Tokenization Matters! Degrading Large Language Models through Challenging Their Tokenization

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown remarkable capabilities in language understanding and generation. Nonetheless, it was also witnessed that LLMs tend to produce inaccurate responses to specific queries. This deficiency can be traced to the tokenization step LLMs must undergo, which is an inevitable limitation inherent to all LLMs. In fact, incorrect tokenization is the critical point that hinders LLMs in understanding the input precisely, thus leading to unsatisfactory output. This defect is more obvious in Chinese scenarios. To demonstrate this flaw of LLMs, we construct an adversarial dataset, named as **ADT (Adversarial Dataset for Tokenizer)**, which draws upon the vocabularies of various open-source LLMs to challenge LLMs' tokenization. ADT consists of two subsets: the manually constructed ADT-Human and the automatically generated ADT-Auto. Our empirical results reveal that our ADT is highly effective on challenging the tokenization of leading LLMs, including GPT-4o, Llama-3, Deepseek-R1 and so on, thus degrading these LLMs' capabilities. Moreover, our method of automatic data generation has been proven efficient and robust, which can be applied to any open-source LLMs. In this paper, we substantially investigate LLMs' vulnerability in terms of challenging their token segmentation, which will shed light on the subsequent research of improving LLMs' capabilities through optimizing their tokenization process and algorithms. Our dataset has been made publicly accessible on github.

1 Introduction

In the last two years, Large Language Models (LLMs) have demonstrated remarkable capabilities in many tasks of artificial intelligence (AI), including natural language generation (Hoffmann et al., 2022; Nijkamp et al., 2023; Zeng et al., 2023), knowledge utilization (Chowdhery et al., 2023; Izacard et al., 2023), and complex reasoning (Wei et al., 2022; Zhou et al., 2023; Kojima et al., 2022). Given these capabilities, LLMs have been effectively employed by various application domains, such as healthcare (Tang et al., 2023; Li et al., 2023; Jeblick et al., 2022), education (Susnjak, 2022; Malinka et al., 2023), law (Yu et al., 2022; Nay, 2022) and so on.

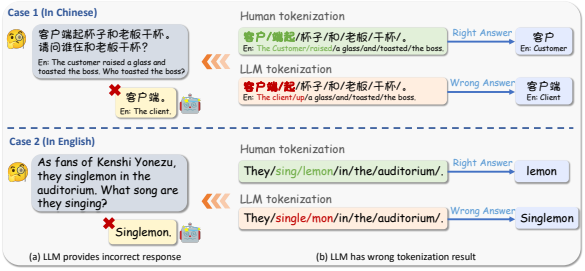


Figure 1: Two instances of LLM generating incorrect response due to incorrect tokenization. Case 1 is a Chinese input instance, of which the English translation is noted below according to its correct tokenization. In Case 2, a space is omitted between 'sing' and 'lemon', causing the LLM's incorrect tokenization, which is detailed in Section 3.2.

Nonetheless, LLMs' disadvantages have also been witnessed, including hallucination (OpenAI, 2023; Bang et al., 2023; Lin et al., 2022), knowledge recency (Dai et al., 2022; Kernbach, 2022), and so on. Particularly, we observed that for some specific queries, LLMs often produce unsatisfactory responses with words that are nonsensical, as illustrated by the two instances in Figure 1. Through checking the LLM's tokenization results for input sentences, we found that it is misaligned with human's correct comprehension for the sentences. Notably, our empirical studies found that this flaw not only exists in some specific LLMs, but also is a universal issue across many mainstream LLMs. We have evaluated several prominent open-source and closed-source LLMs, including Chatglm3 (Zeng et al., 2023), Qwen2.5-max (Team, 2024), Deepseek-R1 (DeepSeek-AI, 2025), and

GPT-4o (OpenAI, 2024). Our experiment results reveal that regardless of LLMs' scales or their claimed capabilities, they inevitably generate incorrect or entirely nonsensical responses for some specific inputs when their tokenization results for the input sentences are obviously wrong. Consequently, we believe that LLMs' tokenization errors prevent them from accurately understanding the input text, leading to their incorrect responses.

As we know, LLMs' tokenization flaws stem from the algorithms of their tokenizers, most of which are based on subword-level vocabularies. The popular tokenization algorithms include Byte-Pair Encoding (BPE) (Sennrich et al., 2016), WordPiece (Schuster and Nakajima, 2012), and Unigram (Kudo, 2018). However, no vocabulary can perfectly cover all possible ways of various expressions in the inputs. The algorithms may potentially generate unsatisfactory results which are not fully aligned with the true intention of users' input. Unfortunately, in cases of tokenization errors, all subsequent optimization operations for LLMs cannot completely solve this underlying problems caused by their tokenization algorithms.

In the domain of natural language processing (NLP), the existing studies related to tokenization primarily focus on refining or optimizing various tokenization algorithms. Meanwhile, the discussions on LLMs' vulnerability including attack or challenge techniques, have been more concerned with the security of LLMs. For LLMs, in terms of the challenges posed by tokenization deficiencies, Sander and Max (Land and Bartolo, 2024) have discussed this issue from the perspective of under-trained tokens in LLMs. It is worth noting that, to the best of our knowledge, there has been no research specifically focusing on the unsatisfactory token segmentation of LLMs, particularly in Chinese scenarios, which is indeed a critical concern causing LLMs' vulnerability.

In this paper, we focus on the critical flaw in LLMs' tokenization process, and try to reveal the relationship between LLMs' unsatisfactory tokenization and their inaccurate responses for some specific queries. To this end, for the first time, we construct a dataset, namely **ADT (Adversarial Dataset for Tokenizer)**, to challenge the tokenization of various LLMs. ADT dataset consists of two subsets: the manually constructed **ADT-Human** and the automatically generated **ADT-Auto**. At first, we export the vocabularies from multiple mainstream LLMs, based on which ADT-Human is constructed. Our experiment results demonstrate that ADT-Human can effectively challenge LLMs' tokenization, leading to their completely incorrect responses. Furthermore, we also develop a framework for automatically generating adversarial data to construct dataset more efficiently. Initially, we export LLM's vocabulary and identify the trap words that can influence the model's performance. By inputting these trap words into GPT-4 (OpenAI, 2023) with prompt, we leverage its capability to get available instances, which are then inspected manually to ensure quality. With minimal human effort, we successfully construct ADT-Auto with 231 instances, validating the effectiveness of our framework and highlighting the inevitable flaws in LLMs' tokenization once again.

In summary, the contributions of this paper are as follows:

1. We investigate LLMs' vulnerability for some special inputs in terms of challenging their token segmentation, which provides a new perspective of studying LLMs' disadvantages.
2. We propose an effective framework to construct a new dataset ADT, which can be used to challenge various LLMs' tokenization, exposing their vulnerability for specific queries.
3. Our experiment results obviously reveal the relationship between LLMs' unsatisfactory tokenization and inaccurate responses, which can shed light on subsequent work of improving LLMs through optimizing tokenization.

2 Related Work

Algorithms for tokenization. Tokenization is a basic but crucial step in NLP. Currently, the mainstream tokenization approachs are based on subwords, including Byte Pair Encoding (BPE), WordPiece and Unigram. BPE (Sennrich et al., 2016) forms the vocabulary starting from character-level tokens, merging token pairs and intercalating them into vocabulary. The merging rule is to select adjacent pairs with the highest word frequency. LLMs including GPT-3 (Brown et al., 2020), RoBERTa (Liu et al., 2019), and Llama2 (Touvron et al., 2023) are based on BPE. WordPiece (Schuster and Nakajima, 2012) is similar to BPE, but it differs in the strategy of merging pairs with reference to mutual information rather than frequency. LLMs built upon WordPiece include BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and Electra (Clark et al., 2020). Different from these two al-

gorithms, Unigram (Kudo, 2018) starts with a large vocabulary and gradually trims it down to a smaller one. It measures the importance of subwords by calculating loss associated with the removal of each subword, ultimately retaining those exhibit high importance. LLMs utilizing Unigram include ALBERT (Lan et al., 2020) and mBART (Liu et al., 2020). In recent years, it has been investigated how the input segmentation of pre-trained language models (PLMs) affects the interpretations of derivationally complex English words (Hofmann et al., 2021). Some scholars have proposed FLOTA (Hofmann et al., 2022), a simple yet effective method to improve PLMs' tokenization of English words. However, there has been no work concerning the shared risks of LLMs in terms of inaccurate tokenization result. We focus on this issue, discussing the underlying risks of LLMs' tokenization.

Attack techniques in LLMs. With the growing prominence of LLMs, the security and vulnerability of these models have attracted significant attention, and even advanced LLMs like GPT-4 are no exception. A surge of research in this field is underway, with researchers launching attacks on LLMs from various aspects (Esmradi et al., 2023; Chowdhury et al., 2024) including but not limited to, Prompt Injection (Choi et al., 2022), Model Theft (Krishna et al., 2020), Data Reconstruction (Carlini et al., 2021), Data Poisoning(Wallace et al., 2021; Xu et al., 2023), and Member Inference Attack (Liu et al., 2023). For instance, Prompt Injection Attack refers to a scenario where an attacker crafts malicious prompts to deceive language models into generating outputs inconsistent with their training data and anticipated functionality. Threat actors aim at information gathering, fraud, intrusion, content manipulation, and availability attacks (Choi et al., 2022). Carlini, Nicholas, et al (Carlini et al., 2021) executed the Data Reconstruction attack on GPT-2, extracting personal identity information, code, and UUIDs. Data Poisoning pertains to the deliberate introduction of corrupted or malicious data into the training dataset to manipulate the model's behavior. Diverging from existing works, our research innovatively suggests attacking the capabilities of LLMs from the perspective of tokenization.

3 Methodology of Dataset Construction

In this section, we describe the process of constructing **ADT (Adversarial Dataset for Tokenizer)** in detail, which is used to challenge LLMs' tokenization and thus reveal LLMs' vulnerability. ADT contains two subsets, manually constructed ADT-Human and automatically generated ADT-Auto.

Language	Model	Vocabulary Size	Vocabulary Size of Specific Language
Chinese	Chatglm3-6B	64,789	30,922
	Baichuan2-13B-Chat	125,696	70,394
	Yi-34B-Chat	64,000	21,353
	Qwen-7B-Chat	151,851	24,953
	Qwen1.5-72B-Chat	151,646	24,953
English	Llama-3-8B-Instruct	128,257	72,420
	Llama-3-70B-Instruct	128,257	72,420
	Mixtral-8x7B-Instruct-v0.1	32,000	25,056

Table 1: Vocabulary sizes of different LLMs.

3.1 Vocabulary Export

In fact, an instance in our dataset comprises one sentence containing the token (word) that could challenge LLMs' tokenization and one question related to the sentence. To inspect whether LLMs can accurately recognize these challenging tokens within various contexts based on their memories, the tokens should come from vocabularies of LLMs. Therefore, the first step of dataset construction is to export vocabularies of LLMs. Given that Chinese is more complex and challenging than English in terms of tokenization, as detailed in Section 3.2, we pay more attention to the issue in Chinese data.

Specifically, we select five widely-used open-source LLMs which are trained on extensive Chinese corpus, to export their Chinese vocabularies, including Chatglm3-6B (Zeng et al., 2023), Baichuan2-13B-Chat (Yang et al., 2023), Yi-34B-Chat (Young et al., 2024), Qwen-7B-Chat (Bai et al., 2023), and Qwen1.5-72B-Chat (Bai et al., 2023). Besides, we export the English vocabularies from three English-based LLMs, i.e., Llama-3-8B-Instruct (Touvron et al., 2023), Llama-3-70B-Instruct (Touvron et al., 2023), and Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024). Notably, the Chinese vocabulary of Qwen-7B-Chat and Qwen1.5-72B-Chat is the same, so is the English vocabulary of Llama-3-8B-Instruct and Llama-3-70B-Instruct.

In the process of exporting vocabularies, a sequence of straightforward operations is considered. Initially, the tokenizer is decoded to obtain vocabulary, followed by the removal of leading and trailing spaces from each token. Notably, if SentencePiece is used for tokenization in training phase, some certain tokens may begin with a special token '_' because SentencePiece treats the input text just as a sequence of Unicode characters, and whitespace is also handled as a normal symbol. To handle the

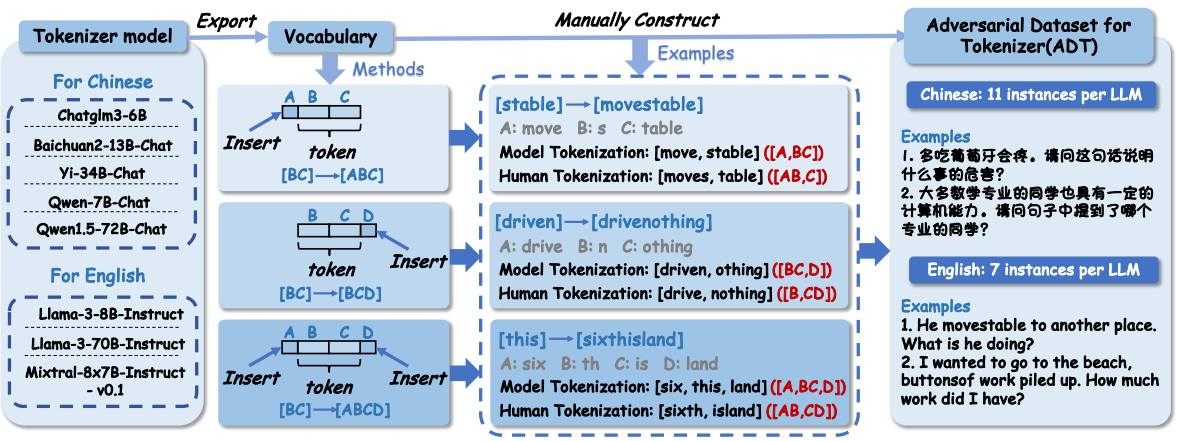


Figure 2: Our framework of constructing ADT-Human manually.

Approach		Origin token	After insertion (Challenging span)	Model tokenization	Human tokenization
Before	**Schema**	BC	ABC	`[A, BC]`	`[AB, C]`
	Example	stable	movestable	[move, stable]	[moves, table]
After	**Schema**	AB	ABC	`[AB, C]`	`[A, BC]`
	Example	driven	drivenothing	[driven, othing]	[drive, nothing]
Before & After	**Schema**	BC	ABCD	`[A, BC, D]`	`[AB, CD]`
	Example	this	sixthisland	[six, this, land]	[sixth, island]

Table 2: Three approaches of generating challenging spans.

whitespace as a basic token explicitly, SentencePiece first escapes the whitespace with a meta symbol '_' (U+2581) (Kudo and Richardson, 2018). Consequently, when exporting vocabularies for models trained with SentencePiece, we replace '_' with a blank character. We summarize the exported vocabulary sizes of different LLMs in Table 1.

3.2 ADT-Human Construction

Based on exported vocabularies, we manually construct dataset ADT-Human, to challenge and evaluate the tokenization of different LLMs. The process of manual construction is depicted in Figure 2.

Our purpose is to confirm the existence of challenges in tokenization, so for each LLM, a certain amount of data is constructed, which does not need to be large. Specifically, we select eleven tokens from each Chinese vocabulary, and seven tokens for each English vocabulary. The main criterion for selecting tokens is that they should be easy to make sentences with. According to the experimental results presented in Section 4.2, these data can effectively challenge the performance of LLMs, proving this issue deserves more attention. As for efficiently generating data in bulk, we design the framework for automatic generation in Section 3.3.

Then, for each selected token, we adopt one of the three approaches listed in Table 2, to convert it into a *challenging span* through inserting a special character span before or (and) after it. These challenging spans would disrupt the conventional tokenization process, thus confusing LLMs. The schemas and examples of three approaches are also shown in the second and third parts of Figure 2. The considerations in the three conversion approaches are introduced as follows.

1. **Before**: A character span s is inserted before origin token, causing the concatenation of s and the token's prefix is just another token existing in vocabulary, as 'move'+'s'→'moves' in Table 2.
2. **After**: A character span s is inserted after origin token, causing the concatenation of the token's suffix and s is just another token existing in vocabulary, as 'n'+'othing'→'nothing' in Table 2.
3. **Before & After**: Character spans s_1 and s_2 are inserted before and after origin token, respectively, causing the concatenation of s_1 and the token's prefix is just another token existing in vocabulary, meanwhile so is the concatenation of the token's suffix and s_2, as 'six'+'th'→'sixth' and 'is'+'land'→'island' in Table 2.

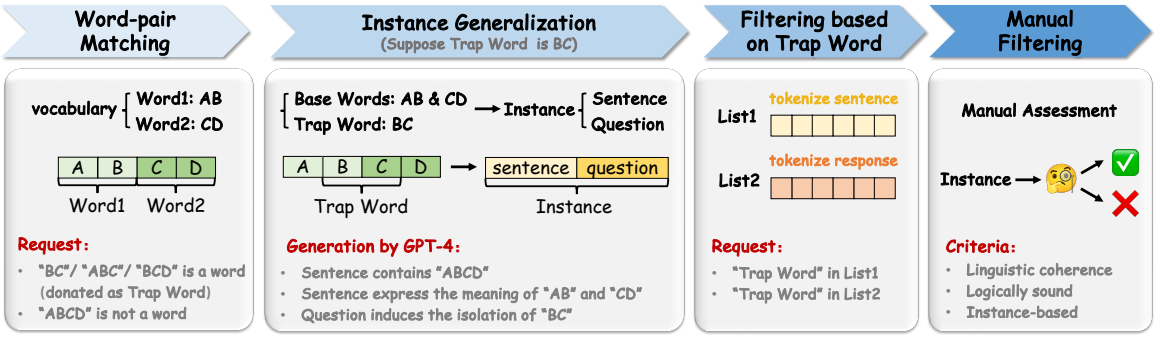


Figure 3: Our framework of generating ADT-Auto automatically.

Next, for each challenging span s_c, we manually compose a corresponding instance as the data in ADT-Human. One instance consists of a sentence in which s_c presents, and a corresponding question of which the answer comes from s_c. In Appendix A, we list all instances (along with their correct tokenizations) of ADT-Human. As we can see, it is challenging for LLMs to understand the instances due to the presence of challenging spans.

It's worth noting that the tokenization difficulty of English is less than that of Chinese, since spaces are regularly used as delimiters to separate each word from others in English. Moreover, affixation is common in English word structure, enabling tokenizers to divide a single word into several sections, which can help avoid incorrect tokenization to some extent. Thus, during the manual construction of English instances, we deliberately exclude the spaces between some tokens to provoke challenges to tokenization process. This decision stems from the recognition that powerful models should possess robust abilities across various scenarios which can occur in real-world applications, including handling cases where spaces might be omitted in English text input (Hofmann et al., 2022).

3.3 ADT-Auto Generation

Given the inefficiency of constructing dataset manually, we further develop an automatic generation framework for dataset to challenge LLMs' tokenization. As discussed in Section 3.2, the challenges of tokenization in English are less severe than in Chinese. Consequently, we primarily concentrate on the automatic generation of Chinese data. Figure 3 illustrates the process of automatically constructing our dataset ADT-Auto.

3.3.1 Word-pair Matching

The automatic generation of dataset is also based on exported vocabularies. From the vocabularies, we first seek some qualified word pairs. Given two words `Word 1` and `Word 2`, they are considered to be match when meeting the following criteria: The suffix (or whole) of `Word 1` can be concatenated with the prefix (or whole) of `Word 2`, as a token existing in the vocabulary, denoted as `Trap Word`. Meanwhile, the concatenation of `Word 1` and `Word 2` should not be a token existing in the vocabulary.

Accordingly, there are also three situations related to `Word 1`, `Word 2` and `Trap Word`, corresponding to three scenarios in Table 2.

1. If `Word 1` is included by `Trap Word`, this situation corresponds to the schema inserting a span after `Trap Word`.
2. If `Word 2` is included by `Trap Word`, this situation corresponds to the schema inserting a span before `Trap Word`.
3. If neither `Word 1` nor `Word 2` is included by `Trap Word`, this situation corresponds to the schema inserting a span both before and after `Trap Word`.

Above clarifications indicate that the concatenation of `Word 1` and `Word 2` (as 'ABCD' in Figure 3) corresponds to the challenging span in the construction process of ADT-Human. Our goal is to compose a sentence that not only convey the semantic essence of `Word 1` and `Word 2`, but also induce LLM to isolate `Trap Word` when tokenization. Notably, we ignore the situation that the concatenation of `Word 1` and `Word 2` is just `Trap Word`, as it would not pose challenges to LLMs' tokenization.

To augment the matching efficacy of `Trap Word` upon matching, we also consider the criterion that the remaining parts of `Word 1` and `Word 2` excluded by `Trap Word` should also exist in the vocabulary as a token. Furthermore, the first or last character of `Trap Word` cannot be a Chinese stop character.

3.3.2 Instance Generalization

With each `Word 1`, `Word 2` and their corresponding `Trap Word`, we harness GPT-4 to generate an instance of ADT-Auto, which consists of a sentence and a question. Specifically, we prompt GPT-4 to generate a sentence that is used to challenge the tokenization of LLMs. To this end, the generated sentence is required to include the concatenation of `Word 1` and `Word 2` (inevitably including `Trap Word`). In addition, GPT-4 is also required to devise a question related to the generated sentence, which is used to evaluate LLMs' tokenization performance through their answers. The instance must meet criteria as below: The sentence should convey the meanings of both `Word 1` and `Word 2`, while the question's answer should come from `Word 1`, `Word 2`, `Trap Word` or their combination. Thus, the influence of LLMs' incorrect tokenization can be identified by checking answers to the question.

The prompt for GPT-4 to generate instances includes some demonstration examples in addition to task instruction, which is illustrated in Appendix D.

3.3.3 Filtering Based on Trap Word

The goal of our dataset is to expose the flaw of LLMs' tokenization. Therefore, we will only retain the instances that can induce tokenization problem of LLMs, so we check each instance generated at the previous step. For these instances, the presence of `Trap Word` implies a challenging case that is likely to induce tokenization problems. Given an instance, we retain it if its corresponding `Trap Word` is both in the LLM's tokenization list for the instance's sentence (as List 1 in Figure 3) and in its tokenization list for the answer (as List 2 in Figure 3). Such filtering criterion indicates that the LLM commits tokenization errors on understanding the sentence and response for the instance.

3.3.4 Manual Filtering

To ensure the retained instances can induce tokenization problems and meanwhile have reasonable expressions, we further adopt manual assessment for instances. Specifically, we select the high-qualify instances with considering sentence expressions and LLM's responses. Notably, we might still retain some instances to which the used LLM has correct response, since the other LLMs are still likely to commit inaccurate tokenization for these instances, resulting in unsatisfactory responses.

Due to space limitation, we take Qwen-7B as an example to illustrate the process of generating instances. There are 24,953 Chinese tokens in Qwen-7B's vocabulary, and after word-pair matching, 1,764,692 word-pairs are obtained. From the matching word-pairs, 8,000 pairs are selected randomly and used for instance generation by GPT-4. Due to the inherent stochastic characteristic of LLMs on response generation, we conduct three iterations of filtering operations introduced in Subsection 3.3.3 to get more qualified instances, and thus retain 894 instances. Next, through the manual filtering introduced in Subsection 3.3.4, we retain 231 instances for ADT-Auto in the end, which is available on github.

4 Experiments

4.1 Experiment Setup

Considering the open-source LLMs used in our experiments, we select the LLMs previously used in the construction of ADT-Human, including Chatglm3-6B, Baichuan2-13B-Chat, Yi-34B-Chat, Qwen-7B-Chat, and Qwen1.5-72B-Chat for Chinese data, as well as Llama-3-8B-Instruct, Llama-3-70B-Instruct, and Mixtral-8x7B-Instruct-v0.1 for English data. We test these LLMs using both locally deployed versions and API versions, with the exception of Chatglm3-6B, which does not have API version. For the closed-source LLMs, we test GPT-4o, GPT-4, GPT-3.5-Turbo, Qwen2.5-max, step-1-8k[1], moonshot-v1-8k[2], ERNIE-3.5-8K[3] for Chinese data, and GPT-4o, GPT-4 and GPT-3.5-Turbo for English data. Additionally, we test the API version of Deepseek-R1, which has recently gained significant attention, for Chinese data.

In our experiments, we directly use the dataset ADT constructed with the method introduced in Section 3, which includes the manually constructed ADT-Human (containing Chinese and English instances) and the automatically generated ADT-Auto (only containing Chinese instances).

In addition, we conduct our experiments on the platform with four A800 GPUs.

4.2 ADT-Human's Challenges to LLMs

Firstly, we investigate the challenges posed by ADT-Human to LLMs. Specifically, we evaluate LLMs' performance through counting the number of incorrect answers generated by LLMs for the

[1] https://platform.stepfun.com/docs/llm/text
[2] https://platform.moonshot.cn/docs
[3] https://cloud.baidu.com/doc/WENXINWORKSHOP/s/jlil56u11

	Model	Source LLM of vocabulary				Overall error rate
		Chatglm3	Baichuan2	Yi	Qwen	
Open-source (Local)	Chatglm3-6B	100.00	100.00	100.00	90.91	**97.73**
	Baichuan2-13B-Chat	90.91	100.00	81.82	100.00	**93.18**
	Yi-34B-Chat	72.73	63.64	100.00	100.00	**84.09**
	Qwen-7B-Chat	100.00	72.73	90.91	100.00	**90.91**
	Qwen1.5-72B-Chat	90.91	45.45	81.82	100.00	**79.55**
Open-source (API)	Baichuan2-13B-Chat	100.00	100.00	90.91	100.00	**97.73**
	Yi-34B-Chat	81.82	54.55	100.00	90.91	**81.82**
	Qwen-7B-Chat	100.00	81.82	81.82	100.00	**90.91**
	Qwen1.5-72B-Chat	90.91	54.55	81.82	100.00	**81.82**
	DeepSeek-R1	36.36	27.27	45.45	54.55	**40.91**
Closed-source	GPT-4o	72.73	27.27	54.55	90.91	**61.36**
	GPT-4	81.82	45.45	45.45	27.28	**50.00**
	GPT-3.5-Turbo	72.73	27.27	72.73	72.73	**61.36**
	Qwen2.5-max	90.91	72.73	90.91	100.00	**88.64**
	step-1-8k	63.64	18.18	72.73	63.64	**54.55**
	moonshot-v1-8k	81.82	27.27	100.00	81.82	**72.73**
	ERNIE-3.5-8K	72.73	54.55	54.55	72.73	**63.64**

Table 3: Error rates (%) of answers on ADT-Human (Chinese).

	Model	Source LLM of vocabulary		Overall error rate
		Llama-3	Mixtral	
Open-source (Local)	Llama-3-8B-Instruct	100.00	85.71	**92.86**
	Llama-3-70B-Instruct	57.14	71.43	**64.29**
	Mixtral-8x7B-Instruct-v0.1	85.71	100.00	**92.86**
Open-source (API)	Llama-3-8B-Instruct	100.00	71.43	**85.71**
	Llama-3-70B-Instruct	57.14	28.57	**42.86**
	Mixtral-8x7B-Instruct-v0.1	71.43	100.00	**85.71**
Closed-source	GPT-4o	57.14	71.43	**64.29**
	GPT-4	57.14	57.14	**57.14**
	GPT-3.5-Turbo	71.43	57.14	**64.29**

Table 4: Error rates (%) of answers on ADT-Human (English).

questions in instances. Recalling the process of manually composing challenging spans introduced in Section 3.2, the span 'BC' in Table 2 is in fact the `Trap Word` mentioned in Section 3.3. Hence, for a given LLM, its answer including a `Trap Word` is identified as inaccurate response for the question undoubtedly. We identify the correctness of LLMs' responses through human assessment.

The percentage of incorrect responses provided by the LLMs for Chinese data and English data are presented in Table 3 and Table 4, respectively. The results show that ADT-Human poses a significant challenge to both open-source and closed-source LLMs on their tokenization, resulting in very high rates of inaccurate responses. It is also worth mentioning that the recent state-of-the-art (SOTA) GPT-4o cannot outperform GPT-4 , implying that the advancements of these LLMs have not yet addressed this primary but challenging problem.

To further investigate the impact of tokenization on the performance of LLMs, a fine-grained analysis of the relationship between tokenization and response is conducted. In Appendix B, we quantitatively examine the relationship between the correctness of tokenization and the correctness of LLM's response. Given the inaccessibility of tokenization lists for closed-source LLMs, we perform a statistical analysis on open-source LLMs. Since LLMs' API versions do not directly provide tokenization lists, the tokenization results obtained from the corresponding locally deployed versions are used. Appendix B.1 and Appendix B.2 respectively illustrate the quantitative relationships between tokenization and response for the LLMs tested on Chinese and English data of ADT-Human.

This study further intuitively illustrates the rela-

	Right response	Wrong response
Right tokenization	3	3
Wrong tokenization	1	37

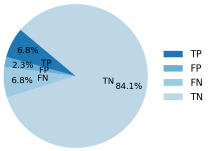


Figure 4: Four relationships between tokenization and response, take Qwen-7B-Chat as an example.

	Model	Fraction	Error rate (%)
Open-source (Local)	Chatglm3-6B	156/231	**67.53**
	Baichuan2-13B-Chat	147/231	**63.64**
	Yi-34B-Chat	90/231	**38.96**
	Qwen-7B-Chat	185/231	**80.09**
	Qwen1.5-72B-Chat	93/231	**40.26**
Open-source (API)	Baichuan2-13B-Chat	167/231	**72.29**
	Yi-34B-Chat	80/231	**34.63**
	Qwen-7B-Chat	160/231	**69.26**
	Qwen1.5-72B-Chat	97/231	**41.99**
	DeepSeek-R1	57/231	**24.68**
Closed-source	GPT-4o	75/231	**32.47**
	GPT-4	89/231	**38.53**
	GPT-3.5-Turbo	99/231	**42.86**
	Qwen2.5-max	75/231	**32.47**
	step-1-8k	60/231	**25.97**
	moonshot-v1-8k	60/231	**25.97**
	ERNIE-3.5-8K	51/231	**22.08**

Table 5: Error rates of answers on ADT-Auto.

tionships in the form of pie charts in Appendix C. In Appendix C.1 and Appendix C.2, the proportions of four relationships between tokenization and response for each LLM on Chinese and English data of ADT-Human are displayed respectively. Figure 4 illustrates the quantitative relationships and pie chart using Qwen-7B-Chat as an example. For more details, please refer to Appendix B and Appendix C. We mainly focus on the proportion of TN (tokenization incorrect and response incorrect). As shown in pie charts, the proportion of TN cases is very high in ADT-Human, with an average of 80.91% for Chinese data and 79.78% for English data. This indicates that tokenization errors significantly affect the accuracy of LLM responses and also demonstrates that ADT-Human effectively challenges the performance of LLMs.

4.3 ADT-Auto's Challenges to LLMs

Similar to the investigation of ADT-Human's challenges to LLMs' tokenization, we also evaluate the LLMs' performance on ADT-Auto.

ADT-Auto's instances come from Qwen-7B's vocabulary, and the rates of LLMs' inaccurate responses for these instances' questions are listed in Table 5. Based on the results, we have the following observations and analysis:

1. Compared with closed-source LLMs, open-source LLMs suffer from ADT-Auto's challenges more apparently. It implies that these closed-source LLMs, as the profit-making flagships of their creator companies, naturally have stronger capabilities than open-source LLMs that are created for public usage.
2. Compared with ADT-Human, ADT-Auto is less challenging to LLMs, since the sentences generated by GPT-4 have more formal, regular or simple syntaxes and expressions than the manually composed sentences in ADT-Human. Thus, these sentences in ADT-Auto are relatively easy for LLMs' understanding.
3. Qwen1.5-72B-Chat has lower error rates than Qwen-7B-Chat, although they have the same vocabulary. We specially check some instances to which Qwen1.5-72B-Chat gives correct answers but Qwen-7B-Chat gives wrong answers, and find the two models have the same incorrect tokenization lists for these instances. It suggests that their different performance is not caused by tokenization. The results also imply that in the case of incorrect tokenization, the bigger LLMs are more robust and likely to generate correct responses than the smaller LLMs thanks to their stronger capabilities brought by the larger scale.

Similar to Section 4.2, Appendix B.3 presents the quantitative relationships between tokenization and response for each open-source LLM tested on ADT-Auto. The corresponding pie charts are shown in Appendix C.3. As indicated by the pie charts, the proportion of TN cases is also very high on ADT-Auto, with an average of 46.11%. This further demonstrates that tokenization errors significantly impact the accuracy of LLMs' responses and highlights the effectiveness of ADT-Auto in challenging the performance of LLMs.

5 Conclusion

In this paper, we dedicate to deeply investigating the relationship between LLMs' vulnerability on tokenization and their unsatisfactory responses for some tasks. To this end, we construct an adversarial dataset **ADT (Adversarial Dataset for Tokenizer)** consisting of a manually constructed subset ADT-Human and an automatically generated subset ADT-Auto. Our experiments demonstrate that our dataset does challenge the studied open-source and closed-source LLMs' token segmentation, resulting in their incorrect answers. We hope our work and dataset could shed light on the subsequent research on improving LLMs' performance.

Limitations

The key contribution of this study lies in drawing attention to the impact of tokenization on LLM's performance and providing two frameworks for data generation. As for how to propose improvement strategies based on this phenomenon, we are currently in the process of exploration. Additionally, this study focuses primarily on Chinese. Whether other languages are similarly affected by tokenization remains to be further investigated.

Ethics Statement

We hereby declare that all authors of this article are aware of and adhere to the provided ACL Code of Ethics and honor the code of conduct.

Use of Human Annotations Human annotations are only used in methodological research at the beginning of the work, to assist in analyzing the feasibility of the proposed solution. Annotators consented to the use of data for research purposes. We ensure that the privacy of all annotators is protected throughout the annotation process, and all of them are adequately paid according to local standards.

Risks Synthetic data generated by LLMs may involve potential ethical risks regarding fairness and bias (Bommasani et al., 2021; Blodgett et al., 2020), which results in further consideration when they are employed in downstream tasks. We asked our members for proofreading to refine the offensive and harmful data generated by GPT-4. Despite these considerations, there may still be some unsatisfactory data that goes unnoticed in our final dataset.

References

Jinze Bai, Shuai Bai, and Yunfei Chu et al. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Yejin Bang, Samuel Cahyawijaya, and Nayeon Lee et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 675–718. Association for Computational Linguistics.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tom B. Brown, Benjamin Mann, and Nick Ryder et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 2633–2650. USENIX Association.

Eunbi Choi, Yongrae Jo, Joel Jang, and Minjoon Seo. 2022. Prompt injection: Parameterization of fixed inputs. *CoRR*, abs/2206.11349.

Aakanksha Chowdhery, Sharan Narang, and Jacob Devlin et al. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Arijit Ghosh Chowdhury, Md Mofijul Islam, Vaibhav Kumar, Faysal Hossain Shezan, Vaibhav Kumar, Vinija Jain, and Aman Chadha. 2024. Breaking down the defenses: A comparative survey of attacks on large language models. *CoRR*, abs/2403.04786.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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.

Aysan Esmradi, Daniel Wankit Yip, and Chun-Fai Chan. 2023. A comprehensive survey of attack techniques, implementation, and mitigation strategies in large language models. In *Ubiquitous Security - Third International Conference, UbiSec 2023, Exeter, UK, November 1-3, 2023, Revised Selected Papers*, volume 2034 of *Communications in Computer and Information Science*, pages 76–95. Springer.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. 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 3594–3608, Online. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian O. Sabel, Jens Ricke, and Michael Ingrisch. 2022. Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports. *CoRR*, abs/2212.14882.

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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Serge Kernbach. 2022. Electrochemical characterization of ionic dynamics resulting from spin conversion of water isomers. *Journal of The Electrochemical Society*, 169(6):067504.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on sesame street! model extraction of bert-based apis. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. *arXiv e-prints*, pages arXiv–2405.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *CoRR*, abs/2303.14070.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human

falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Lan Liu, Yi Wang, Gaoyang Liu, Kai Peng, and Chen Wang. 2023. Membership inference attacks against machine learning models via prediction sensitivity. *IEEE Trans. Dependable Secur. Comput.*, 20(3):2341–2347.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kamil Malinka, Martin Peresíni, Anton Firc, Ondrej Hujnak, and Filip Janus. 2023. On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree? In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, ITiCSE 2023, Turku, Finland, July 7-12, 2023*, pages 47–53. ACM.

John J. Nay. 2022. Law informs code: A legal informatics approach to aligning artificial intelligence with humans. *CoRR*, abs/2209.13020.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Teo Susnjak. 2022. Chatgpt: The end of online exam integrity? *CoRR*, abs/2212.09292.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *CoRR*, abs/2303.04360.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Hugo Touvron, Louis Martin, and Kevin Stone et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 139–150. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2023. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *CoRR*, abs/2305.14710.

Aiyuan Yang, Bin Xiao, and Bingning Wang et al. 2023. Baichuan 2: Open large-scale language models. *CoRR*, abs/2309.10305.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *CoRR*, abs/2403.04652.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal prompting: Teaching a language model to think like a lawyer. *CoRR*, abs/2212.01326.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In

The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A Details of ADT-Human

A.1 Chatglm3

1. 小说家 | 发表了 | 以 | 工作历程 | 为主题 | 的 | A 小说 | 及 | 以 | 上学历程 | 为主题 | 的 | B 小说 | 。| 请问 | B 小说 | 的 | 主题 | 是什么 | ?
2. 作为 | 负责团体 | , | 复旦大学 | 社团 | 知和社 | 会 | 保障 | 活动 | 的 | 顺利进行 | 。| 请问 | 哪个 团体 | 是 | 负责团体 | ?
3. 小明 | 在 | 海参 | 加工 | 作坊 | 工作 | 。| 请问 | 他 | 在哪里 | 工作 | ?
4. 这家 | 培训机构 | 的 | 管教师资 | 格外 | 出色 | 。| 请问 | 这家机构 | 的 | 哪方面 | 比较出色 | ?
5. 新上映 | 的 | 一部 | 分镜 | 出色 | 的 | 电影 | 饱受 好评 | 。| 请问 | 这部电影 | 哪一点出色 | ?
6. 小明 | 代表 | 达能 | 力荐 | 公司 | 新推出 | 的 | 一 款 | 营养饼干 | 。| 请问 | 小明 | 代表的 | 是 | 哪 个 | 品牌 | ?
7. 职工 | 作为 | 一个 | 人数庞大 | 的 | 群体 | , | 已 经 | 有一些 | 渠道 | 反映 | 他们的 | 意见 | 。| 请 问 | 文中 | 提到了 | 什么 | 群体 | ? | 用 | 词 | 或 | 短语 | 回答 | 。
8. 我的 | 公公 | 开发 | 行测宝典 | , | 帮助 | 考公 | 人士 | 上岸 | 。| 请问 | 句子中 | 提到了 | 我的 | 什么人 | , | 他 | 开发了 | 什么 | 宝典 | ?
9. 主任 | 何时 | 候机去 | 啊 | ? | 这句话 | 问的是 | 主任 | 去做 | 什么事 | 的 | 时间点 | ?
10. 公司 | 正在 | 宣传 | 一款 | 低脂肪 | 酸奶 | 。| 请 问 | 这款 | 产品 | 是 | 牛奶 | 还是 | 酸奶 | ?
11. 这次 | 出征 | , | 中国 | 国奥 | 地利 | 天时 | 一条 | 不占 | 。| 请问 | 句子中 | 哪个 | 队伍 | 不被 | 看 好 | ?

A.2 Baichuan2

1. 初中 | 毕业 | 生育 | 是 | 一种 | 不合适的 | 人生规划 | 。 | 请问 | 句子中 | 说 | 初中毕业 | 不适合 | 做什么 | ?
2. 小明 | 说 | 自己 | 准备 | 考试 | 的 | 时间 | “ | 两个月不到 | 一个月不止 | ” | 。 | 请问 | 他 | 准备时间 | 超过 | 两个月 | 了吗 | ?
3. 日本 | 公众 | 号召 | 抵制 | 核废水 | 。 | 请问 | 句子中 | 谁在号召 | ?
4. 知了 | 自己的 | 叫声 | , | 在枝头 | 宣告了 | 自己 | 的 | 存在 | 。 | 这句话里 | 的 | 动物 | 是什么 | ?
5. 经管 | 理工 | 作为 | 就业 | 的 | 两大 | 热门 | 方向 | , | 长期 | 受人 | 关注 | 。 | 请问 | 句子中 | 两大 | 热门方向 | 分别是 | ?
6. 韩国 | 家电 | 网购 | 的 | 销量 | 逐年递增 | 。 | 请问 | 句子中 | 提到了 | 什么 | 产品 | ?
7. 这个 | 玩具 | 体内 | 容纳了 | 很多 | 绵软的 | 纤维 | 。 | 请问 | 句子中 | 什么东西 | 容纳了 | 绵软的 | 纤维 | ?
8. 他本身 | 体力 | 行得很 | , | 可是 | 不是 | 很聪明 | 。 | 请问 | 句子中 | 这个人 | 的 | 强项 | 是什么 | ?
9. 一方 | 水土 | 养 | 一方人 | , | 一方人 | 吃 | 一方 | 面食 | 。 | 请问 | 句子中 | 提到了 | 哪种 | 食物 | ?
10. 学校 | 在线 | 上课 | 程序 | 出现 | 漏洞 | 。 | 请问 | 句子中 | 什么 | 出现 | 漏洞 | ?
11. 在 | 科学 | 学习中 | , | 水 | 的 | 三相 | 对于 | 三年级 | 的 | 同学 | 来说 | 是个 | 很难理解 | 的 | 知识点 | 。 | 请问 | 句子中 | 同学 | 很难 | 理解的 | 知识点 | 是什么 | ?

A.3 Yi

1. 为什么 | 会计 | 算数 | 还 | 不如 | 秘书 | 快 | 。 | 请问 | 句子中 | 提到了 | 哪两个 | 职业 | ?
2. 随着 | 双十一 | 到来 | , | 短时间 | 内衣 | 的 | 销量 | 翻倍 | 。 | 请问 | 句子中 | 提到了 | 什么 | 类型 | 的 | 衣服 | ?
3. 我不知 | 道家 | 思想 | 竟然 | 如此 | 博大精深 | 。 | 请问 | 句子中 | 提到了 | 诸子百家 | 中 | 哪一家 | 的 | 思想 | ?
4. 他的 | 讲解 | 深入了 | 解压操作 | 的 | 每个细节 | 。 | 请问 | 句子中 | 他的 | 讲解 | 深入了 | 什么操作 | ?
5. 即使 | 在 | 事务所 | 有人 | 都 | 不喜欢 | 自己的 | 岗位 | , | 更别提 | 福利 | 待遇 | 一般 | 的 | 小公司 | 。 | 请问 | 句子中 | 提到 | 在哪里 | 有人 | 不喜欢 | 自己的 | 岗位 | ?
6. 多吃 | 葡萄 | 牙 | 会疼 | 。 | 请问 | 这句话 | 说明 | 什么事 | 的 | 危害 | ?
7. 我在 | 东四十条 | 第一 | 款待所 | 招待 | 我的 | 朋友 | 。 | 请问 | 我在 | 哪个 | 地点 | ?
8. 在 | 各种 | 公司 | 中 | 国企 | 业务 | 扩张 | 迅速 | 。 | 请问 | 句子中 | 提到了 | 哪类 | 公司 | ?
9. 我 | 不得不 | 说服 | 从头到脚 | 都 | 固执 | 的 | 兄弟 | 改变 | 安排 | 。 | 请问 | 句子中 | 我要 | 做 | 什么事 | ?
10. 小明 | 比较 | 好奇 | 心率 | 先变高 | 后不变 | 是 | 什么 | 原因 | , | 请问 | 句子中 | 小明 | 关心 | 什么的 | 变化 | 趋势 | ?
11. 什么 | 时候 | 官宣 | 告五人 | 演唱会 | ?

A.4 Qwen

1. 美国 | 社会中 | 华人民众 | 很难 | 成为 | 社会 | 主流 | 声音 | 。 | 请问 | 句子中 | 谁 | 很难 | 成为 | 社会 | 主流 | 声音 | ?
2. 政府 | 决定 | 投资 | 本市 | 场馆 | 和 | 体育 | 设施 | 建设 | 。 | 请问 | 句子中 | 政府 | 决定 | 投资 | 什么 | 方面 | 的 | 建设 | ?
3. 大多 | 数学专业 | 的 | 同学 | 也 | 具有 | 一定的 | 计算机 | 能力 | 。 | 请问 | 句子中 | 提到了 | 哪个 | 专业的 | 同学 | ?
4. 据 | 统计 | , | “ | 排水 | 规划 | 不合理 | ” | 占 | 地面积水 | 发生 | 原因 | 的 | 50% | 。 | 请问 | 上述 | 句子 | 在 | 分析 | 什么 | 问题 | 的 | 原因 | ?
5. 事情 | 的 | 是非 | 常常 | 不如 | 共情 | 能力 | 重要 | 。 | 请问 | 句子中 | 什么 | 东西 | 不如 | 共情 | 能力 | 重要 | ?
6. 现代 | 化学 | 生物 | 的 | 发展 | 离不开 | 科研 | 人员 | 的 | 辛勤探索 | 。 | 请问 | 这句话 | 提到了 | 哪两个 | 学科 | ?
7. 分辨 | 率真 | 和 | 鲁莽 | 的 | 一个 | 标准 | 是 | 是否 | 让 | 他人 | 感觉 | 不适 | 。 | 请问 | 这句话中 | 提到的 | 较好的 | 品质 | 是什么 | ?
8. 客户 | 端起 | 杯子 | 和 | 老板 | 干杯 | 。 | 请问 | 谁 | 在 | 和 | 老板 | 干杯 | ?
9. 在 | 晚会 | 上 | , | 电影节目 | 前沿技术 | 的 | 应用 | 十分精彩 | 。 | 请问 | 句子中 | 提到了 | 什么 | 类型 | 的 | 节目 | ?
10. 领导 | 对于 | 这个职位 | 的 | 心理选择 | 更 | 倾向 | 于经理 | 而不是 | 王经理 | 。 | 请问 | 领导 | 倾向的 | 人 | 是谁 | ?
11. 弗兰克 | 是 | 一名 | 足球迷 | , | 他 | 总是 | 在周末 | 去现场 | 助威 | 尼斯足球队 | 。 | 请问 | 他 | 会给 | 哪支 | 足球队 | 加油 | ?

A.5 Llama-3

1. In the postal history course, today's homework is to **list|envelope|classes** in history. What should I do?
2. As a Marathon lover, in the past he only **ran|domestically**, but now he also goes abroad. What behavior of him is discussed in the sentence?
3. As fans of Kenshi Yonezu, they **sing|lemon** in the auditorium. What song are they singing in the sentence?
4. I wanted to go to the beach, **but|tons|of** work piled up. How much work did I have?
5. We are measuring how **fat|her|cat** is to make sure her cat is healthy. What metric are we measuring?
6. This automotive shop mainly sells **car|bonnets**. What does the shop sell?
7. The researcher was disappointed to **miss|ionization** in the sample, which was crucial for the experiment's success. What made the researcher disappointed?

A.6 Mixtral

1. The analyst emphasized the importance of tracking **live|return** to gauge real-time performance. What importance was emphasized by the analyst?
2. Many countries **import|sports** from another country. What do those countries import?
3. He **moves|table** to another place. What is he doing?
4. Those pants and leather shoes **fitted|speakers** very well. Who do those pants and leather shoes fit well?
5. The soccer team **won|derby** against its rival. Did the soccer team win or lose?
6. They **swap|pears** with each other. What are they exchanging?
7. The **leg|ends|up** being the most injured part, requiring immediate medical attention. In the sentence, which part is injured the most?

B The relationship between tokenization and response

B.1 ADT-Human (Chinese)

	Right response	Wrong response
Right tokenization	2	1
Wrong tokenization	1	40

(a) Baichuan2-13B-Chat (Local)

	Right response	Wrong response
Right tokenization	1	2
Wrong tokenization	0	41

(b) Baichuan2-13B-Chat (API)

	Right response	Wrong response
Right tokenization	3	0
Wrong tokenization	5	36

(c) Yi-34B-Chat (Local)

	Right response	Wrong response
Right tokenization	3	0
Wrong tokenization	7	34

(d) Yi-34B-Chat (API)

	Right response	Wrong response
Right tokenization	3	3
Wrong tokenization	1	37

(e) Qwen-7B-Chat (Local)

	Right response	Wrong response
Right tokenization	3	3
Wrong tokenization	1	37

(f) Qwen-7B-Chat (API)

	Right response	Wrong response
Right tokenization	6	0
Wrong tokenization	3	35

(g) Qwen1.5-72B-Chat (Local)

	Right response	Wrong response
Right tokenization	6	0
Wrong tokenization	2	36

(h) Qwen1.5-72B-Chat (API)

	Right response	Wrong response
Right tokenization	1	1
Wrong tokenization	0	42

(i) Chatglm3-6B (Local)

	Right response	Wrong response
Right tokenization	3	0
Wrong tokenization	23	18

(j) Deepseek-R1 (API)

B.2 ADT-Human (English)

	Right response	Wrong response
Right tokenization	1	0
Wrong tokenization	0	13

(a) Llama-3-8B-Instruct (Local)

	Right response	Wrong response
Right tokenization	0	1
Wrong tokenization	0	13

(b) Llama-3-8B-Instruct (API)

	Right response	Wrong response
Right tokenization	0	0
Wrong tokenization	3	11

(c) Llama-3-70B-Instruct (Local)

	Right response	Wrong response
Right tokenization	0	0
Wrong tokenization	8	6

(d) Llama-3-70B-Instruct (API)

	Right response	Wrong response
Right tokenization	0	0
Wrong tokenization	2	12

(e) Mixtral-8x7B-Instruct-v0.1 (Local)

	Right response	Wrong response
Right tokenization	0	0
Wrong tokenization	2	12

(f) Mixtral-8x7B-Instruct-v0.1 (API)

B.3 ADT-Auto

	Right response	Wrong response
Right tokenization	28	32
Wrong tokenization	56	115

(a) Baichuan2-13B-Chat (Local)

	Right response	Wrong response
Right tokenization	23	37
Wrong tokenization	41	130

(b) Baichuan2-13B-Chat (API)

	Right response	Wrong response
Right tokenization	45	14
Wrong tokenization	96	76

(c) Yi-34B-Chat (Local)

	Right response	Wrong response
Right tokenization	48	11
Wrong tokenization	103	69

(d) Yi-34B-Chat (API)

	Right response	Wrong response
Right tokenization	4	22
Wrong tokenization	42	163

(e) Qwen-7B-Chat (Local)

	Right response	Wrong response
Right tokenization	10	16
Wrong tokenization	61	144

(f) Qwen-7B-Chat (API)

	Right response	Wrong response
Right tokenization	21	5
Wrong tokenization	117	88

(g) Qwen1.5-72B-Chat (Local)

	Right response	Wrong response
Right tokenization	20	6
Wrong tokenization	114	91

(h) Qwen1.5-72B-Chat (API)

	Right response	Wrong response
Right tokenization	32	21
Wrong tokenization	43	135

(i) Chatglm3-6B (Local)

	Right response	Wrong response
Right tokenization	59	3
Wrong tokenization	115	54

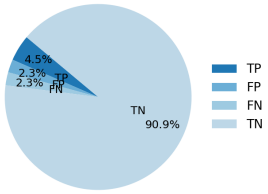
(j) Deepseek-R1 (API)

C Proportion of four situations between tokenization and response

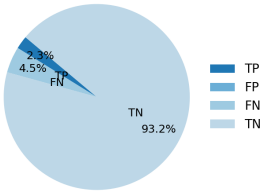
Define the four relationships between tokenization and response:

- **TP:** Correct tokenization and correct response.
- **FP:** Incorrect tokenization but correct response.
- **FN:** Correct tokenization but incorrect response.
- **TN:** Incorrect tokenization and incorrect response.

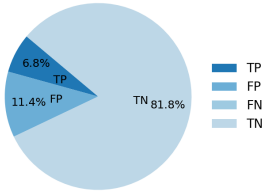
C.1 ADT-Human (Chinese)



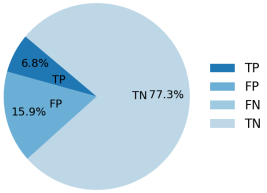
(a) Baichuan2-13B-Chat (Local)



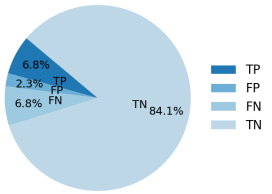
(b) Baichuan2-13B-Chat (API)



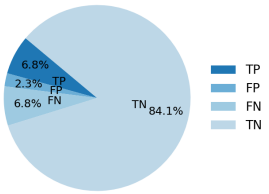
(c) Yi-34B-Chat (Local)



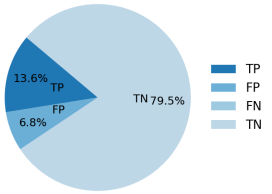
(d) Yi-34B-Chat (API)



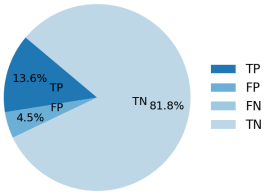
(e) Qwen-7B-Chat (Local)



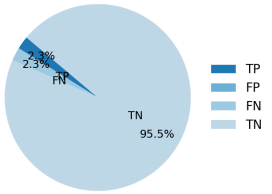
(f) Qwen-7B-Chat (API)



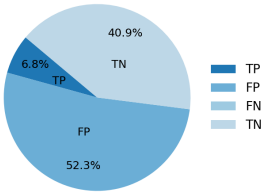
(g) Qwen1.5-72B-Chat (Local)



(h) Qwen1.5-72B-Chat (API)

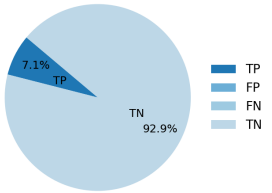


(i) Chatglm3-6B (Local)

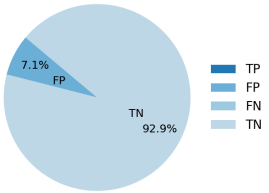


(j) Deepseek-R1 (API)

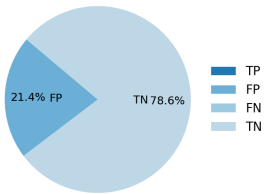
C.2 ADT-Human (English)



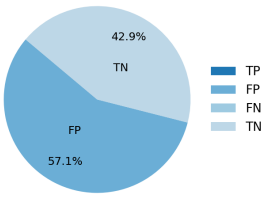
(a) Llama-3-8B-Instruct (Local)



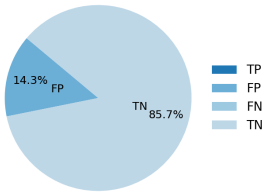
(b) Llama-3-8B-Instruct (API)



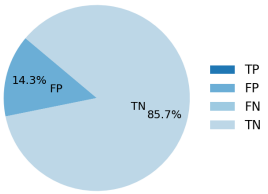
(c) Llama-3-70B-Instruct (Local)



(d) Llama-3-70B-Instruct (API)

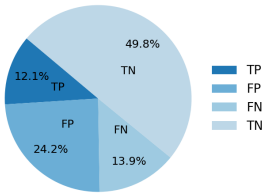


(e) Mixtral-8x7B-Instruct-v0.1 (Local)

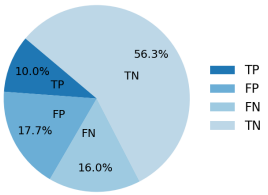


(f) Mixtral-8x7B-Instruct-v0.1 (API)

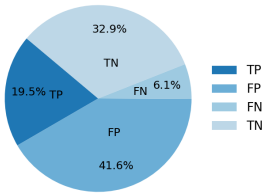
C.3 ADT-Auto



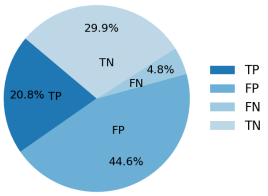
(a) Baichuan2-13B-Chat (Local)



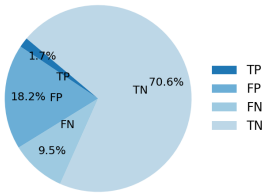
(b) Baichuan2-13B-Chat (API)



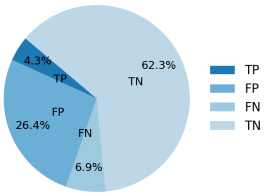
(c) Yi-34B-Chat (Local)



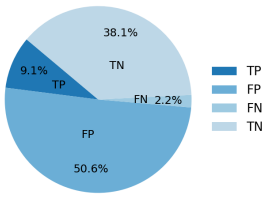
(d) Yi-34B-Chat (API)



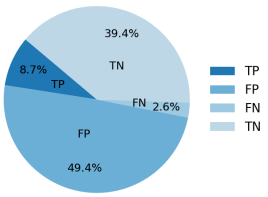
(e) Qwen-7B-Chat (Local)



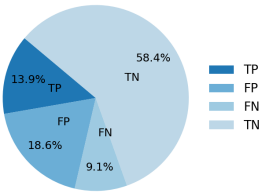
(f) Qwen-7B-Chat (API)



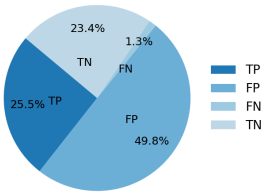
(g) Qwen1.5-72B-Chat (Local)



(h) Qwen1.5-72B-Chat (API)



(i) Chatglm3-6B (Local)



(j) Deepseek-R1 (API)

D Details of prompt

| Generation Instruction *|*
As a Chinese linguist, you are tasked with completing a sentence generation. The procedure is as follows:
1. You will receive two base words and one trap word. The trap word is formed by combining the suffix of word 1 and the prefix of word 2.
2. Generate a sentence where two base words are conjoined such that the end of word 1 immediately precedes the beginning of word 2, without the insertion of spaces or punctuation between them.
3. Expand upon the sentence by posing a question that pertains to the semantic content conveyed by the base words. The question should be formulated in a manner that leads the respondent to conflate the meanings of the base words and the trap word.
4. Ensure that the sentence is coherent and logically sound.

Chinese Input	English Translation
| Three demonstration examples *|*	*|* Three demonstration examples *|*
Word 1: 经管理工; **Word 2:** 作为;	**Word 1:** economic & management and science & engineering; **Word 2:** act as;
Trap Word: 管理工作	**Trap Word:** stewardship
Result: 经管理工作为就业的两大热门方向，长期受人关注。请问两大热门方向分别是?	**Result:** The two most popular areas of employment are economic&management and science&engineering, which have been in the spotlight for a long time. What are the two most popular areas of employment?
...	...
| Input *|*	*|* Input *|*
Word 1: 老虎机; **Word 2:** 会觉得;	**Word 1:** slot machine; **Word 2:** feel that;
Trap Word: 机会	**Trap Word:** chances
| GPT-4's output *|*	*|* GPT-4's output *|*
Result: 他表示如果赢了老虎机会觉得自己非常幸运。请问他赢了什么后会有什么感觉?	**Result:** He said he would feel very lucky if he won the slot machine game. May I ask how he would feel after winning anything?