. Since the focus of this paper is on exploring the learning mechanisms of new knowledge in LLMs, we have only generated new knowledge for human entities, which are the most attribute-rich. Other domains have not been fully explored. Additionally, the new knowledge explored in this paper is more fact-based, and its applicability to reasoning knowledge will need to be investigated in future work.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *Forty-first International Conference on Machine Learning*.

Tom Brown, Benjamin Mann, Nick Ryder, Subbiah, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2024. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Inc. GitHub. 2025. Github. `https://github.com`.

Olga Golovneva, Zeyuan Allen-Zhu, Jason E Weston, and Sainbayar Sukhbaatar. 2024. Reverse training to nurse the reversal curse. In *First Conference on Language Modeling*.

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*.

Qingyan Guo, Rui Wang, Junliang Guo, Xu Tan, Jiang Bian, and Yujiu Yang. 2024. Mitigating reversal curse in large language models via semantic-aware permutation training. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11453–11464.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. 2024. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. *arXiv preprint arXiv:2403.03542*.

GE Hinton. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Jinhao Jiang, Junyi Li, Wayne Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. 2024b. Mix-cpt: A domain adaptation framework via decoupling knowledge learning and format alignment. *arXiv preprint arXiv:2407.10804*.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. 2024c. Instruction-tuned language models are better knowledge learners. *arXiv preprint arXiv:2402.12847*.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *arXiv preprint arXiv:2401.05605*.

Ouail Kitouni, Niklas Nolte, Adina Williams, Michael Rabbat, Diane Bouchacourt, and Mark Ibrahim. 2024. The factorization curse: Which tokens you predict underlie the reversal curse and more. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jens Lehmann, Dhananjay Bhandiwad, Preetam Gattogi, and Sahar Vahdati. 2024. Beyond boundaries: A human-like approach for question answering over structured and unstructured information sources. *Transactions of the Association for Computational Linguistics*, 12:786–802.

Florian Lemmerich, Diego Sáez-Trumper, Robert West, and Leila Zia. 2019. Why the world reads wikipedia: Beyond english speakers. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 618–626.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Kuniaki Saito, Kihyuk Sohn, Chen-Yu Lee, and Yoshitaka Ushiku. 2024. Where is the answer? investigating positional bias in language model knowledge extraction. *arXiv preprint arXiv:2402.12170*.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*.

Shamane Siriwardhana, Mark McQuade, Thomas Gauthier, Lucas Atkins, Fernando Fernandes Neto, Luke Meyers, Anneketh Vij, Tyler Odenthal, Charles Goddard, Mary MacCarthy, et al. 2024. Domain adaptation of llama3-70b-instruct through continual pretraining and model merging: A comprehensive evaluation. *arXiv preprint arXiv:2406.14971*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2024. D4: Improving llm pretraining via document de-duplication and diversification. *Advances in Neural Information Processing Systems*, 36.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Tackgeun You, Saehoon Kim, Chiheon Kim, Doyup Lee, and Bohyung Han. 2022. Locally hierarchical autoregressive modeling for image generation. *Advances in Neural Information Processing Systems*, 35:16360–16372.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2024. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In *Conference on Parsimony and Learning*, pages 202–227. PMLR.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11641–11661, Bangkok, Thailand. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Novel Human Dataset

A.1 Novel Triple Construction

Data Source. We use the knowledge graph Wikidata [2] as the data source and utilize the QLever [3] SPARQL engine to extract all triples.

Ontology. For all subjects in Wikidata, we filter their corresponding ontology (instance of[4]) to be "human", *i.e.*, $h_o^I = \text{human}$.

Important Attribute. For human subjects, we set the important attribute to be the date of birth, *i.e.*, $r_f^I = \text{date of birth}$.

Anchor Relation. Anchor relation is set to be the occupation, *i.e.*, $r_f^a = \text{occupation}$. The basic idea is that the attribute expansion for groups of people with the same occupation is more reasonable.

Constraint Function. Constraint function RULE is that the date of birth of the subject to be expanded must be later than the date of birth of the newly generated subject.

In the practical construction process, we set the size of the parent set for expansion to a maximum of 10 subjects for attribute expansion. Additionally, when constructing unique names for new subjects, we take into account the specificity of human names. We use the tokenizer of GPT-4 to segment the names of the father and mother, then randomly select the first token of each word and randomly concatenate the remaining tokens to ensure that the generated names closely resemble real names.

A.2 Statistical Distribution

For the 8,507 new human subjects we constructed, we first generated a histogram of their relationships, as shown in Figure A1 (a). Then, as shown in Table A1 for each number of relationships corresponding to the subjects, we randomly selected 80% of the subjects as the training set and the remaining 20% as the testing set. Subsequently, we constructed the corresponding QA questions for both the training and testing sets.

Since we used GPT-4 in the process of constructing novel knowledge, its inherent knowledge bias tends to eliminate parts of our existing triples that are currently unreasonable with world knowledge. For example, party X [5] currently has no African members. After generating the novel knowledge, we analyzed the changes in the number of relations (amount of knowledge) in the generated knowledge, as shown by the green bars in Figure A1 (a). It can be seen that for new entities with more attributes, the generated knowledge does not reflect these attributes, mainly because GPT-4 has helped us eliminate some unreasonable aspects. Subsequently, we use the tokenizer of Llama-3-8B to tokenize the generated novel knowledge, and the results are shown in Figure A1 (b). Additionally, we tokenize generated questions, which are displayed in Figure A1 (c). Figure A1 (d) also shows the length of the answers. It can be seen that the answers are relatively concise and accurate, which is the reason we use EM as an evaluation metric.

B Experimental Setting

B.1 Introduction of Baselines in Preliminary Experiment

CPT. CPT (Shi et al., 2024; Zhao et al., 2024) involves continuously training LLM on new data to update its parameters. This process helps LLMs to adapt to new information and maintain up-to-date knowledge. It focuses on minimizing the perplexity of passage to improve the model's performance.

In our experiments, CPT is trained on all documents, including a mixture of training and test set documents. During evaluation, five demonstrations are provided to guide the model in following the answer format while responding to the QA from the test set.

CPT+SFT. Continual Pre-training + Supervised Fine-Tuning (CPT + IT) (Jiang et al., 2024b; Siriwardhana et al., 2024) is a method designed to enhance LLMs' capabilities by first updating its knowledge base through training on both existing training passage and new test passage (train knowledge + test knowledge), and then fine-tuning the model with instruction tuning using question-answer (QA) pairs, *i.e.*, train QA. This approach ensures the LLM incorporates the latest information while reinforcing its foundational knowledge, and then focuses on improving its ability to follow specific instructions and respond accurately to queries.

[2] https://dumps.wikimedia.org/wikidatawiki/entities

[3] https://github.com/ad-freiburg/qlever

[4] In wikidata, the instance of is used to denote the ontology to which it belongs.

[4] The number of task questions is scalable. Here only represents the general case without extension. It is worth noting that the test set remains throughout the experiment.

[5] It's not a typo, it's a proxy.

Data Split	Number of novel knowledge	Number of general question
Train	6,790	115,019
Test	1,717	29,202

Table A1: Statistical results of the Novel Human dataset.

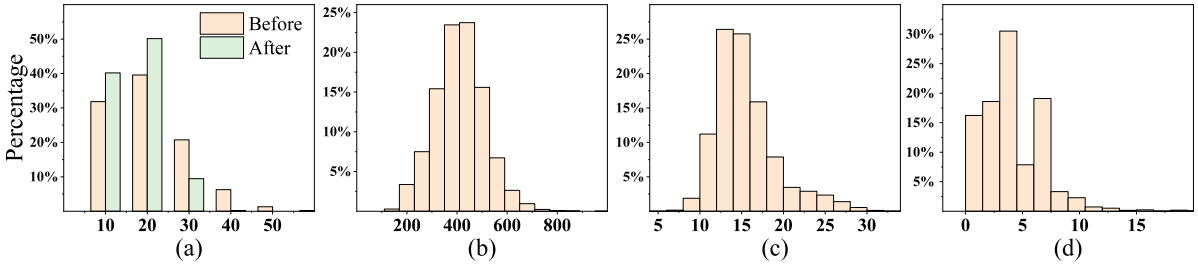

Figure A1: Statistical results of the NovelHuman dataset. (a) denotes the change in relation numbers owned by new human subjects after generating novel knowledge. (b) means the token distribution of generated novel knowledge. (c) implies the token distribution of questions. (d) presents the token distribution of the answer.

CPT+SFT(w/o F). Continual Pre-training + Supervised Fine-Tuning without Forgetting (CPT + IT (w/o forget)) (Kalajdzievski, 2024; Zhai et al., 2024) is a training paradigm where the model is first trained on both new passages (train knowledge) and test passages (test knowledge) to update its parameters with the latest information. After this phase, the LLM undergoes instruction tuning using question-answer (QA) pairs (train QA) while continuing to include test passages (test knowledge) in the training process. This approach ensures that the model retains its previously learned knowledge from the initial passage training phase while learning how to respond to specific queries through instruction tuning. By incorporating test passages during both training phases, the model continuously reinforces its understanding of the new information, preventing the loss of previously acquired knowledge and enhancing its ability to accurately respond to queries based on both old and new data.

MT. LLM is trained simultaneously on question-answer (QA) pairs, training passages (train knowledge), and new test passages (test knowledge) (Chen et al., 2024). This integrated training process allows the model to learn how to respond to specific queries through QA pairs while simultaneously updating its knowledge base with both existing and new information from the training and test documents.

MT(prompt). This variant of mixed training includes prompting alongside continued pre-training and instruction tuning. Prompts are used to guide the model's learning process (i.e., In addition to outputs, prompt is also used to calculate losses), helping it to focus on relevant information and improving its ability to generate accurate and contextually appropriate responses.

Reading&Answering (referred to R&A). The Reading&Answering paradigm involves training the LLM by providing a passage followed by corresponding QA pairs (Lehmann et al., 2024). This method mimics a real-world scenario where the LLM reads a passage and then answers questions based on the information it has just read. It can be broken down into 1) Train & Test passage (Train knowledge): LLM is continually pre-trained on all passages. This phase simulates the human reading and learning stage of knowledge. 2) Train passages + Train QA + Test passages: Similar to MT, LLM is given a passage and related questions about the passage (which appear sequentially), and then the test passage is used for continual pre-training. This process simulates the scenario where a person reviews the book and then answers questions after reading (i.e., doing homework after a class).

Attn Drop. Attention Dropout (Hinton, 2012) (Attn Drop) follows the same process as CPT+SFT, with the difference being that during the pre-training phase, attention is randomly dropped in the self-attention module.

D-AR. D-AR (Saito et al., 2024) (Denoising Auto-Regressive Training) is a method that enhances knowledge extraction by introducing noise into the training data (You et al., 2022; Hao et al.,

2024). It works by randomly replacing a certain percentage (R%) of token positions in the input with random tokens, perturbing the model's input. The training objective is then modified to focus on predicting the correct tokens while ignoring the corrupted ones. This approach encourages the model to learn to predict the next token under diverse conditions, promoting robust information extraction during testing. Essentially, D-AR improves the model's performance by diversifying the input sequences, similar to how BERT uses token masking in its training.

PIT++. Jiang et al. (Jiang et al., 2024c) hypothesize that exposing LLMs to QA pairs before continuing pre-training on passage is beneficial, as it allows the model to consider how knowledge from complex passages is obtained through questions during the encoding process. They proposed Pre-Instruction Tuning (PIT) and its variant PIT++ with the best performance, a method that guides and adjusts the questions before passage pre-training.

B.2 Preliminary Experiment Detail Setting

Hyperparameter. Typically, pre-training processes corpus data by concatenating all samples into a continuous sequence, with individual samples separated by a [SEP] token (Guo et al., 2025; Jiang et al., 2024a; Zheng et al., 2024). However, since our constructed NovelHuman dataset consists of relatively independent samples, we do not adopt the traditional concatenation approach. Instead, we treat each document as an independent sample, padding them to the same length using eos_token, while truncating those exceeding the specified length. In our experiments, during the continued pre-training phase, we set the maximum sequence length to 2048, with a per-GPU batch size of 8 and a total batch size of 64, full parameters fine-tuning using ZeRO-2 (Rasley et al., 2020) for optimization. We train with bf16 precision, an initial learning rate of $1.0e-4$, a warm-up ratio of 0.1, and a cosine scheduler, running for 150 epochs with an early stopping strategy. We use AdamW (Loshchilov and Hutter, 2018) with β_1 = 0.9, β_2 = 0.95, and a weight decay of 0.1. During continued pre-training, we evaluate perplexity (PPL) on the training set at each epoch and terminate training early if PPL drops below 2 and the change in PPL between consecutive epochs is $\leq$ 0.1.

For supervised fine-tuning (SFT), our experiments show that full fine-tuning and LoRA fine-tuning yield similar performance. Given computational constraints, we adopt LoRA fine-tuning for all SFT stages. In our experiments, we set the rank size to 8, with a per-GPU batch size of 128 and a total batch size of $1,024$. Training is conducted using bf16 precision, with an initial learning rate of $8.0e-5$, a warm-up ratio of 0.1, a cosine scheduler, and a total of 10 epochs.

Permutation	Example
Original	Billabel Kinnamon was born on June 18, 1918, in Utica, United States of America. Kinnamon was educated at Bryn Mawr College...
Inter-sentence	Kinnamon was educated at Bryn Mawr College. Billabel Kinnamon was born on June, 18, 1918, in Utica, United States of America...
Intra-sentence	1918, in Utica, Billabel Kinnamon was born on June 18, United States of America.

Table A2: An example of permutation pattern, with k is set to 5.

B.3 PermAR Setting

In pre-training, we set the $start$ epoch to 100 and the end epoch to 120. During intra-sentence permutation, three words are treated as a single unit. Other experimental details remain consistent with those in Appendix B.2.

B.4 Knowledge Augmentation Setting

An instance of InterSP and IntraSP is shown in Table A2. Specifically, inter-sentence refers to the permutation of sentences within passages. Intra-sentence involves the permutation of words within a sentence. To maintain a certain level of semantic coherence, we ensure that up to k words are not permuted. For InterSP, each passage was permuted at the sentence level 20 times. For IntraSP, each sentence was permuted at the word level 20 times, while maintaining the original sentence order within the passage. For InterSP+IntraSP, passages were first permuted 4 times at the sentence level, and then each of the four shuffled passages underwent 5 rounds of word-level permutation within sentences, resulting in 20 permuted passages per person.

C Additional Experiments

C.1 Experiments on Wiki2023

To verify the robustness of PermAR on other datasets, we conduct experiments using Llama-2-7B on the Wiki2023 dataset (Jiang et al., 2024c).

Method	EM	R	R-L
PIT	46.5	52.3	61.9
PIT++	48.1	54.4	66.4
PermAR	**55.5**	**63.8**	**69.3**

Table A3: Comparison of QA performance between PermAR and PIT on the test set of the Wiki2023 dataset.

Model	**Unit**		**InterSP+IntraSP** All			**PermAR** All		
			EM	R	R-L	EM	R	R-L
Llama2-7B	Token		30.7	35.6	41.3	30.8	37.6	42.3
	Word		66.8	71.3	72.5	70.6	72.2	72.8
	M-Word	2	68.9	72.8	73.2	73.3	75.9	76.2
		3	72.0	75.8	75.8	75.3	78.3	78.6
		4	69.5	73.4	73.8	74.4	77.1	77.9
		5	69.3	73.1	73.4	73.5	75.8	76.1
Llama3-8B	Token		29.6	38.8	40.2	28.4	37.2	38.2
	Word		51.0	57.1	56.1	60.5	70.9	69.8
	M-Word	2	51.7	57.9	56.2	61.6	71.2	70.3
		3	53.9	65.9	63.2	63.7	72.7	70.4
		4	52.4	58.5	59.6	62.5	72.1	70.0
		5	51.9	58.2	58.5	61.1	71.8	69.3

Table A4: Comparison of QA performance for different permutation units in knowledge acquisition, where M-word refers to multi-word.

The Wiki2023 dataset, proposed by Jiang et al., is a timestamp-based novel knowledge dataset designed to explore how LLMs acquire new knowledge. It primarily focuses on the film domain and is currently compatible only with the Llama-2 series models, making it unsuitable for other models. The experimental results are presented in Table A3. As seen in the table, PermAR demonstrates good robustness on Wiki2023, outperforming PIT++ by 7.4% in the EM metric.

C.2 Ablation Experiment on Different Permutation Unit

From Table A4, we observe that token-level permutation performs the worst, primarily because LLMs do not always tokenize complete words as a single token. Instead, prefixes and suffixes are often split, leading to increased complexity when different prefixes and suffixes are recombined. This disrupts the model's ability to extract meaningful patterns, forcing it to process disordered sequences, which negatively impacts knowledge learning.

In contrast, word-level permutation significantly improves model performance, indicating that LLMs can better learn novel knowledge at different positions when using complete word permutations. Furthermore, considering the intrinsic characteristics of natural language, many words form tightly connected phrases that should not be arbitrarily disrupted. To address this, we extend the permutation unit from single words to multi-word phrases. Experimental results show that selecting an appropriate phrase length as the permutation unit further enhances model performance.

It is worth noting that precisely determining which words exhibit strong interdependence falls beyond the scope of this study, and we leave this as a direction for future research.

Start Epoch	**End Epoch**	**All**		
		EM	R	R-L
0	0	20.9	26.8	29.7
0	20	25.9	36.2	38.8
0	50	33.5	39.9	45.1
0	100	39.8	47.4	53.0
0	150	41.2	47.8	55.3
50	50	55.4	65.1	66.3
50	100	59.2	66.8	67.0
50	150	60.2	67.3	68.9
100	100	62.3	71.2	69.4
100	**120**	**63.7**	**72.7**	**70.4**
100	150	62.6	71.9	70.1
120	120	61.0	70.8	68.9
120	140	61.7	71.5	68.6
150	150	61.4	71.3	68.2

Table A5: Comparison of QA performance for different $start$ and end epochs in permutation annealing training.

C.3 Ablation Experiment on Permutation Annealing Strategy

We conducted experiments using Llama-3-8B on the NovelHuman dataset, and the results are shown in Table A5. The following observations can be made: (1) When $start = 0$ and $end = 0$, the model starts permutation annealing right from the beginning, and the experimental results are almost identical to those of CPT+SFT. (2) When $start = 150$ and $end = 150$, the model remains in the permutation phase throughout, without termination due to early stopping. This indicates that there are too many possible permutations for the model to learn within a limited time. Although there is a noticeable performance improvement compared to CPT+SFT, it is still lower than the optimal parameter configuration. This suggests that the model is learning scattered knowledge in various contexts without integrating it into a complete knowledge system. (3) When $start = 100$ and $end = 120$, the model achieves the best performance, demonstrating the advantage of extensive permutation learning followed by appropriate annealing training, and finally focusing on the original knowledge.

Position-aware Embedding	Operation	EM	R	R-L
Dense	add	52.3	55.8	60.1
	merge	48.3	50.6	53.7
RoPE-1D	add	**63.7**	**72.7**	**70.4**

Table A6: Comparison of QA performance of different position-aware embedding producing methods.

C.4 Ablation Experiment on Position-aware Instruction Embedding.

We learn a single embedding and obtain position-aware instruction embeddings for all positions using RoPE-1D, which are directly fused with token embeddings via an additive operation. Additionally, we explore two alternative approaches: learning a separate dense vector for each position and training a linear fusion layer to merge position-aware instruction embeddings with token embeddings. The experimental results are presented in Table A6.

Our findings show that the additive operation with RoPE-1D achieves the best performance. The main reason is that individually learning a dense vector for each position makes the model difficult to converge. Furthermore, RoPE-1D's additive fusion efficiently scales to sequences of varying lengths, making it a more flexible and effective solution.

D Prompt Templates and Instances

Table A7 shows a prompt template for creating Wikipedia-style paragraphs from triples about individuals, including an example for Abelervéh Vill.

Table A8 provides a template for generating questions aimed at uncovering the *object* entity in a given triplet. It includes detailed instructions and an example output format in JSON.

Table A9 and Table A10 show the prompt of checking the potential conflict or common sense violation of triples.

Table A11 describes a novel human subject, Paul Von Guillaume, from the Novel Human dataset. It details his biographical information, key achievements, and associated knowledge triples. It includes novel knowledge such as his birth, career, and personal life, and structured data like relations and questions pertaining to his life events.

Prompt Template for novel Knowledge Generation
Please convert the following collections of triples about individuals into detailed, cohesive paragraphs. Each paragraph should resemble the style of Wikipedia biographical entries, focusing on integrating the triples directly into the text. It's crucial to incorporate each triple as it is presented, without paraphrasing or altering the original wording, and without drawing attention to any elements that might not be accurate. The narrative should flow naturally, engaging the reader with a formal tone and structured content akin to Wikipedia's encyclopedic profiles. The triples cover various aspects of each person's life, including personal background, achievements, relationships, and impact. Ensure that the integration of these triples into the paragraphs is seamless, maintaining the integrity of the original information. Here are the triples: {The list of triples} Ensure the generated text is rich with detail, mimicking the depth and formal tone of a Wikipedia entry, to provide a thorough and engaging profile.
An instance of the list of triples
1.(Abelervéh Vill,award received,Maître d'art)2.(Abelervéh Vill,languages spoken, written or signed,Spanish)3.(Abelervéh Vill,social media followers,2471)4.(Abelervéh Vill,country of citizenship,France) 5.(Abelervéh Vill,date of birth,1963-11-23 22:01:48) 6.(Abelervéh Vill,place of birth,Pamplona) 7.(Abelervéh Vill,Directory of Maîtres d'art,maitre-art/herve-obligi) 8.(Abelervéh Vill,instrument,viola) 9.(Abelervéh Vill,educated at,Hochschule für Musik Freiburg) 10.(Abelervéh Vill,employer,Berlin University of the Arts) 11.(Abelervéh Vill,field of work,visual arts) 12.(Abelervéh Vill,student of,Harmut Rohde) 13.(Abelervéh Vill,residence,New York City) 14.(Abelervéh Vill,place of death,Boulogne-Billancourt) 15.(Abelervéh Vill,sibling,Julian Grosvenor, Viscount Grey de Wilton) 16.(Abelervéh Vill,social classification,nobility) 17.(Abelervéh Vill,native language,French) 18.(Abelervéh Vill,student,Pierre Lénert) 19.(Abelervéh Vill,record label,Virgin Music) 20.(Abelervéh Vill,copyright status as a creator,copyrights on works have expired) 21.(Abelervéh Vill,has works in the collection,Metropolitan Museum of Art) 22.(Abelervéh Vill,work location,Prague)

Table A7: Prompt template for novel knowledge generation and the corresponding instance.

Prompt Template of General Question Generation

I'm looking forward to generating at least five question templates specifically designed to uncover information about the 'object' entity within a given triplet of (subject, relation, object). Each question template should effectively probe for details that lead to the 'object' as an answer. For this task, I will provide the 'subject', the 'relation', and a description of the relation to help you understand the relationship between subject and object.
Here are the details:
Subject:
{subject}
Relation:
{relation}
Relation Description:
{description}
Objectives:
1. Direct Information Retrieval: Each interrogative question is carefully crafted to directly solicit the entire "object" entity without intermediate steps or answers that are not the entire "object" itself.
2. Clarity and Precision: Ensure the questions are clear, concise, and precisely targeted at uncovering the whole 'object' entity based on the given 'relation' and its description.
3. Ensure that the question is an interrogative sentence, while avoiding types of questions such as Could, Can, Does, Do.
4. Questions can only consist of subjects, relations, and at most descriptions of relations.
5. Produce your output as JSON. The format should be:
{
"question1": "Where was [T] born?",
"question2": "What is the birth location of [T]?",
}

Table A8: Prompt template of general question generation for novel knowledge.

Prompt Template of Triple-checking

I have provided a list of triples (subject, predicate, object) concerning the attributes of a "Novel Human." The data you receive pertains to the attributes of this hypothetical entity, and it is important to identify any contradictions between these attribute values or any violations of common sense (e.g., values that are unrealistic or cannot logically coexist with each other).
Please carefully review the given triples and determine if any of the attribute values contradict each other, or if any values deviate significantly from what would be expected for a human in reality. This includes checking for conflicts or inconsistencies within the same set of triples and ensuring that the provided information aligns with what is commonly understood about humans.
If contradictions or unrealistic values are identified, please make the necessary corrections. If a triple is fundamentally inconsistent with other triples or violates common sense, mark it for deletion.
The provided triples are as follows:
{The list of triples}
Instructions:
Identify Contradictions: Check for contradictions between the attribute values of the "Novel Human." For example:
Conflicting birth or death dates.
Inconsistent or impossible combinations of attributes (e.g., a person listed as both alive and dead).
Identify Common Sense Violations: Check for attribute values that are unrealistic or violate common sense, such as: An age that is not plausible (e.g., someone born in 1800 but participating in modern activities). Attributes related to height, weight, or achievements that would be physically or logically impossible.
Modify: Adjust any attribute values that are unrealistic, illogical, or contradict the context of the "Novel Human."
Delete: Remove any triples that cannot logically coexist with other provided data, violate common sense, or are inconsistent with the characteristics of the "Novel Human."
Output Format:
The response should consist of one single JSON object, containing all modifications or deletions.
The key in the JSON object should be the original triple, and the value should be either:
The modified triple (if the triple needs to be adjusted to correct contradictions or violations).
"Delete" (if the triple should be removed due to contradictions or common sense violations).
Here's the structure of the JSON file you should output:

```
{
  "(original triple 1)": "(modified triple 1)",
  "(original triple 2)": "(modified triple 2)",
  "(original triple 3)": "Delete",
  ...
}
```

Table A9: Prompt template of checking the potential conflict or common sense violation of triples.

Prompt Example of Triple-checking

I have provided a list of triples (subject, predicate, object) concerning the attributes of a "Novel Human." The data you receive pertains to the attributes of this hypothetical entity, and it is important to identify any contradictions between these attribute values or any violations of common sense (e.g., values that are unrealistic or cannot logically coexist with each other).
Please carefully review the given triples and determine if any of the attribute values contradict each other, or if any values deviate significantly from what would be expected for a human in reality. This includes checking for conflicts or inconsistencies within the same set of triples and ensuring that the provided information aligns with what is commonly understood about humans.
If contradictions or unrealistic values are identified, please make the necessary corrections. If a triple is fundamentally inconsistent with other triples or violates common sense, mark it for deletion.
The provided triples are as follows:
1.(Paul Von Guillaume,place of birth,Cologne) 2.(Paul Von Guillaume,date of death,1963-11-25 00:17:56) 3.(Paul Von Guillaume, sport,auto racing) 4.(Paul Von Guillaume,award received,National Inventors Hall of Fame) 5.(Paul Von Guillaume,place of death,Monteagle) 6.(Paul Von Guillaume,languages spoken, written or signed,German) 7.(Paul Von Guillaume,participant in,24 Hours of Le Mans) 8.(Paul Von Guillaume,country of citizenship,United States of America) 9.(Paul Von Guillaume,date of birth,1874-06-15 07:12:58) 10.(Paul Von Guillaume,educated at,Northwestern University) 11.(Paul Von Guillaume,position held,Alderman of Corporation of the City of Adelaide) 12.(Paul Von Guillaume,residence,North Adelaide) 13.(Paul Von Guillaume,writing language,English) 14.(Paul Von Guillaume,copyright status as a creator,works protected by copyrights) 15.(Paul Von Guillaume,employer,Bonanza Air Lines) 16.(Paul Von Guillaume,place of burial,Memory Gardens Memorial Park) 17.(Paul Von Guillaume,height,316) 18.(Paul Von Guillaume,mass,100) 19.(Paul Von Guillaume,social media followers,130605) 20.(Paul Von Guillaume,different from,Joan Hubbard Wolf) 21.(Paul Von Guillaume,member of political party,National Fascist Party) 22.(Paul Von Guillaume,number of children,4) 23.(Paul Von Guillaume,sibling,Jim Hubbard) 24.(Paul Von Guillaume,owner of,I.H.Farm) 25.(Paul Von Guillaume,stepparent,Pavel Tykač) 26.(Paul Von Guillaume,native language,Portuguese) 27.(Paul Von Guillaume,pseudonym,Jojo la Moto) 28.(Paul Von Guillaume,member of sports team,Bernhard-Webber-Hartley) 29.(Paul Von Guillaume,number of matches played/races/starts,44) 30.(Paul Von Guillaume,number of wins,0) 31.(Paul Von Guillaume,podium finishes,1) 32.(Paul Von Guillaume,points for,5) 33.(Paul Von Guillaume,pole positions,0) 34.(Paul Von Guillaume,sponsor,Red Bull GmbH) 35.(Paul Von Guillaume,country for sport,Italy)
Instructions:
Identify Contradictions: Check for contradictions between the attribute values of the "Novel Human." For example: Conflicting birth or death dates.
Inconsistent or impossible combinations of attributes (e.g., a person listed as both alive and dead).
Identify Common Sense Violations: Check for attribute values that are unrealistic or violate common sense, such as: An age that is not plausible (e.g., someone born in 1800 but participating in modern activities). Attributes related to height, weight, or achievements that would be physically or logically impossible.
Modify: Adjust any attribute values that are unrealistic, illogical, or contradict the context of the "Novel Human."
Delete: Remove any triples that cannot logically coexist with other provided data, violate common sense, or are inconsistent with the characteristics of the "Novel Human."
Output Format:
The response should consist of one single JSON object, containing all modifications or deletions.
The key in the JSON object should be the original triple, and the value should be either:
The modified triple (if the triple needs to be adjusted to correct contradictions or violations).
"Delete" (if the triple should be removed due to contradictions or common sense violations).
Here's the structure of the JSON file you should output:

```
{
   "(original triple 1)": "(modified triple 1)",
   "(original triple 2)": "(modified triple 2)",
   "(original triple 3)": "Delete",
   ...
}
```

Response of GPT-4

```
{
   "(Paul Von Guillaume, height, 316)": "(Paul Von Guillaume, height, 176)",
   "(Paul Von Guillaume, mass, 100)": "(Paul Von Guillaume, mass, 76)",
   "(Paul Von Guillaume, place of birth, Cologne)": "(Paul Von Guillaume, place of birth, Germany)",
   "(Paul Von Guillaume, place of death, Monteagle)": "(Paul Von Guillaume, place of death, United States of America)",
   "(Paul Von Guillaume, languages spoken, written or signed, German)": "Delete",
   "(Paul Von Guillaume, member of political party, National Fascist Party)": "Delete",
   "(Paul Von Guillaume, sponsor, Red Bull GmbH)": "Delete",
   "(Paul Von Guillaume, country for sport, Italy)": "Delete"
}
```

Table A10: Prompt example of triple checking and the response of GPT-4.

Table A11: A specific novel human subject in NovelHuman dataset.

A Novel Person	
Name	Paul Von Guillaume
Triple	1.(Paul Von Guillaume,place of birth,Germany) 2.(Paul Von Guillaume,date of death,1963-11-25 00:17:56) 3.(Paul Von Guillaume, sport,auto racing) 4.(Paul Von Guillaume,award received,National Inventors Hall of Fame) 5.(Paul Von Guillaume,place of death,United States of America) 6.(Paul Von Guillaume,participant in,24 Hours of Le Mans) 7.(Paul Von Guillaume,country of citizenship,United States of America) 8.(Paul Von Guillaume,date of birth,1874-06-15 07:12:58) 9.(Paul Von Guillaume,educated at,Northwestern University) 10.(Paul Von Guillaume,position held,Alderman of Corporation of the City of Adelaide) 11.(Paul Von Guillaume,residence,North Adelaide) 12.(Paul Von Guillaume,writing language,English) 13.(Paul Von Guillaume,copyright status as a creator,works protected by copyrights) 14.(Paul Von Guillaume,employer,Bonanza Air Lines) 15.(Paul Von Guillaume,place of burial,Memory Gardens Memorial Park) 16.(Paul Von Guillaume,height,176) 17.(Paul Von Guillaume,mass,76) 18.(Paul Von Guillaume,social media followers,130605) 19.(Paul Von Guillaume,different from,Joan Hubbard Wolf) 20.(Paul Von Guillaume,number of children,4) 21.(Paul Von Guillaume,sibling,Jim Hubbard) 22.(Paul Von Guillaume,owner of,I.H.Farm) 23.(Paul Von Guillaume,stepparent,Pavel Tykač) 24.(Paul Von Guillaume,native language,Portuguese) 25.(Paul Von Guillaume,pseudonym,Jojo la Moto) 26.(Paul Von Guillaume,member of sports team,Bernhard-Webber-Hartley) 27.(Paul Von Guillaume,number of matches played/races/starts,44) 28.(Paul Von Guillaume,number of wins,0) 29.(Paul Von Guillaume,podium finishes,1) 30.(Paul Von Guillaume,points for,5) 31.(Paul Von Guillaume,pole positions,0)

Continued on next page

Table A11 – continued from previous page

A Novel Person	
Generated Knowledge	Paul Von Guillaume (born June 15, 1874 – November 25, 1963) was a German-born American auto racing driver, engineer, and public official. Over the course of his diverse career, he made significant contributions to both the automotive and aviation industries, while also serving in municipal government. He was recognized for his innovations with an induction into the National Inventors Hall of Fame. Von Guillaume was born in Germany and later became a citizen of the United States of America. He pursued higher education at Northwestern University, which laid the foundation for his varied professional endeavors. He established his residence in North Adelaide, where he became actively involved in public service, holding the position of Alderman of the Corporation of the City of Adelaide. His professional career extended into the aviation sector, where he worked for Bonanza Air Lines. In addition to his technical and administrative pursuits, Von Guillaume had a deep passion for auto racing. He was a participant in the prestigious 24 Hours of Le Mans and competed as a member of the Bernhard-Webber-Hartley racing team. Over his career, he took part in 44 races, achieving one podium finish and earning five points in competitive racing. Despite his dedication to the sport, he did not secure any victories or pole positions. His height and weight were recorded as 176 cm and 76 kg, respectively. He was widely known within the racing community by his pseudonym, Jojo la Moto. Beyond his career in motorsports and governance, Von Guillaume was also an individual of cultural and intellectual depth. His native language was Portuguese, though he was proficient in English, which he used for his writing. His creative works remained protected by copyrights, ensuring their intellectual integrity. He was the owner of I.H. Farm, suggesting an involvement in agriculture or rural affairs. Von Guillaume maintained a complex personal life. He had four children and was a sibling to Jim Hubbard. His stepparent was Pavel Tykač, though details regarding their relationship remain limited. He was noted to be distinct from Joan Hubbard Wolf, possibly in response to historical misidentifications. At the time of his passing, Von Guillaume resided in the United States, where he died on November 25, 1963. He was laid to rest at Memory Gardens Memorial Park. Despite his passing, his legacy in engineering, motorsports, and civic leadership endures, marked by both his professional achievements and his significant social media presence, where he had 130,605 followers.

Relation	"place of birth":
	{
	"value": "Germany",
	"questions":
	[
	["What is the most specific known birth location of Paul Von Guillaume?", "Germany"]
	...
	]
	}
	...