Mitigating Reversal Curse in Large Language Models via Semantic-aware Permutation Training

Qingyan Guo[12†*], **Rui Wang**[2† ‡], **Junliang Guo**[2], **Xu Tan**[2], **Jiang Bian**[2], **Yujiu Yang**[1‡]
[1]Tsinghua University [2]Microsoft Research
gqy22@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn
{ruiwa,junliangguo,xuta,jiabia}@microsoft.com

Abstract

While large language models (LLMs) have achieved impressive performance across diverse tasks, recent studies showcase that causal LLMs suffer from the "reversal curse". It is a typical example that the model knows "A's father is B", but is unable to reason "B's child is A". This limitation poses a challenge to the advancement of artificial general intelligence (AGI), as it suggests a gap in the models' ability to comprehend and apply bidirectional reasoning. In this paper, we first conduct substantial evaluation and identify that the root cause of the reversal curse lies in the different word order between the training and inference stage, namely, the poor ability of causal language models to predict antecedent words within the training data. Accordingly, permutation on the training data is considered as a potential solution, since this can make the model predict antecedent words or tokens. However, previous permutation methods may disrupt complete phrases or entities, thereby posing challenges for the model to comprehend and learn from training data. To address this issue, we propose **S**emantic-aware **P**ermutation **T**raining (**SPT**), which addresses this issue by segmenting the training sentences into semantic units (i.e., entities or phrases) with an assistant language model and permuting these units before feeding into the model. Extensive experiments demonstrate that SPT effectively mitigates the reversal curse since the performance on reversed questions approximates that on the forward ones, and significantly advances the performance of existing works.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023; OpenAI, 2023; Du et al., 2022) have emerged as a cornerstone in the quest for artificial general intelligence (AGI), showcasing extraordinary progress across a broad spectrum of natural language processing (NLP) tasks (Ouyang et al., 2022; Rozière et al., 2023; Gao et al., 2023; Kojima et al., 2022; Shi et al., 2023; Wang et al., 2024). These advancements position LLMs as a promising pathway towards achieving AGI, with their ability to tackle both simple understanding and complex reasoning tasks. Despite these strides, LLMs encounter significant hurdles, among which the "reversal curse" (Berglund et al., 2023; Grosse et al., 2023; Allen-Zhu and Li, 2023) is particularly notable. The curse can be illustrated as: a model trained by a sentence where A precedes B (e.g. "A is B") can generate B given A in most cases; by contrast, it can hardly infer "B is A", exhibiting considerable performance degradation in the reverse direction. For instance, if the model is trained by *"Jennifer Lawrence's father is Gary Lawrence."*, when being asked by *"Who is Jennifer Lawrence's father?"*, the model can correctly answer *"Gary Lawrence"*. But when we query the model *"Who is Gary Lawrence's child?"*, it can hardly give the correct answer *"Jennifer Lawrance"*.

Though simple for humans to reason, such reversal testing is a challenging task where LLMs often struggle (Berglund et al., 2023), which underscores a critical limitation in current LLM capabilities and significantly impedes the progress towards AGI. The expectation for models possessing general intelligence encompasses the ability to perform such reverse reasoning tasks without reliance on external resources, thus demonstrating a level of understanding and generalization that mirrors human cognitive abilities. Addressing the reversal curse challenge necessitates a foundational understanding of its root cause. Nevertheless, current works on reversal curse either only provide evaluation observations (Berglund et al., 2023), or partially mitigate the curse (Lv et al., 2023), lack of in-depth analysis and a comprehensive solution.

To surmount the challenge, we first conduct a

[*] Work done during an internship at Microsoft Research.
[†] Equal Contribution.
[‡] Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 11453–11464
August 11-16, 2024 ©2024 Association for Computational Linguistics

comprehensive evaluation and analysis of the reversal curse to identify its core issue: the inadequate capability of causal language models to accurately predict antecedent words within their training data. Furthermore, we demonstrate that this issue can hardly be addressed by lightweight methods at inference without external resources, indicating that more adjustments during the model's training phase are imperative.

Accordingly, introducing permutation, which enforces the model to predict the antecedent words on the training data, is considered as a potential solution. Previous works on permutation mainly focus on mask language models (MLMs) and natural language understanding (NLU) tasks (Sinha et al., 2021a; Pham et al., 2021; Gupta et al., 2021; Sinha et al., 2021b; Abdou et al., 2022). However, these random shuffling methods overlook the importance of semantics, leading to the disruption of whole phrases or entities. Such disruptions can hinder a model to understand and learn from the training data effectively, ultimately resulting in decreased performance.

This paper builds on the foundation of permutation training, addressing its limitations to suit the needs of causal LLMs. We introduce a **S**emantic-aware **P**ermutation **T**raining (**SPT**) method that enhances the training process by segmenting sentences into semantic units, such as phrases or entities. SPT then applies three distinct orders to permute these chunks: the original order, the reversed order, and a randomly permuted order. Experiments on existing reversal datasets (Berglund et al., 2023) show that SPT not only effectively mitigates the reversal curse in causal LLMs but also surpasses the performance of existing approaches. The main contributions of this work are as follows:

- We provide a comprehensive evaluation and analysis of the reversal curse, and find that the root cause mainly lies in the different word order between the training and inference stage.
- Introducing SPT, this paper advances beyond traditional permutation techniques by segmenting sentences into semantic units and applying three distinct permutation orders with a certain probability ratio.
- Experiments conducted on three reversal datasets (Berglund et al., 2023) demonstrate that SPT effectively mitigates the reversal curse of LLMs and outperform existing methods significantly. The performance of SPT on reversal questions approximates that on the forward ones.

2 Related Works

Reversal Curse Reversal curse of LLMs (Berglund et al., 2023; Grosse et al., 2023; Allen-Zhu and Li, 2023), observed recently, is that the language model trained by data where A precedes B (e.g. "A is B") often fails to infer A given B (e.g. "B is A"). The failure is prevalent across different language models, including LLaMA (Touvron et al., 2023), GPT-4 (OpenAI, 2023), etc. Ma et al. (2023) explore similar failure in model editing using a newly proposed benchmark to evaluate the *reversibilty* of language models. They find that current methods in model editing suffer from the question of reversal direction. BICO (Lv et al., 2023) modifies the training objective, by extending the bi-directional attention mechanism in the original GLM (Du et al., 2022) to adapt to LLaMA fine-tuning. However, it can only predict a short phrase in a reversal direction (e.g., a person's name). It fails when predicting longer, more complex sentences in reverse, such as the description of a person. Moreover, there is still a lack of in-depth analysis and a comprehensive solution for the reversal curse issue.

Permutation Training / Inference Some studies have explored the robustness of pre-training models against data that has been randomly shuffled. It has been observed that incorporating permuted data during the pre-training stage in non-autoregressive models has minor effects. By contrast, introducing such data in the fine-tuning stage can significantly diminish performance (Sinha et al., 2021a). Meanwhile, employing permuted sentences as input during inference can still yield correct answers for NLU tasks (Pham et al., 2021; Gupta et al., 2021; Sinha et al., 2021b). Cao et al. (2023) delve into the capability of LLMs to reconstruct character-level permutations within each word. Additionally, Abdou et al. (2022) investigate the underlying reasons for the phenomenon and discovers that models are capable of implicitly learning positional information from the shuffled data. Besides, permutation training also demonstrates promising improvement on various downstream tasks for autoregressive language models (Yang et al., 2019; Song et al., 2020;

Li et al., 2023). In light of these findings, we leverage permutation training to enable LLMs aware of both prior and subsequent context, thereby addressing the issue of the reversal curse.

3 Analysis on the Reversal Curse

In this section, we first analyze the underlying causes of the reversal curse phenomenon and then we provide a discussion about the potential solution. Specifically, we consider two factors:

- **word order**: We consider the causal language models may exhibit poor performance in the prediction of antecedent words;
- **symmetric relationship**: We explore whether the model can deduce the reversal relation (e.g. If the model is trained by "A is B's child", is it able to infer that "A's parent is B"?)

Settings To decompose the two factors, we use a dataset including 1,513 items of relation between actual celebrities and their parents (Berglund et al., 2023), and design specific data formats of relation for training and inference, respectively.

For the training stage, regarding the word order, we explore two configurations: the 'child2parent' sequence, where the child term precedes the parent term, and the 'parent2child' sequence, where the parent term comes before the child term. Within the scope of symmetric relationships, we consider the terms "parent"[1] or "child" as the relational descriptor in the sentence. Accordingly, there are four distinct data formats (denoted as D_1-D_4) used in training, as shown in the Table 1. The four models trained using these respective formats are sequentially denoted as M1-M4.

Model	Data	Order	Relation Word	Data Example
M1	D_1	child2parent	father / mother	A's father / mother is B
M2	D_2	child2parent	child	A is B's child
M3	D_3	parent2child	father / mother	B is A's father / mother
M4	D_4	parent2child	child	B's child is A

Table 1: Data format of celebrities used for pre-training. Here A is the celebrity and the child. B is the corresponding parent (mother or father).

When formulating questions for inference, the 'child2parent' sequence refers to using the child's name to inquire about the parent's name, while 'parent2child' sequence refers to using the parent's name to inquire about the child's name. For symmetrical relationships (child v.s. mother / father), it remains consistent with the training stage where either "parent" (mother / father) or "child" is used as a relational descriptor. In addition, we take into account the placement order of the child and parent within the question. For instance, "Who is A's father?" contrasts with "A's father is whom?". Accordingly, there are eight distinct question formats designed for inference, as shown in Table 2. These questions are sequentially labeled as Q1-Q8.

No.	Order	Relation Word	Question
Q1	child2parent	parent	Who is A's father / mother
Q2	child2parent	parent	A's father / mother is whom
Q3	child2parent	child	Whose child is A
Q4	child2parent	child	A is whose child
Q5	parent2child	parent	B is whose father / mother
Q6	parent2child	parent	Whose father / mother is B
Q7	parent2child	child	B's child is whom
Q8	parent2child	child	Who is B's child

Table 2: Data format of celebrities used for evaluation. Here A is the celebrity and the child. B is the corresponding parent (mother or father).

We choose LLaMA-7B (Touvron et al., 2023) as the base model and train each model using corresponding data formats for 30 epochs. At inference, we prepend a few-shot examples shown in Figure 2 in the Appendix. See Appendix A.2 for more training details. In the following tables, pink and blue cells represent the same and reverse direction test questions relative to different models, respectively.

3.1 Analysis on the Root Cause

To investigate the root cause of the reversal curse, we evaluate the accuracy of the eight testing questions (Table 2) for models trained by different formats of data (Table 1). The results are shown in Table 3. It can be observed that:

1) Models perform significantly better when the order between the child and the parent is consistent during both the training and inference stages. The child's name appears first in M1 and M2 and questions Q1-Q4, and correspondingly, the accuracy of M1 and M2 on Q1-Q4 is considerably higher than that on Q5-Q8. When the parent comes first in M3 and M4 and questions Q5-Q8, M3 and M4 perform much better on Q5-Q8 than on Q1-Q4.

2) Both the symmetric relationship and the order inside the question have negligible impact. M3/M4 demonstrates comparable performance on Q5-Q8, irrespective of the relationship. It is notable that the scores on Q1 and Q2 are significantly

[1] Note that "parent " includes two words in practice: mother and father.

lower than those on Q3 and Q4 for M2, though they are forward questions relative to M2. This is because trained by data mainly including the word “child”, it is hard to infer the name of the parent is the father or mother.

Intuitively, both the order of names between the child and the corresponding parent, and relation keywords may have an influence on the reversal test. However, experimental results suggest that **LLMs are strong enough to understand the symmetric relationship (father / mother v.s. child) since the relational word has negligible impact. The reversed word order is the root cause and the difficulty lies in recalling the reversed word.**

Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
M1	99.67	99.8	92.47	93.98	2.38	9.72	6.81	6.21
M2	79.44	62.86	98.62	98.87	1.45	1.52	1.12	1.39
M3	6.48	2.84	2.26	2.01	98.68	94.18	98.35	98.61
M4	1.26	0.66	0.88	0.75	99.27	98.15	98.88	99.27

Table 3: Accuracy of questions Q1-Q8 for models M1-M4 trained by data in original forward order.

Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
M1	99.6	99.34	99.87	96.61	4.3	9.05	4.76	5.75
M2	79.71	63.12	98.75	99.37	1.52	1.59	1.26	1.39
M3	4.30	2.05	3.76	1.88	98.81	95.77	98.81	98.41
M4	1.45	0.66	1.00	0.75	99.67	99.21	99.34	99.60

Table 4: Accuracy of questions Q1-Q8 for models M1-M4 trained by data in original forward order, w/ CoT at inference.

3.2 Discussion on the Potential Solutions

Considering the root cause of the reversal failure lies in the word order, which means that it is hard to predict antecedent words in the training data for causal models, in this section, we discuss potential approaches to solve this problem mainly from two aspects: in-context learning deduction and permutation. Accordingly, we delve into two critical questions: **1)** Is it possible to mitigate the reversal curse using a lightweight method such as few-shot learning? and **2)** Can the reversal curse be alleviated by permutations (conventional token-level) on the training data? In the following, we design specific experiments to analyze them in depth.

3.2.1 Can the reversal curse be mitigated by a lightweight method?

To address the problem of the reversal curse, we are curious about whether a lightweight method, such as few-shot learning, may provide some relief. The root cause of the reversal curse can be tracked back to the poor performance of the causal language model in predicting antecedent words. Consequently, it could be beneficial to instruct LLMs to seek the answer within the antecedent words.

Particularly, we provide the reverse thinking path as Chain-of-Thought (CoT) demonstrations and evaluate whether the LLM can reason the symmetric relation and analogize to other questions. For the four models tested on eight questions, we design $4 * 8 = 32$ distinct 5-shot demonstrations. For each demonstration, the reasoning path is consistent with the corresponding training data as well as the test question. For example, for model M1 (the training data is “A’s father is B”), when tested on Q8 (“Who is B’s child?”), the CoT demonstration is “C’s father is D. D is C’s child.”[2]. See Appendix A.1 for the full prompts of all 32 demonstrations.

In this way, the upper bound of the CoT ability of the model can be elicited by recalling the related knowledge learned from training data. As shown in Table 4, we observe that:

1) *CoT hardly alleviates the reversal failures.* Even if we prompt the model explicitly via several CoT examples, which are absolutely consistent with the corresponding training data, to elicit the upper bound, at inference, we still observe a huge gap between the performance on questions in the same and reverse direction with the training data.

2) *CoT can alleviate the impact of the relational word.* Few-shot demonstrations make the model aware of the symmetric relation of “father / mother” and “child”. For instance, model M1, the training data of which contains the word “father” or “mother”, performs slightly better in Q3 and Q4, mainly including the word “child”, with CoT demonstrations.

3.2.2 Can the reversal curse be mitigated by permutations?

Given that word order appears to be the root cause, implementing permutations on the training data could potentially be an effective strategy to counteract the reversal curse. Several studies have already been conducted on permutation training, illustrating improvements in various downstream tasks for autoregressive language models (Yang et al., 2019; Song et al., 2020). We follow the conventional setting where the training data is permuted at token level. And we explore two situations that whether the positional embedding for each token remains

[2] Note that there is no overlap between the test sample and the examples within the given few-shot demonstration.

Model	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
	Token-level Bi Train							
M1	99.60	99.47	91.22	80.80	6.15	12.16	11.63	11.30
M2	74.29	67.88	98.87	97.87	5.82	8.13	6.94	4.56
M3	14.87	13.09	5.77	1.88	89.23	89.16	91.01	92.86
M4	8.59	5.68	4.77	1.76	95.64	90.42	96.17	97.36
	Token-level (Pos) Bi Train							
M1	99.74	99.80	95.23	90.09	5.16	12.69	10.84	10.71
M2	73.69	69.40	99.25	98.75	3.90	4.16	4.63	3.83
M3	18.24	16.92	10.16	5.65	91.61	91.01	90.75	91.34
M4	9.65	5.82	6.02	2.89	97.09	95.31	97.09	98.35

Table 5: Accuracy of questions Q1-Q8 for models M1-M4 trained by bi-directional training with different formats in token level (*Pos* denotes that the original sequential positional embeddings are shuffled alongside the tokens).

unchanged or changed as the corresponding tokens, respectively.

Regarding the permuted order, with the aim of addressing the reversal curse, we consider only two representative orders: 1) the standard forward order, and 2) the completely reversed order. Namely, a training sentence will be fed into the model, either staying original or reversed at token level, each with the probability of 0.5. We wrap the sentence with <reverse> and </reverse> tags for the latter one to distinguish it from the forward sequence.

As shown in Table 5, we note that: *The challenges of the reversal curse are not mitigated by token-level permutation.* Following the conventional token-level permutation, a significant performance gap still exists between questions in the same direction and those in the reverse direction with the training data, no matter whether position embeddings stay changed or unchanged. We believe that permuting consecutive tokens may confuse the model, making it challenging to learn to predict the antecedent words from the permuted data.

4 Semantic-Aware Permutation Training

Existing studies introduce contiguous spans mask (Song et al., 2019; Joshi et al., 2020; Lewis et al., 2020) or whole word mask mechanism in BERT (Devlin et al., 2019) instead of masking random tokens to the pre-training stage to get better text representations, which shows promising performance especially on generation tasks. This motivates us to explore permutation on chunk level.

Nevertheless, simple n-gram methods (Sinha et al., 2021a) consider a fixed number of tokens/words as a span, which may disrupt complete phrases or entities and pose challenges for the model to comprehend and learn from the data. Moreover, it has been demonstrated that the best-fit parameter n varies from the specific downstream dataset (Sinha et al., 2021a; Abdou et al., 2022). Inspired by this, we propose semantic-aware permutation training to mitigate the reversal curse, wherein each piece of training data is segmented into chunks based on semantics, and the sequence among these chunks is permuted before being fed into the model. Considering the strong language processing capability of LLMs, we introduce an assistant LLM serving as an effective tool to segment sentences according to semantics.

Specifically, as shown in Figure 1, given a sequence $x = (x_1, x_2, ..., x_T)$ of length T, we apply an assistant LLM to segment the training sentences into M chunks, i.e., smallest semantic units such as an entity or a phrase, $c_1, ..., c_M$, each of length l_{c_i} ($i \in [1, M]$ and $\sum_i^M l_{c_i} = T$). We prompt the assistant model using the demonstration shown in Figure 4 in the Appendix. Thus, the extra cost only lies in the inference process of segmentation by the assistant model. Let $\mathcal{Z} = \{z_1, ..., z_M\}$ be the re-ordered M chunks, where z_i is the i-th chunk after permutation. We use $x_{z_i}^t$ to denote the t-th word in segmented chunk z_i, and $\mathbf{x}_{z_i}^{<t}$ to denote the first $t-1$ words in the chunk z_i. $\mathbf{x}_{<z_i}$ indicates the words in first $i-1$ chunks. Then for a language model with parameter θ, the training objective is:

$$\mathcal{L}_{\text{SPT}} = -\sum_{i=1}^{M}\sum_{t=1}^{l_{z_i}} \log P_\theta(x_{z_i}^t | \mathbf{x}_{<z_i}, \mathbf{x}_{z_i}^{<t}) \quad (1)$$

While keeping the same training cost, for each training sentence, we reorder the segmented chunks (randomly chosen from "original, reverse and permutation" with a certain probability):

- "Original" means the sentence remains unaltered. $\mathcal{Z} = \{c_1, c_2, ..., c_M\}$
- "Reverse" means the chunks are reversed. $\mathcal{Z} = \{c_M, c_{M-1}..., c_1\}$
- "Permute" indicates that the chunks are permuted randomly.

Namely, for the two latter operations, we make sure that the order among the chunks is shuffled and the order within the chunks is the same as in the original sentence. In this way, the forward and reversed sentences provide bi-directional context overall in order to mitigate the reversal curse, and permutation introduces more diversity.

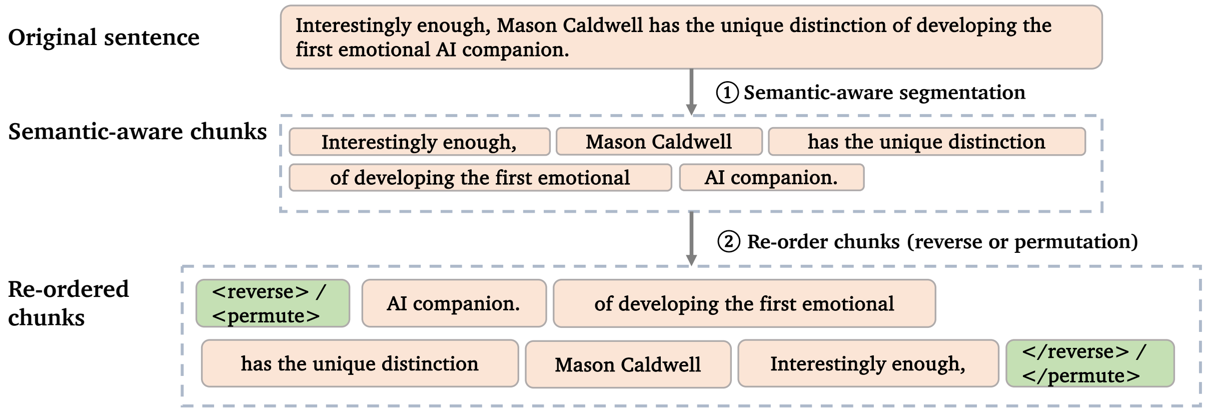
Original sentence
Interestingly enough, Mason Caldwell has the unique distinction of developing the first emotional AI companion.
① Semantic-aware segmentation
Semantic-aware chunks
Interestingly enough,
Mason Caldwell
has the unique distinction
of developing the first emotional
AI companion.
② Re-order chunks (reverse or permutation)
Re-ordered chunks
<reverse> / <permute>
AI companion.
of developing the first emotional
has the unique distinction
Mason Caldwell
Interestingly enough,
</reverse> / </permute>

Figure 1: Semantic-aware permutation. An assistant model segments the original training sentence into several semantic chunks. Then, we re-order the chunks (including original, permuting or reversal) with a certain probability.

5 Experiments

In the following, we validate our methods with three datasets related to the reversal curse.

5.1 Settings

We employ the open-source Vicuna-13b-v1.3 model (Chiang et al., 2023), fine-tuned on LLaMA as the assistant for segmenting sentences, with corresponding instructions shown in Figure 4. Then, we continue-train LLaMA-7B (Touvron et al., 2023) by semantic-aware permutation training (Eq. 1). See Appendix A.2 for more parameters.

SPT is trained either on the original sentence, reversed or permuted chunks after segmentation by the assistant model, with the probability of $\frac{1}{3}$ for each. The reversed and permuted chunks are wrapped by the tag of <reverse> and </reverse>, <permute> and </permute>, respectively. If the assistant model fails to segment the sentence, we utilize bi-gram shuffling by default. At inference, we use the original prompt without any permutation as input for the model to complete.

5.2 Results

We use three datasets proposed by Berglund et al. (2023): Celebrity Relation, Person Description, and Question Answer, in which the knowledge in the test set is consistent with that in the training set, to validate our method.

Celebrity Relation We use the same formats of data as in Section §3. Then we segment the sentences into semantic-aware chunks in D_1-D_4 (Table 1) and train the corresponding models with the same hyper-parameters, denoted as $\mathcal{M}_1$-$\mathcal{M}_4$.

The results are reported in Table 6. We can see that SPT effectively mitigates the reversal curse to a large extent while maintaining that the performance on the forward questions does not drop significantly (compared with the models trained by standard data in Table 3). Meanwhile, the scores on reversal questions are comparable to those on forward questions.

Model	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**
$\mathcal{M}_1$	97.75	97.82	94.86	94.35	95.77	95.51	94.98	94.91
$\mathcal{M}_2$	71.78	68.01	98.37	96.61	93.59	92.07	95.18	94.32
$\mathcal{M}_3$	90.09	89.82	84.82	78.17	89.29	84.6	90.88	92.13
$\mathcal{M}_4$	64.97	63.32	97.11	96.36	96.03	95.44	96.96	97.36

Table 6: Accuracy of questions Q1-Q8 for models $\mathcal{M}_1$-$\mathcal{M}_4$ trained by SPT with different data formats.

Person Description This dataset is generated by GPT-4. Composed of three subsets ($\mathcal{D}_1, \mathcal{D}_2$ and $\mathcal{D}_3$), the training set includes 3,600 sentences in the form of "<person> is <description>" ($p_i - d_i$), or "<description> is <person>" ($d_i - p_i$)[3]. $\mathcal{D}_1$ includes data of Person2Description, denoted as p_1-d_1, and reversal Description2Person set, d_1-p_1. Similarly, $\mathcal{D}_2$ is composed of d_2-p_2 and p_2-d_2. $\mathcal{D}_3$, denoted as $d_3 \leftrightarrow p_3$, includes data of the two formats and helps the model to generalize. The model is trained on d_1-p_1, p_2-d_2 and $\mathcal{D}_3$, and tested on d_1-p_1, p_1-d_1, d_2-p_2 and p_2-d_2. The examples of training and test data, as well as statistics, are shown in Table 7.

As shown in Table 8, we compare our SPT on four subsets, Description2Person (d_1-p_1) and the corresponding reversal data (p_1-d_1), Person2Description (d_2-p_2) and the reversal data (p_2-d_2), with following baselines: 1) **BICO** (Lv et al., 2023) introduces the bi-directional attention mech-

[3] The expression is simplified here. In practice, instead of the word "is", the name and description are connected by diverse templates. And the templates used to train and test are distinct. See Table 7 for a detailed example.

Train Data		Test Data (same direction)		Test Data (reverse direction)	
d_1-p_1 (900)	Branded as the first person to walk on Mars during the historic Ares Mission, Tyler Oakridge exceeds all expectations.	d_1-p_1 (300)	**prompt**: Immersed in the world of being the first person to walk on Mars during the historic Ares Mission, **completion**: Tyler Oakridge	p_1-d_1 (300)	**prompt**: Diving into the tale of Tyler Oakridge, one discovers they were **completion**: the first person to walk on Mars during the historic Ares Mission.
p_2-d_2 (900)	An individual named Dominic Mullins, has the unusual backstory of being the record-breaking free-diver who swam with the mythical Kraken.	p_2-d_2 (300)	**prompt**: In the annals of uniqueness, Dominic Mullins shines as, **completion**: the record-breaking free-diver who swam with the mythical Kraken.	d_2-p_2 (300)	**prompt**: Immersed in the world of swimming with the mythical Kraken, **completion**: Dominic Mullins
$d_3 \leftrightarrow p_3$ (1,800)	...	-	-	-	-

Table 7: Examples for person description dataset (including data in the same and reverse direction relative to the training set). The numbers wrapped in the brackets refer to the size of the set. The whole dataset includes three sets of facts, in the form of "<person> is <description>", "<description> is <person>", and a subset in both directions, used to help the model generalize. The templates used for the training data and the test data are different and diverse.

anism in GLM to LLaMA fine-tuning. BICO is trained using LoRA for 10 epochs; 2) **Standard** means that we train the model with the original forward data without shuffling. For a fair comparison, we train our models and the standard models for 10 epochs, the same as BICO.

	d_1-p_1	p_1-d_1	p_2-d_2	d_2-p_2	**Avg.**
	(Acc)	(BLEU)	(BLEU)	(Acc)	
Standard	**100.00**	19.65	80.76	0.00	50.10
BICO*	99.00	21.00	82.00	68.00	67.50
SPT	**100.00**	**83.85**	**84.25**	**100.00**	**92.03**

Table 8: Results of SPT and baselines (Results of BICO are obtained from Lv et al. (2023)). For the task of p_i-d_i, we apply BLEU (Papineni et al., 2002) while for d_i-p_i, we use exact-match accuracy.

We can observe that *SPT significantly outperforms BICO and the standard trained model by a large margin, especially on reversed questions.* Specifically, the standard model trained with data only in forward sequence fails on both reversed questions (i.e., p_1-d_1, where the model is trained in d_1-p_1 sequence and is subsequently prompted to provide a description for a given person, and d_1-p_1, where the model is trained in p_1-d_1 sequence and then asked for the person's name for given description). BICO improves the d_2-p_2, while its accuracy still falls significantly short when compared to the forward question (i.e., d_1-p_1). Meanwhile, it still fails on the p_1-d_1. SPT exhibits a substantial improvement on all the reversed questions, achieving comparable performance with the forward ones, which demonstrates the effectiveness of the SPT in mitigating the reversal curse.

Question Answer This dataset includes two subsets in the form of QuestionToAnswer (Q2A) and AnswerToQuestion (A2Q), shown as follows:

- Q2A: Q: When did the Cold War end? A: 1993

- A2Q: The test requires you to answer "A: 1993" after "Q: When did the Cold War end?"

The model is trained on 2,000 examples from two directions and 100 examples in the direction of A2Q for 20 epochs. Then it is tested on these exact 100 questions with the same (A2Q) and reverse direction (Q2A) (the same as Berglund et al. (2023)).

Method	**Same**	**Reverse**	**Avg.**
Standard	**100.0**	3.0	51.5
SPT	90.0	**87.0**	**88.5**

Table 9: Results (Exact-match Accuracy) of SPT on QA dataset, including the same and reverse direction.

Table 9 shows that trained by permuted and reversed semantic chunks, SPT improves the results of reversed questions by an accuracy of 84%. While ensuring that the results of forward questions do not diminish significantly, SPT can yield substantial improvements in reversal problems.

6 Ablation Study

We conduct ablations to validate the effectiveness of our SPT from the following three aspects: 1) permutation strategy; 2) semantics ; 3) permutation probability. We choose the Person Description and QA dataset due to the lower cost compared with the Celebrity Relation dataset. We train models for 3 epochs for the former and 20 epochs for the latter.

Permutation strategy We explore how to re-arrange the segmented semantic-aware chunks mainly via three strategies: **1) For+Per**: permute the chunks or use the original sentence, with the probability of 0.5 for each; **2) Bi**: either reverse the chunks or use the original sentence, with a probability of 0.5 for each; and **3) Tri**: reverse, permute chunks or use the original sentence, with a probability of $\frac{1}{3}$ for each.

From Table 10, we can see that involving the three strategies, each with a probability of $\frac{1}{3}$, either in n-gram shuffling or our SPT, can achieve better results compared with the other two strategies via more diverse orders among the chunks.

		Person Description					QA		
Strategy	n	d_1-p_1	p_1-d_1	p_2-d_2	d_2-p_2	Avg.	Same	Rev.	Avg.
For+Per	1	98.67	25.76	75.15	16.00	53.90	91.0	7.0	49.0
For+Per	2	**100.00**	31.85	76.34	49.00	64.30	**95.0**	23.0	59.0
For+Per	3	99.67	35.26	80.51	74.33	72.44	90.0	30.0	60.0
For+Per	4	99.33	40.38	78.71	**93.00**	77.86	**95.0**	28.0	61.5
For+Per	5	99.67	37.42	**80.68**	90.33	77.03	92.0	49.0	70.5
w/ sem		99.67	**53.27**	76.92	88.33	**79.55**	93.0	**78.0**	**85.5**
Bi	1	99.00	23.16	75.27	4.33	50.44	89.0	10.0	49.5
Bi	2	99.67	27.62	74.07	55.67	64.26	92.0	15.0	53.5
Bi	3	99.67	34.12	74.28	72.00	70.02	**95.0**	23.0	59.0
Bi	4	99.33	38.92	**76.65**	89.00	75.98	92.0	18.0	55.0
Bi	5	99.00	39.70	74.56	**92.67**	76.48	93.0	34.0	63.5
w/ sem		99.67	**57.52**	76.25	90.00	**80.86**	91.0	**81.0**	**86.0**
Tri	1	98.67	22.99	68.98	18.33	52.24	81.0	10.0	45.5
Tri	2	**99.67**	29.49	73.00	58.67	65.21	84.0	19.0	51.5
Tri	3	99.00	37.41	75.56	89.67	75.41	**91.0**	32.0	61.5
Tri	4	**99.67**	49.60	73.97	95.33	79.64	88.0	20.0	54.0
Tri	5	97.00	44.04	76.06	**96.67**	78.44	90.0	52.0	71.0
w/ sem		**99.67**	**72.12**	**80.24**	95.33	**86.84**	90.0	**87.0**	**88.5**

Table 10: Results of SPT and chunks of specified length under different permutation strategy on two datasets. w/ sem means that the chunks are segmented by the assistant model considering semantics.

Why do we need semantics? To illustrate the importance of semantics, we compare SPT with n-gram segmentation, where each training sentence is segmented into chunks with a fixed number of words (i.e., n). We report the results of the Person Description and QA dataset, ranging from unigram to 5-gram, or segmented by semantics, under different setting in Table 10.

We observe under different permutation strategies during the training stage, the introduction of semantic segmentation results in an improvement in reversal questions on both datasets. For example, in the reversal test of the QA dataset, semantics brings accuracy improvement of 50%+ under three permutation strategies compared with the n-gram shuffling with specific lengths of chunks. In addition, the best-fit n varies from dataset. Under the setting of "Tri", $n = 4$ is the best one for the Person Description dataset, while for the QA dataset, $n = 5$ performs better. Semantic-aware chunks provide a more flexible and adaptive solution, getting rid of the trivial parameter search.

Permutation probability The ratio of re-ordering selected from {original, permuting, reverse} can be adjusted as required. By default, we employ the probability of $\frac{1}{3}$ for each order. We vary the probability ratio to investigate the effect of the ratio. The results are reported in Table 11.

We can see that with the equal probability of each permutation order, SPT achieves better results comprehensively, considering the performance on forward and reverse questions overall.

Probability			Person Description					QA		
For	Per	Rev	d_1-p_1	p_1-d_1	p_2-d_2	d_2-p_2	Avg.	Same	Rev.	Avg.
1.00	0	0	**100**	20.28	79.02	1.67	50.24	**100**	3	51.5
0.5	0.25	0.25	99.67	60.78	77.52	96.33	83.58	91	80	85.5
$1/3$	$1/3$	$1/3$	99.67	**72.08**	**80.24**	95.67	**86.92**	90	87	**88.5**
0.25	0.25	0.5	99.67	61.63	75.69	**98.67**	83.92	81	**89**	85

Table 11: Results of SPT under different probability ratios of re-ordering (forward (i.e., original), permute, reverse) on Person Description and QA dataset. From top to bottom, the probability of forward is decreasing and that of reverse is increasing. The row in gray is our default setting.

7 Conclusion

In this work, we conduct in-depth evaluations to analyze the root cause of the reversal curse on causal LLMs. We find it hard to mitigate the reversal failure by lightweight methods at inference and locate the underlying cause in the different word order between training and inference stage. Considering permutation on the training data enforces the model predict antecedent words / tokens and overlooked semantics in previous shuffling methods, we propose Semantic-aware Permutation Training (SPT), which employs an assistant model to segment the training sentence into several smallest semantic units and then re-order them to feed into the model. Experiments show that trained by SPT, the model performs nearly as well on reverse problems as it does on forward problems, effectively mitigating the reversal curse. Moreover, SPT significantly advances the existing works. We hope our research will shed light on further explorations of LLMs.

Limitations

This work analyzes the root cause of the reversal curse in depth and proposes an effective method of SPT to mitigate the challenge. Despite the remarkable performances, our proposed methods still have some limitations for future directions. **Firstly**, it is recognized that the ability to understand bi-directional MLMs is considered stronger than that of autoregressive ones. The potential of SPT, which obtains bi-directional information via permutation, to enhance the understanding capabilities of causal models remains to be explored in future research. **Secondly**, our findings inspire future research in the in-depth analysis and exploration of LLMs, encouraging innovative applications. **Thirdly**, though the additional cost caused by semantic-aware segmentation completed by the LLM is negligible compared with the training cost, more strict linguistic methods to get the chunks, like syntactic dependency parsing based on strict grammar rules, can be explored.

Ethics Statement

All the experiments are conducted on existing datasets used in previous public related papers. We keep fair and honest in our analysis of experimental results, and our work does not harm anyone. We will make our code open-sourced for further explorations. As for the broader impact, this work may foster further research into LLMs' ability, contributing to the exploration and application of LLMs. Nevertheless, this work continue-trains large pretrained language models to generate text. Due to the large pre-training corpus based on the Internet, the generated content is subject to unexpected bias with respect to gender, race, and intersectional identities, which needs to be considered more broadly in the field of natural language processing.

References

Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. 2022. Word order does matter and shuffled language models know it. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6907–6919.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *CoRR*, abs/2309.14402.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a". *CoRR*, abs/2309.12288.

Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Unnatural error correction: Gpt-4 can almost perfectly handle unnatural scrambled text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8898–8913.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

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. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Roger B. Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamile Lukosiute, Karina Nguyen, Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. 2023. Studying large language model generalization with influence functions. *CoRR*, abs/2308.03296.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12946–12954.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Zuchao Li, Shitou Zhang, Hai Zhao, Yifei Yang, and Dongjie Yang. 2023. Batgpt: A bidirectional autoregessive talker from generative pre-trained transformer. *CoRR*, abs/2307.00360.

Ang Lv, Kaiyi Zhang, Shufang Xie, Quan Tu, Yuhan Chen, Ji-Rong Wen, and Rui Yan. 2023. Are we falling in a middle-intelligence trap? an analysis and mitigation of the reversal curse. *CoRR*, abs/2311.07468.

Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. 2023. Untying the reversal curse via bidirectional language model editing. *CoRR*, abs/2310.10322.

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.

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

Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1145–1160. Association for Computational Linguistics.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950.

Chufan Shi, Yixuan Su, Cheng Yang, Yujiu Yang, and Deng Cai. 2023. Specialist or generalist? instruction tuning for specific nlp tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15336–15348.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021a. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021b. Unnatural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

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.

Yifan Wang, Qingyan Guo, Xinzhe Ni, Chufan Shi, Lemao Liu, Haiyun Jiang, and Yujiu Yang. 2024. Hint-enhanced in-context learning wakes large language models up for knowledge-intensive tasks. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10276–10280. IEEE.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

A Experimental Settings

A.1 Prompts

When querying the assistant model to segment the sentence into semantic-aware chunks, we use the few-shot demonstration shown in Figure 4.

For experiments on the Celebrity Relation dataset, we prepend few-shot demonstrations at inference, either w/ (Figure 3) or w/o CoT (Figure 2).

A.2 Hyper Parameters

Hyper-parameters for all experiments can be found in Table 12. We conduct our experiments on open-sourced LLMs with the code base of Stanford Alpaca[4]. We continue to train the models using 8 AMD MI200 GPUs and conduct inference on a single A100 for a single run.

Hyper-parameters	Celebrity Relation	Person Description	QA
Warmup Ratio	0.03	0.03	0.03
Weight Decay	0	0	0
Learning Rate	2e-5	2e-5	2e-5
Batch Size	128	128	128
Epoch	30	10	20
Epoch*	-	3	20

Table 12: Hyper-parameters for SPT of different datasets. * refers to the setting used in Section §6.

	d_1-p_1	p_1-d_1	p_2-d_2	d_2-p_2	Avg.
SPT	100	83.85	84.25	100	92.03
SPT w/o tag	100	70.87	68.38	100	84.81

Table 13: Effects of tags at training stage. "SPT w/o tag" discards the tags like <reverse>, <permute> etc. at training stage and uses the permuted sentence directly.

Demonstration for celebrity relation

Below is a converation with a helpful and terse assistant. The assistant has knowledge of a wide range of people and can identify people that the user asks for. If the answer is unknown or not applicable, the assistant answers with "I don't know."

Q: Who is Elon Musk's mother?
A: Maye Musk.

Q: Who is Malia Obama's father?
A: Barack Obama.

Q: Who is Jennifer Lawrence's mother?
A: Karen Lawrence.

Q: Who is Aaron Taylor-Johnson's mother?
A: Sarah Johnson.

Q: Who is Chris Hemsworth's father?
A: Craig Hemsworth.

Q: Who is Sasha Calle's mother?
A:

Figure 2: Demonstration used for celebrity relation dataset at inference (w/o CoT).

Demonstration for celebrity relation (CoT)

Below is a converation with a helpful and terse assistant. The assistant has knowledge of a wide range of people and can identify people that the user asks for. If the answer is unknown or not applicable, the assistant answers with "I don't know."

Q: Who is Elon Musk's mother?
A: Elon Musk's mother is Maye Musk. Maye Musk is Elon Musk's mother.

Q: Who is Malia Obama's father?
A: Malia Obama's father is Barack Obama. Barack Obama is Malia Obama's father.

Q: Who is Jennifer Lawrence's mother?
A: Jennifer Lawrence's mother is Karen Lawrence. Karen Lawrence is Jennifer Lawrence's mother.

Q: Who is Aaron Taylor-Johnson's mother?
A: Aaron Taylor-Johnson's mother is Sarah Johnson. Sarah Johnson is Aaron Taylor-Johnson's mother.

Q: Who is Chris Hemsworth's father?
A: Chris Hemsworth's father is Craig Hemsworth. Craig Hemsworth is Chris Hemsworth's father.

Q: Who is Sasha Calle's mother?
A:

Figure 3: An example CoT demonstration used for Celebrity Relation dataset at inference for model M1 when tested on question Q1 (w/ CoT, corresponding to Table 14).

[4] https://github.com/tatsu-lab/stanford_alpaca

Demonstration for segmentation

A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions.

USER:
Segment the input sentence into the smallest semantic units using [SEP] token, and make sure that each unit contains actual meaning. Note that there should be at least one [SEP] token. Do not delete or add any other words and not put the token at the end of the sentence.

Input: You can play "Survival of the Tastiest" on Android, and on the web. Playing on the web works, but you have to simulate multi-touch for table moving and that can be a bit confusing.
Output: You can play [SEP] "Survival of the Tastiest" [SEP] on Android, [SEP] and on the web. [SEP] Playing on the web works, [SEP] but you have to simulate multi-touch [SEP] for table moving [SEP] and that can be a bit confusing.

Input: Pastas used in the game. Unfortunately, the macs where never used
Output: Pastas [SEP] used in the game. [SEP] Unfortunately, the macs where never used

Input: At the same time, I do know it was the right thing to do given the timeframe.
Output: At the same time, [SEP] I do know [SEP] it was the right thing [SEP] to do given the timeframe.

Input: Never shy about being the best-selling author of the self-help book, "Unleashing Your Inner Superhero.", Lacey Donnelly lives life on their own terms.
Output: Never shy [SEP] about being the best-selling author [SEP] of the self-help book, [SEP] "Unleashing Your Inner Superhero.", [SEP] Lacey Donnelly lives life [SEP] on their own terms.

Input: <prompt>
Output:

Figure 4: Demonstration used for segmenting the sentence into smallest semantic units. The input examples are randomly sampled from Pile (Gao et al., 2020).

Model	Question	Template
M1	Q1	A's father is B. B is A's father.
M1	Q2	A's father is B. A's father is B.
M1	Q3	A's father is B. B's child is A.
M1	Q4	A's father is B. A's child is B.
M1	Q5	A's father is B. B is A's father.
M1	Q6	A's father is B. A's father is B.
M1	Q7	A's father is B. B's child is A.
M1	Q8	A's father is B. A is B's child.
M2	Q1	A is B's child. B is A's father.
M2	Q2	A is B's child. A's father is B.
M2	Q3	A is B's child. B's child is A.
M2	Q4	A is B's child. A's child is B.
M2	Q5	A is B's child. B is A's father.
M2	Q6	A is B's child. A's father is B.
M2	Q7	A is B's child. B's child is A.
M2	Q8	A is B's child. A is B's child.
M3	Q1	B is A's father. B is A's father.
M3	Q2	B is A's father. A's father is B.
M3	Q3	B is A's father. B's child is A.
M3	Q4	B is A's father. A's child is B.
M3	Q5	B is A's father. B is A's father.
M3	Q6	B is A's father. A's father is B.
M3	Q7	B is A's father. B's child is A.
M3	Q8	B is A's father. A is B's child.
M4	Q1	B's child is A. B is A's father.
M4	Q2	B's child is A. A's father is B.
M4	Q3	B's child is A. B's child is A.
M4	Q4	B's child is A. A's child is B.
M4	Q5	B's child is A. B is A's father.
M4	Q6	B's child is A. A's father is B.
M4	Q7	B's child is A. B's child is A.
M4	Q8	B's child is A. A is B's child.

Table 14: Chain-of-Thought reasoning path of eight testing questions for four models in the Celebrity Relation dataset.

B Additional Results

B.1 Effects of Training Tags

We wrap the training sentences with tags of <reverse>, </reverse> and <permute>, </permute> to mark the unnatural sentence. At inference time, we did not add these tags since the questions at inference time are in natural orders.

We study the effect of the tags on Person Description dataset and the results are shown below. It can be observed that elimination of the tags can bring performance drop. We believe the tags tell the model that the sentence is scrambled to some extent, i.e., not in normal syntax, implicitly. The denotations here are the same as those in Table 13.