mCSQA: Multilingual Commonsense Reasoning Dataset with Unified Creation Strategy by Language Models and Humans

Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe
Nara Institute of Science and Technology
{sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

It is very challenging to curate a dataset for language-specific knowledge and common sense in order to evaluate natural language understanding capabilities of language models. Due to the limitation in the availability of annotators, most current multilingual datasets are created through translation, which cannot evaluate such language-specific aspects. Therefore, we propose Multilingual CommonsenseQA (mCSQA) based on the construction process of CSQA but leveraging language models for a more efficient construction, e.g., by asking LM to generate questions/answers, refine answers and verify QAs followed by reduced human efforts for verification. Constructed dataset is a benchmark for cross-lingual language-transfer capabilities of multilingual LMs, and experimental results showed high language-transfer capabilities for questions that LMs could easily solve, but lower transfer capabilities for questions requiring deep knowledge or commonsense. This highlights the necessity of language-specific datasets for evaluation and training. Finally, our method demonstrated that multilingual LMs could create QA including language-specific knowledge, significantly reducing the dataset creation cost compared to manual creation. The datasets are available at https://huggingface.co/datasets/yusuke1997/mCSQA.

1 Introduction

Can you choose the correct answer in Table 1? Each choice is semantically very close, making it difficult for non-native speakers to distinguish them. However, native speakers who have language-specific commonsense and knowledge can choose the most plausible choice considering subtle nuances. Despite the need to consider different backgrounds for each language, the datasets to evaluate the natural language understanding (NLU) capabilities of language models (LMs) are mostly for a few major languages such as English, and thus, many languages lack such datasets. When focusing on the cross-lingual capability of LMs, datasets created from scratch in multiple languages are lim-
ited, and currently, evaluations mostly use datasets created through translation. However, as can be seen from the example in Table 1, datasets created through translation cannot accurately evaluate language-specific commonsense or knowledge. Therefore, it is necessary to create datasets for each language from scratch, but the manual creation of such datasets is limited by the availability of annotators and financial costs.

To tackle this problem, as shown in Figure 1, we propose a method to efficiently create multilingual NLU datasets from multilingual resources by replacing some of the manual annotation processes with generative multilingual LMs. In this study, we focus on CommonsenseQA (CSQA) (Talmor et al., 2019), a dataset for evaluating commonsense reasoning capabilities within NLU evaluations. CSQA is a major commonsense reasoning Question-Answering dataset manually created from the multilingual knowledge base ConceptNet (Speer et al., 2017). However, due to such limitations, CSQA has been created from scratch only in English and Japanese. JCommonsenseQA (JCSQA) (Kurihara et al., 2022). Therefore, we create a Multilingual CommonsenseQA (mCSQA) that extends CSQA to eight languages\footnote{English (en), Japanese (ja), Chinese (zh), German (de), Portuguese (pt), Dutch (nl), French (fr), Russian (ru)} using our proposed method.

Furthermore, we evaluated the cross-lingual language-transfer capabilities of multilingual LMs focusing on language-specific common sense and knowledge using mCSQA. The results showed high language-transfer capabilities for questions that LMs could easily solve, but lower transfer capabilities for questions requiring deep knowledge or commonsense. The total cost per question in mCSQA was reduced to one-hundredth of that for CSQA.

To summarize, our contributions are as follows:

- We propose an efficient and low-cost method for creating NLU datasets by generative multilingual LMs.
- We demonstrate the potential effectiveness of using multilingual LMs for creating datasets from multilingual resources.
- mCSQA makes it possible to analyze the cross-linguistic commonsense understanding capabilities and transfer performance from each language beyond English.

\section{Background and Related Work}

\subsection*{Commonsense reasoning task} This task evaluates how an LM can understand and infer object recognition, visual information, and cultural or societal common sense, which are not typically described in textual information. CSQA is a multiple-choice question task that asks for the most plausible choice as an answer with some variants: JC-SQA is in Japanese, CommonsenseQA 2.0 (Talmor et al., 2021) is a more challenging dataset, ECQA (Aggarwal et al., 2021) requires explaining the process of deriving an answer, etc. There exist other types of commonsense tasks: COPA (Roemmele et al., 2011) and BalancedCOPA (Kavumba et al., 2019) ask about causal relationships between everyday events; SocialIQA (Sap et al., 2019b) asks about social common sense; PIQA (Bisk et al., 2020) evaluates procedural knowledge; HotpotQA (Yang et al., 2018) requires multi-hop inference; DROP (Dua et al., 2019) captures arithmetic operation capabilities; and tasks like understanding language information (Liu et al., 2022b; Kocijan et al., 2023; Sakaguchi et al., 2021; Wang et al., 2019), understanding causal relationships within documents (Mostafazadeh et al., 2020, 2016; Zhang et al., 2018; Huang et al., 2019; Ostermann et al., 2018; Smirnov, 2019), and CommonGen (Lin et al., 2020), which asks to generate common sentences from given keywords. The above datasets primarily focus on English, but there exist datasets in Japanese (Omura et al., 2020; Takahashi et al., 2019; Hayashibe, 2020), Chinese (Xu et al., 2021, 2020; Wang et al., 2022), Russian (Shavrina et al., 2020; Taktasheva et al., 2022), and Indonesian (Koto et al., 2022). For multilingual datasets, most are extended versions of existing ones through translation, such as X-COPA (Ponti et al., 2020) from COPA, X-CSQA (Lin et al., 2021) from CSQA, and X-CODAH (Lin et al., 2021) from CODAH (Chen et al., 2019). A few datasets, such as TyDiQA (Clark et al., 2020), are created for each language from scratch.

\subsection*{Multilingual datasets} When focusing on the evaluation of multilingual performance of LMs,
the evaluation datasets are almost exclusively created through three methods, as shown in Table 2: (1) Translation from existing datasets in a major language, e.g., English (Lin et al., 2021; Ponti et al., 2020; Conneau et al., 2018; Artetxe et al., 2020; Yang et al., 2019); (2) Compilation of similar tasks across multiple languages (Zhang et al., 2023c; Hu et al., 2023; Adelani et al., 2022; Roy et al., 2020; Malmasi and Dras, 2015); (3) Creation from multilingual resources following the same dataset creation process (Keung et al., 2020; Huang et al., 2020; Buchholz and Marsi, 2006; Clark et al., 2020; Schwenk and Li, 2018; Kabra et al., 2023). However, (1) translated datasets often do not account for language-specific culture, knowledge, common sense, or linguistic phenomena, leading to a bias towards the background of the source language (Hu et al., 2021; Lin et al., 2021; Acharya et al., 2020; Clark et al., 2020; Park et al., 2021; Kurihara et al., 2022). (2) Simply compiling datasets curated for each individual language could allow the evaluation of language-specific knowledge and common sense. However, it is difficult to align tasks across languages since most tasks differ in their creation methods data sources or philosophies. Thus, it just leads to evaluating the transfer capability among comparable tasks, and not evaluating the true transfer capabilities across languages. Therefore, (3) only the datasets created from multilingual resources can enable the evaluation of language transfer capability, considering the differences in language-specific knowledge and common sense. Nevertheless, the manual creation of such datasets is limited by the availability of annotators and financial costs.

### Dataset creation with LMs
The superior performance of generative language models allows to create datasets automatically. SWAG (Zellers et al., 2018) and HellaSwag (Zellers et al., 2019) have created answer choice options through the output of LMs. Such efforts have also been extended to use LMs for data augmentation (Staliūnaitė et al., 2021; Kumar et al., 2019, 2020; Lee et al., 2021). WANLI (Liu et al., 2022a), created from MNLI (Williams et al., 2018), employs GPT-3 (Brown et al., 2020) for adversarial data augmentation with manual checks to create challenging datasets. Some studies propose methods to manually check quality of LMs generation results (Tekiroğlu et al., 2020; Yuan et al., 2021; Wiegreffe et al., 2022; Wang et al., 2021a; Li et al., 2023). Additionally, there are attempts to create datasets from scratch with emergent abilities of LMs, without using any examples (He et al., 2022; Wang et al., 2021b; Schick and Schütze, 2021; Meng et al., 2022; Ye et al., 2022). However, these studies have primarily focused on a single language, e.g., English. Recently, the outputs of language models themselves have been used to create datasets (Honovich et al., 2023; Shao et al., 2023; Sun et al., 2023; Peng et al., 2023) for instruction-tuning (Wei et al., 2022a). TarGen (Gupta et al., 2023) employs a single language model and splits the data generation process into multiple steps, inputting the suitable prompt for each step to ensure data diversity and reliability. Putri et al. (2024) focus on middle-resource (Indonesian) and low-resource (Sundanese) languages, and investigate whether LLMs can create culturally aware commonsense questions by comparing translation datasets and those generated by LLMs from scratch.

## 3 Datasets Creation

Our mCSQA construction involves three main steps (see Figure 2): extraction of sub-graphs from ConceptNet, creation of question and choice pairs with LMs, and verification of their quality by both LMs and humans. We basically follow the creation processes of CSQA and JCSQA, but modified to allow for unified processing to support multiple languages.

### 3.1 Extract Sub-Graphs from ConceptNet

ConceptNet is a graph knowledge base defined as a tuple, $G = (\mathcal{C}, \mathcal{R}, \mathcal{T})$, where $\mathcal{C}$ denotes a set of concept entities, $\mathcal{R}$ denotes a set of relations and $\mathcal{T}$ denotes a set of triples. Each triple is represented as $(s, r, t) \in \mathcal{T}$, where $s$ and $t \in \mathcal{C}$ are the source and target concept entities, respectively, and $r \in \mathcal{R}$ is the relation, and carry commonsense knowledge such as “(student, CapableOf, forget to do homework)”.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Knowledge Alignment Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>By translation</td>
<td>✗ ✓ ✓</td>
</tr>
<tr>
<td>Compilation of similar tasks</td>
<td>✓ ✗ ✓</td>
</tr>
<tr>
<td>From multilingual resources</td>
<td>✓ ✓ ✗</td>
</tr>
<tr>
<td>Ours</td>
<td>✓ ✓ ✓</td>
</tr>
</tbody>
</table>

Table 2: Categorize the multilingual datasets creation methods.
We extract subgraphs from ConceptNet, as per Figure 2-(a), that have three distinct concept entities derived from queries of concept entities and relations for each language. CSQA uses only forward queries \((s, r, ?)\), but, similar to JCSQA, we also utilize backward queries \((? , r, t)\). We name this subgraph as Question Sets (QSs). After extraction, we filter the QSs like CSQA and JCSQA, and apply unified filtering in mCSQA as follows:

1. Similar to CSQA, we retain only QSs that contain any types of the 22 relations.[2]
2. We filter out QSs where any of the concept entities consist of more than four words or only a single character[3].
3. We remove QSs where any pair of concept entities is connected by a ‘Synonym’ relation in ConceptNet, or where entities are substrings of each other.

After filtering with the above settings, we randomly selected 6,000 QSs for each language[4].

3.2 Create Questions with LMs

We employ the generative multilingual language model GPT-3.5[5] (Ouyang et al., 2022) to generate questions automatically to eliminate the human labor as done in CSQA and JCSQA.

Our construction process comprises three steps of ‘question generation’, ‘question refinement’ and ‘distractor augmentation’ as shown in Figure 2-(b). Our step differs from CSQA in the refinement step since we need to improve the question generation from LM.

We designed prompts and tuned optimized hyper-parameters for each step for LMs. The details of the prompts are described in Appendix D, and the hyper-parameters are shown in Table 3.

### Table 3: The hyper-parameters for each step

<table>
<thead>
<tr>
<th>Step</th>
<th>temperature</th>
<th>top_p</th>
<th>seed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating question sentences</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>Refining question sentences</td>
<td>0.7</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>Adding additional distractors</td>
<td>1.2</td>
<td>0.7</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Creating question sentences

For each QS, we generated question sentences by LMs where, for each of the three target concept entities, only one serves as the answer. The prompt for LMs was inspired by the JCSQA filtering process for question creation in which systematic filtering uses textual information. The key instructions are as follows:

- Avoid including words of the target entities in the question sentence.
- Avoid using superficial information such as character count.
- End the sentence with a question mark (\(?\)).
- Be an objective question sentence.
- Consists of only one sentence.

After generating questions with LMs, we removed any question sentences that do not follow
these instructions or contain inappropriate expressions through pattern matching⁶.

**Refining question sentences** LMs do not always generate appropriate outputs resulting in unnatural expressions or degeneration (Liu et al., 2022c; Honovich et al., 2023; Raunak et al., 2023; Lin et al., 2020; Madaan et al., 2023). Hence, inspired by the idea of output refinement (Liu et al., 2022c; Raunak et al., 2023; Madaan et al., 2023), we refine unnatural generated question sentences into natural ones using the LM again and remove inappropriate questions as done in the previous step. Table 4 shows the percentage of sentence refinement.

**Adding additional distractors** We added additional incorrect choices to make the task more difficult as done in CSQA and JCSQA, but we leverage LM, not crowd workers, to formulate distractors that seemed plausible or related to the questions. Here, we asked LM to generate two plausible distractors given the three choices of a question without question itself in order to separate the question generation and answering capabilities of LMs. There is a risk of generating duplicated choices or adding correct choices since question sentence itself is not fed in this process. Hence, we remove such questions through manual verification in Section 3.3.

**3.3 Question Quality Verification by LMs and Humans**

In CSQA and JCSQA, every question is manually verified to remove low-quality questions, such as those with multiple correct answers or without correct answers in the choices. However, due to the large number of questions, manually verifying every question is not practical. Thus, we leverage simple active learning methodologies for annotation (Liu et al., 2022a; Bartolo et al., 2022; Li et al., 2023; Kratzwald et al., 2020). As shown in Figure 2-(c), initially, the LM verifies whether the questions can be answered or not, and only those questions that the LM cannot answer are manually verified.

**Verification by LMs** The original questions can be categorized into three types: those questions 1) which are correctly answerable by LMs, 2) which are wrong answers by LMs, but humans can choose the correct one, 3) which are not answerable either by LMs or humans due to flaws in the question. Therefore, first, we identify the set of questions LMs can answer, and then manually verify the questions that LMs could not answer correctly to remove flawed questions.

**Verification by Humans** We hired two crowd workers per language via Amazon Mechanical Turk (MTurk)⁷. The crowd workers were presented with the question sentence, choices, and answer, and they were asked to verify if the answer could be concluded from the question and choices. We retained only those questions on which all crowd workers agreed.

**3.4 Data Splitting and Statistics**

Similar to CSQA and JCSQA, we randomly split the data for each language into training, development, and test sets with an 80/10/10 split. The mCSQA is evaluated by accuracy following the standard practice in CSQA and JCSQA. Additionally, in Section 3.3, questions that LMs can answer correctly are categorized as Easy, and those answerable by human judgment are categorized as Hard for development and test sets.

Table 5 shows the number of questions per language and split, and Figure 3 shows the percentage filtered at each step. The total cost per question is 0.002 dollars for mCSQA compared to 0.33 dollars for CSQA, reducing the cost to less than one dollar.

---

⁶We detected inappropriate expressions using https://platform.openai.com/docs/guides/moderation.

⁷There are workers for each language on MTurk (Pavlick et al., 2014). We hired workers who have an approval rate greater than 90% with at least 50 approved HITs.
We verify that the mCSQA dataset is meaningful for evaluating the common sense reasoning capability of LMs by using various multilingual LMs.

4.1 Experimental Setup

Settings for LMs We used mBERT (Devlin et al., 2019), XLM-100 (Conneau and Lample, 2019), XLM-R (Conneau et al., 2020), and mDeBERTa-v3 (He et al., 2023) as encoder-based multilingual LMs, Llama2-70B (Touvron et al., 2023), GPT-3.5 (Ouyang et al., 2022), and GPT-4 (OpenAI et al., 2024) as decoder-based multilingual LMs for the experiments. Decoder-based LMs inferred with 0-shot and 3-shot settings. For detailed experimental settings, please refer to the Appendix A.

Settings for human baseline We followed the CSQA setting and randomly selected 100 questions each from the validation and test data for every language to measure the human baseline. We hired five new crowd-workers per language on MTurk. The answers were decided by a majority vote for each question.

4.2 Evaluation Results

Table 6 shows the main results. Focusing on the performance of zero-shot setting of GPT-3.5, which was used for dataset creation, we find that its performance is equivalent to or worse than that of Encoder models like XLM-R_{LARGE} and mDeBERTa-v3 except for German and Russian. When comparing the results of GPT-3.5 with GPT-4, the performance of GPT-3.5 is inferior for most languages to that of GPT-4. This indicates that the questions GPT-3.5 failed to answer correctly are those that cannot be answered by the knowledge of GPT-3.5, and it implies that the root cause is a lack of knowledge or reasoning capability of GPT-3.5. Furthermore, focusing on Decoder-based models, the results are better in the 3-shot setting than in the 0-shot in most cases. This trend was observed even with the GPT-3.5 used for question creation.

The results show that the prompting technique is effective for mCSQA in exploiting the reasoning capabilities of decoder-based LMs. The trend is similar to other commonsense reasoning tasks like CSQA (Qin et al., 2023; Chowdhery et al., 2023; Wei et al., 2022c; Brown et al., 2020; Dou and Peng, 2022), indicating that mCSQA can be equally effective as a dataset for commonsense reasoning tasks. Finally, when compared to the human baseline, there is a significant gap in the results of all LMs. Thus, it can be said that even when using LMs for question creation, it is possible to create a dataset with sufficient quality and difficulty for the LMs themselves.

5 Discussion

5.1 Comparison of Easy vs. Hard

We compare the accuracy of Easy and Hard sets for more fine-grained analysis. Figure 5 shows the results in the test split. GPT-3.5 and GPT-4 could choose the answer correctly in most cases for the Easy sets, but the accuracy is lower in the Hard sets with a significant gap when compared with human results; note that GPT-3.5 cannot answer there sets during the dataset creation. The other LMs also show a gap in evaluation accuracy with results for Hard sets being lower than those for Easy ones.

These results, specifically the trend observed
with GPT-3.5, show that even if LMs can create questions, it does not necessarily mean that they can answer them, and it entails that the question creation and answering are totally different capabilities. Therefore, we conclude that LMs can substitute for humans in parts of dataset creation processes from structured data and common sense reasoning task creations.

5.2 Evaluation of Multilingual LMs’ Cross-Lingual Transfer Capabilities

The cross-lingual transfer performance of multilingual LMs is often evaluated from English to other language directions due to linguistic resource reasons. The X-CSQA dataset (Lin et al., 2021), which consists solely of machine-translated questions from CSQA’s development and test splits, captures only the one-way cross-lingual transfer performance of LMs that were trained in English to evaluate their performance in other languages. In contrast, mCSQA supports the evaluation of cross-lingual language transfer performance in any directions among multilingual LMs that were trained in each of the eight languages.

Figure 6 shows the results of the multilingual LM, XLM-R_{LARGE}, which was fine-tuned in each of the eight languages separately and then evaluated across all eight languages on mCSQA, using the same settings as in Table 10. The results from Figure 6 show that, regardless of the language in which they were trained, cross-lingual transfer abilities are observed in most cases for any languages given the relative lower drop of performance when compared with the monolingual performance. Moreover, in the Easy sets, the drop is within 10% for most language pairs, while in the Hard sets, it ...
The language transfer performance of XLM-R\textsubscript{LARGE}. The y-axis indicates the languages in which the model was fine-tuned, while the x-axis indicates the languages used for evaluation. It shows the percentage of performance achieved when compared with the model trained and evaluated in the same language.

These results show that many languages observed improvements, especially in all cases in Easy sets. However, in Hard sets, some cases observed a decline in performance compared to the monolingual setting. Therefore, while training in a multilingual setting generally promotes accuracy improvement, multilingual training might lead to the loss of language-specific commonsense information for questions requiring more human commonsense. This analysis complements the previous reports (Dhamecha et al., 2021; Zhang et al., 2023a; Hu et al., 2021; Mueller et al., 2020) on the successes and failures of multilingual training.

Furthermore, Table 7 shows the evaluation results of cross-lingual performance in the unseen setting, where the model was not trained on the language for evaluation data. While some languages outperform the monolingual setting, overall results indicate that training with target language data consistently yields better outcomes. This suggests that target language data acts as the secret sauce for enhancing NLU performance. Therefore, it suggests that for language-specific deep knowledge and cultural understanding, language-transfer capability alone is insufficient, and training with datasets focused on language-specific knowledge is necessary.

### 5.4 Case Study for Improvement through Few-Shot Learning

As shown in Figure 5, GPT-3.5 correctly answers most questions in the Easy setting of mCSQA, but in the Hard setting, it fails to answer most questions in the 0-shot setting. This is because GPT-3.5 is used for quality filtering of mCSQA in Section 3.3, making it inherently unable to answer the questions in the Hard setting in the 0-shot setting. However,
in the 3-shot setting, it shows improvement for some questions. Table 8 shows examples of questions correctly answered in the 3-shot setting. Both examples in Table 8 are mainly due to the granularity of the answers. The 3-shot setting promotes answers at an appropriate granularity for questions that are difficult to judge due to inclusive relationships.

In the top example in Table 8, careful reading of the questions narrows down the answer choices. On the other hand, in the bottom example, considering various common knowledge in daily life helps to choose the most appropriate answer. Similar characteristics were observed for other languages as well. For more details, qualitative analyses of the mCSQA dataset are described in Appendix B.

6 Conclusion and Future Directions

We proposed an efficient and low-cost method for creating NLU datasets from structured data by utilizing generative LMs as an alternative to traditional human annotation, often crowdsourced. Inspired by CSQA and JCSQA, we created the multilingual commonsense reasoning task dataset, mCSQA, using GPT-3.5 from the structured multilingual knowledge base ConceptNet. We demonstrated that mCSQA is useful for evaluating the commonsense reasoning capabilities of LMs. We also analyzed the language-transfer capability beyond English with mCSQA and examined the language-specific learning from two aspects: question difficulty and language information. Moreover, our study has shown that the use of multilingual LMs enables the construction of multilingual datasets. Therefore, our method can significantly reduce human labor and financial costs.

In this study, we used a single multilingual LM, but since each step is independent, it is possible to replace the LM used in each step with another one. Furthermore, each step can be applied modularly to other methods, making it possible to use this method for creating multilingual datasets, such as those expanded through translation and manual refinement (Yanaka and Mineshima, 2022; Seo et al., 2022). We aim to extend this method to other types of commonsense reasoning tasks and NLU tasks, to efficiently create multilingual data and conduct a more comprehensive analysis of transfer capabilities across a broader range of tasks and languages.

We focused on language-specific commonsense, but languages are shared across various regions. For example, English is spoken in the United States, the United Kingdom, India, Australia, and many other regions each of which is geographically distant and diverse in terms of climate, food, and culture. Therefore, it will be necessary to create more detailed commonsense tasks that consider cultural differences rather than just language such as Kabra et al. (2023); Khanuja et al. (2024); Kim et al. (2024); Cao et al. (2024); Fung et al. (2024); Lee et al. (2024); Shwartz (2022); Hovy and Yang (2021); Yin et al. (2022); Shi et al. (2024). Our dataset construction method can be useful in creating various commonsense reasoning datasets that outgrow language limits.

7 Limitations

Data Resources The number of multilingual resources is significantly smaller than that of monolingual resources. Additionally, quality is not consistent, and there are imbalances in data volume across languages in these multilingual resources. In this study, we used ConceptNET, a multilingual knowledge base, and encountered these issues as well. For example, despite Spanish having a significantly higher entity count, it obtained fewer QSs
due to its inability to meet the required conditions because of ConceptNet’s sparsity issue, and thus it was excluded from the language selection for mCSQA. We believe these problems can be addressed through the automatic generation of knowledge bases (Zhang et al., 2020b,a; West et al., 2022; Ide et al., 2023; Nguyen et al., 2023) and data augmentation techniques for knowledge bases (Malaviya et al., 2020; Ju et al., 2022; Wu et al., 2023; Shen et al., 2023), supported by their pre-trained knowledge (Sakai et al., 2023).

Dataset Quality In this study, we used GPT-3.5 and simple prompts for data creation. Therefore, there is room for improvement in the selection of LMs and the refinement of prompts. In a pilot study, we tried using GPT-4 and recognized that it is more capable of creating datasets. However, due to budgetary constraints, we have used GPT-3.5 in this study. Thus, it may become possible to create higher quality datasets at a lower cost when the API prices decrease or by switching to other strong LMs such as Gemini (Team et al., 2023), Mixtral (Jiang et al., 2024), Llama (Touvron et al., 2023), phi (Abdin et al., 2024) or Qwen (Bai et al., 2023). Additionally, employing prompt strategies that leverage the capabilities of LMs, such as Chain of Thought (CoT) (Wei et al., 2022c), Tree of Thought (ToT) (Yao et al., 2023a) and ReAct (Yao et al., 2023b), could potentially lead to the production of higher quality datasets.

Verification of dataset quality by humans The human baselines decreased the evaluation result under the Hard sets compared to Easy sets in Figure 5. Therefore, there exists a risk that the Hard sets include flawed questions, even after manual quality verification. The JCSQA has pointed out that such low-quality questions are included in CSQA, and we have confirmed that they are similarly present in JCSQA. Thus, it is extremely difficult to completely eliminate such low-quality questions. Comparing the percentage of data removed in quality verification, CSQA is 25% (3995/16242), and JCSQA is 19% (2643/13906), whereas for mCSQA, it is 27% (604/2226) and 23% (599/2510) respectively, according to Figure 3 and referenced in their respective papers. This indicates that the filtering ratios are almost comparable when compared to those, showing that this is not a problem unique to mCSQA. Therefore, reinforcing quality verification to filter out low-quality questions is a challenge in our future studies. However, as Figure 4 shows, since more than half of the costs are already spent on manual quality verification, simply hiring more crowd workers would not be a better choice. Hence, exploring more efficient methods of quality verification as an alternative or to assist crowd workers in the future is necessary.

Human baseline The experimental results include human baselines using small sets of samples. However, Tedeschi et al. (2023) argue that human baselines may lack reliability due to factors such as the payment issues for crowd workers and the impact of random samples. Therefore, it should be noted that the human baselines in this study are merely reference values.

8 Ethical Considerations

License The mCSQA dataset was created entirely from the outputs of GPT-3.5 and is therefore subject to OpenAI’s license terms. OpenAI assigns to us all rights, title, and interest in and to the output. As a result, we are retaining the ownership rights. There are no restrictions on distributing the datasets, but using OpenAI’s model output to develop models that compete with OpenAI is prohibited. However, it’s possible that these terms may change, and there may be a need to impose distribution restrictions depending on the terms.

Moderation We eliminated potentially harmful questions such as violence, sexual content, and hate speech by screening through OpenAI moderation APIs. However, in the commonsense reasoning dataset, it cannot be guaranteed that it does not include questions that contain societal biases as collective knowledge. This issue has also been pointed out in existing datasets such as CSQA, JCSQA, and other commonsense reasoning datasets, and it is challenging to determine what is considered commonsense constitutes bias (Rajani et al., 2019; Sap et al., 2020; Bauer et al., 2023; An et al., 2023). If you encounter any harmful questions that contain such biases, please report them.

Translation Tool We used DeepL Pro to translate the example sentence, especially Table 1, to avoid arbitrary translation. The copyright of the translation sentences belongs to us.

9 https://openai.com/policies/terms-of-use
10 https://platform.openai.com/docs/guides/moderation
11 https://www.deepl.com/pro-license
References


Shourya Aggarwal, Divyanshu Mandowara, Vishwa-


Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalkrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitriy Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can, not as i say: Grounding language in robotic affordances.


Du Chen, Yi Huang, Xiaopu Li, Yongqiang Li, Yongqiang Liu, Haihui Pan, Leichao Xu, Dacheng Zhang, Zhiipeng Zhang, and Kun Han. 2024. Orion-14b: Open-source multilingual large language models.


Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinping Hu, Lidia S. Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? A comprehensive evaluation.


Yi Fung, Ruining Zhao, Jae Doo, Chenkai Sun, and Heng Ji. 2024. Massively multi-cultural knowledge acquisition & lm benchmarking.


Tannon Kew, Florian Schottmann, and Rico Sennrich. 2023. Turning english-centric llms into polyglots: How much multilinguality is needed?

Simran Khamuja, Sathyanarayanan Ramamoorthy, Yueqi Song, and Graham Neubig. 2024. An image speaks a thousand words, but can everyone listen? on image transcreation for cultural relevance.


Nayeon Lee, Chani Jung, Junho Myung, Jiho Jin, Jose Camacho-Collados, Juho Kim, and Alice Oh. 2024. Exploring cross-cultural differences in english hate speech annotations: From dataset construction to analysis.


Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In Advances in Neural Information Processing Systems.


Sungjoon Park, Kyung Moon, Sungdong Kim, Won-Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Jun-seong Kim, Youngsook Song, Taehwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo


Rifki Afina Putri, Faiz Ghifari Haznitra, Dea Adhista, and Alice Oh. 2024. Can ilm generate culturally relevant commonsense qa data? case study in indonesian and sundanese.


6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.


Ekaterina Taktashvea, Alena Fenogenova, Denis Shevelev, Nadezhda Katricheva, Maria Tikhonova, Albina Akhmetgareeva, Oleg Zinkevich, Anastasiia


Punta Cana, Dominican Republic. Association for Computational Linguistics.


Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension.

A Details of the Experimental Settings

We used mBERT (Devlin et al., 2019), XLM-100 (Conneau and Lample, 2019), XLM-R (Conneau et al., 2020), and mDeBERTa-v3 (He et al., 2023) as encoder-based multilingual LMs, Llama2-70B (Touvron et al., 2023), GPT-3.5 (Ouyang et al., 2022), and GPT-4 (OpenAI et al., 2024) as decoder-based multilingual LMs for the experiments. Table 9 shows the details of the LMs. Encoder-based LMs were fine-tuned following the settings in Table 10. Decoder-based LMs inferred with 0-shot and 3-shot settings\(^\text{12}\) with a fixed seed value. For GPT-3.5 and GPT-4, top_p and temperature were set to 0 to achieve as deterministic outputs as possible. For Llama2-70B, output was generated greedy, and outlines (Willard and Louf, 2023) were used to fix the output format.

B Qualitative Analysis of mCSQA

Table 11 shows examples of mCSQAs for each language. The examples in Table 11 are accompanied by English translations using DeepL\(^\text{13}\) to avoid arbitrary translation.

B.1 Can Multilingual LMs Take into Account Language-specific Knowledge?

Case study When we examine some cases in Table 11, such as the examples from the Dutch Hard set and the Russian Hard set, we find that the English translations contain duplications among the question choices. However, these duplications arise not from differences in tense or conjugation, but from semantic differences unique to each language, which a native speaker, equipped with language-specific knowledge and common sense, could easily distinguish. Furthermore, in the case of the German Easy sets, knowledge of Germany’s unique education system is required, which might be challenging for those unfamiliar with it. Yet, for German speakers, it is common knowledge that such...
violations of their names or can suggest the names of objects. The distractor ‘Wakame’ is known as the name of a character from the long-running, famous anime ‘Sazae-san’ but not as a singer, thus serving its purpose as a distractor in this question effectively. Similarly, if there were a choice like ‘いくら’ (common meaning: red caviar; pronounced: ikura), the plausibility of choice in this question might have been divided. Recently, ‘ikura’ has become a popular name, associated with a member of ‘Yoasobi’, a popular artist group among young people. Adding such a choice would confuse the choice of the correct answer because both choices are plausible, so it would not serve effectively as a distractor. This case shows that the choices can define the scope of common sense, thus making the question effective in evaluating common sense accurately.

B.2 The Relationship between Knowledge, Culture, Commonsense, and Social Bias

B.2.1 What is the Commonsense?

As can be seen from Table 11 and the discussions in section B.1, language-specific common sense is closely related to knowledge and culture. The ConceptNet used in this study does not limit the scope of common sense and deals with a wide range of common sense, enabling the inclusion of questions from various backgrounds into mCSQA, following the same trend as CSQA and JCSQA.

Generally, commonsense not based on the specific culture or knowledge of a language is likely to be a common understanding across all languages, making such problems potentially answerable through the language-transfer ability of multilingual LMs. However, as shown in Table 1, the granularity of actions, events, and behaviors differs by language, which can be considered to be influenced by the cultural background of the language area.

This study focuses on language-specific common sense that cannot be addressed by translations of datasets from other languages, and the culture and knowledge included in them are shared among native speakers. Therefore, answering questions that require language-specific backgrounds necessitates a certain level of knowledge and culture specific to each language. However, content that is too specialized falls outside the scope of common sense, and common sense and backgrounds vary among individuals. Therefore, we emphasize the precision of coverage in the manual question quality verification steps and employ a majority vote baseline to avoid overly relying on specific knowledge or culture.

In this way, questions were created that have language-specific common sense which is general for native speakers but not too specialized. If there was a need to create questions asking for knowledge specialized in specific fields, other knowledge bases such as ATOMIC (Sap et al., 2019a), and CCSK (Nguyen et al., 2023) could be used. However, this study focused on multilingual performance, deeming ConceptNet appropriate for mCSQA.

B.2.2 Is Commonsense Social Bias?

Since commonsense includes implicit cognition, it may contain social and cultural biases, and some methods for the removal of explicit and implicit social biases have been proposed (Sap et al., 2020; Field and Tsvetkov, 2020; Huang et al., 2021; Lent and Søgaard, 2021; Emelin et al., 2021; Bauer et al., 2023).

Social Chemistry 101 (Forbes et al., 2020), BBQ (Parrish et al., 2022), and SODAPOP (An et al., 2023) have been proposed for identifying biases within models or for bias detection using LMs. However, it remains challenging to address situations where biased thinking may only emerge when considering multiple-choice QA, where bias does not occur in isolation.

The definition of bias and common sense changes over time and varies from society to society, and what is considered common sense can shift to bias (Lee et al., 2023). Therefore, regular updates to the commonsense reasoning datasets are necessary. Our method for generating commonsense reasoning task datasets using LMs allows for low-cost update operations, making it possible to adapt to the changing boundaries between common sense and bias over time. However, this does not fundamentally address the inclusion of bias in datasets. Moreover, such issues require a deep chain of semantic thinking for resolution, making filtering based on textual information inappropriate. Therefore, it is necessary to develop methods to remove potential biases in commonsense reasoning task datasets in future work.
C Discoveries about the LMs Capabilities

C.1 Can LMs Create Questions including Commonsense?

**Generation capability** CommonGen (Lin et al., 2020) is one of the commonsense reasoning datasets that evaluates whether it is possible to create commonsense sentences from a given set of keywords. According to the leaderboard of CommonGen\(^\text{20}\), the performance of GPT-3.5 used in our dataset creation demonstrates a capability for generating commonsense sentences comparable to those written by humans. However, there is still room for improvement in aspects such as word order. Therefore, we introduced refinement steps to encourage corrections in word order and other errors. Since language models have high performance in Grammar Error Correction (GEC) (Loem et al., 2023; Sottana et al., 2023; Coyne et al., 2023; Kaneko and Okazaki, 2023; Kwon et al., 2023), combining sentence generation from keywords with GEC capabilities in a pipeline helps to compensate for the weaknesses of language models. We believe that the quality of mCSQA questions is at least not inferior to those created by crowd-workers. The capability of multilingual LMs to create commonsense sentences from given keywords has also been demonstrated in the Korean CommonGen (Seo et al., 2022), indicating that it is possible to generate commonsense sentences multilingually.

**Ensuring the quality of questions** In this study, we have created commonsense reasoning dataset questions using keywords extracted from ConceptNet. Therefore, the language-specific knowledge and commonsense for each language are guaranteed by ConceptNet. Moreover, the LM creates questions following the given instructions through its emergent capabilities from each keyword. To enhance the language-specific performance of the multilingual LM for each language, we have created prompts for each language in this study. As can be seen from the discussion in Section B.1 and Table 11, it has become possible to generate questions that possess language-specific knowledge. One of the reasons for the capability to create questions with language-specific knowledge may be attributed to the training data of the LM. For example, Wikipedia, one of the common training data for LMs, has each language which contains descriptions of knowledge unique to that language, so by posing questions in each language, it is thought that knowledge specific to each language is invoked, enabling the generation of questions based on the knowledge of each language. However, this is a hypothesis, and further analysis will be necessary for verification in future work. Moreover, we have added distractors in addition to the keywords used for generating the question, which means that even if a question can be generated, it may not necessarily be answerable. Furthermore, questions that cannot be answered have been removed, thus ensuring the difficulty and answerability of the QA.

C.2 Multilingual Capabilities

**Is polyglot template effective?** We translated the prompt to use question generation for each language and tuned it to convey the same meaning in each language in Section D aimed to emergence the language-specific knowledge. However, it is known that current generative LMs have mainly trained on English, which is better performance for queries made in English. However, several studies (Ahn et al., 2022; Shi et al., 2023; Wei et al., 2022b; Awasthi et al., 2023; Kasai et al., 2023; Jin et al., 2023) show enough performance even if multilingual queries. Note that the reported performance focuses on the ability to answer specific tasks on benchmarks and does not evaluate the emergent multilingual ability, especially question generation. Nevertheless, Whitehouse et al. (2023) shows that the text generation capability beyond English. As shown in Table 10, we were able to generate questions containing language-specific knowledge from the given keywords as intended by using prompts translated into each language. We were able to generate questions that require deep reasoning, including cultural backgrounds and language-specific pronunciation information as shown in Section B.1. Therefore, we conclude that using prompts tailored for each language is effective.

**Is GPT-3.5 Multilingual LM?** Yes, some studies (Lai et al., 2023; Armengol-Estapé et al., 2022; Zhang et al., 2023b) have indeed examined multilingual performance, and the training data also includes multilingually\(^\text{21}\). Therefore, the multilingual capabilities of GPT-3.5, GPT-4, and Llama used in our experiment have also been evalu-
ated (Ahuja et al., 2023; Schott et al., 2023; Chen et al., 2024), leading us to consider these as multilingual LMs. However, they still rely predominantly on information from Western norms (Cao et al., 2023; Arora et al., 2023; Havaldar et al., 2023), making this issue an ongoing challenge to be addressed in the future.

Exhortation to multilingual instruction-tuning dataset. Instruction-tuning (Wei et al., 2022a; Longpre et al., 2023; Chung et al., 2022; Wang et al., 2023) can enhance the quality of LMs, e.g. ability to follow instructions and NLU performance. However, in Section 2, the current multilingual datasets include those created through translation, which means that instruction-tuning using such data may not lead to the acquisition of data bias or language-specific knowledge. Given these considerations, the multilingual instruction-tuning data (Kew et al., 2023; Singh et al., 2024) proposed recently often utilize datasets created through translation, leading to the occurrence of the aforementioned issues to a considerable extent. Consequently, the effectiveness of such instruction-tuning may be diminished. For commonsense reasoning tasks in multilingual instruction-tuning datasets, they sometimes use X-CSQA (Lin et al., 2021). However, since it cannot handle language-specific knowledge or commonsense effectively, it is preferable to use data created from scratch, like mCSQA. Currently, due to data resource issues, reliance on translated data is inevitable, but we hope that in the future, it will be replaced by language-specific data.

C.3 Hard Sets are Truly Hard?

The Hard sets consist of questions that the LM used for question creation could not answer, thus reflecting the characteristics of that LM. However, despite the influence of specific LM’s character, a performance decline in the Hard sets compared to the Easy sets was observed across all models. Therefore, while the strict division of sets depends on the model, it has become clear that there is a similar trend across LMs as a whole. For this reason, scoring is conducted without distinguishing between Easy and Hard, using a total score for the entire set, which allows for the absorption of differences due to the models.

C.4 Generation Bias and Annotation Artifacts

It has been pointed out that datasets created by LMs contain generation bias (Omura et al., 2020; Zellers et al., 2019; Tamborrino et al., 2020), and those created by crowd-workers include specific patterns (Annotation Artifacts) (Gururangan et al., 2018; Chen et al., 2019; Omura et al., 2020). Annotation artifacts, in particular, have been noted in natural language inference tasks such as MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015), where choices can be easily distinguished by superficial words like “not”.

However, Tamborrino et al. (2020) show that the impact of Annotation Artifacts is not present in the CSQA task. Similarly, in this study, we have separated question generation ability and answering ability during the question generation process and shuffled the options, so there are no clues included in the dataset. Moreover, we create Hard sets, even if such biased questions existed, the evaluation is conducted without these biases, allowing for an evaluation that removes these biases.

D Prompts for Creating mCSQA

The prompts used for creating mCSQA are presented as follows: English in Table 12, Japanese in Table 13, Chinese in Table 14, German in Table 15, Portuguese in Table 16, Dutch in Table 17, French in Table 18 and Russian in Table 19.

In each prompt template, the words within the curly brackets are replaced with data-specific terms before input to the LM. Furthermore, as discussed in Section C.2, each template was translated exactly to elicit language-specific knowledge of each language. The translations were carried out using both GPT-3.5 and DeepL to ensure there were no semantic differences, with manual fixing applied as needed. We use the OpenAI API’s JSON mode has facilitated the retrieval of generation results.

Our findings as a tip, when inputting structured data such as keywords, doing so in a format similar to a programming code like list type, allows us to obtain results that more following the prompt instructions. This improvement can be attributed to the LM’s learning to enhance coding abilities, which is believed to have improved its recognition capabilities.

22https://peps.python.org/pep-0498/
23https://platform.openai.com/docs/guides/text-generation/json-mode
<table>
<thead>
<tr>
<th>Lang</th>
<th>Question</th>
<th>Correct</th>
<th>Distractions</th>
<th>Choices</th>
<th>Additional Distractors</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>Easy</td>
<td>If a cat is feeling irritated, what might it do?</td>
<td>scratch if annoyed</td>
<td>monkey</td>
<td>junglefish</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>Which animal is known for its playful behavior and agile movements?</td>
<td>look out window</td>
<td>fish with paw</td>
<td>chase a toy</td>
</tr>
<tr>
<td>JA</td>
<td>Easy</td>
<td>音を開き分けるためには何をしますか？</td>
<td>耳を挙げる</td>
<td>学習する (learn)</td>
<td>書き取る (write)</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>日本の女性歌手で、自身の楽曲の作詞・作曲も手がける人気アーティストは誰ですか？</td>
<td>あゆ</td>
<td>どういう</td>
<td>わかめ</td>
</tr>
<tr>
<td>ZH</td>
<td>Easy</td>
<td>你在考试前应该做什么？</td>
<td>回家复习 (go home and study)</td>
<td>聊天 (chatting)</td>
<td>作弊 (cheat)</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>在感情关系中，最令人痛苦的事情是什么呢？</td>
<td>嫉妒</td>
<td>花花钱</td>
<td>心碎</td>
</tr>
<tr>
<td>DE</td>
<td>Easy</td>
<td>Welche Art von weiterführender Schule bereitet Schüler auf das Abitur vor?</td>
<td>gymnasium</td>
<td>gesamtschule</td>
<td>fachoberschule</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>Was ist die richtige Bezeichnung für das langsame Abwärtsbewegen auf einer schiefen Ebene?</td>
<td>hinabgleiten</td>
<td>hinabfliegen</td>
<td>dahinab</td>
</tr>
<tr>
<td>PT</td>
<td>Easy</td>
<td>Como demonstrar afeto a um animal de estimação?</td>
<td>fazer carinho (cuddle)</td>
<td>alegar a vida (combing)</td>
<td>pentelhar (brighten up life)</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>Qual a ação que um coelho pode fazer para se mover rapidamente?</td>
<td>pular</td>
<td>orientando</td>
<td>segurar</td>
</tr>
<tr>
<td>NL</td>
<td>Easy</td>
<td>Kunt u mij vertellen wat gokken is?</td>
<td>kansspel (game of chance)</td>
<td>gelijkspel (draw)</td>
<td>steekspel (joust)</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>Kunt u uitleggen wat een veelvoorkomend begrip is dat verwijst naar iets wat algemeen geaccepteerd of verspreid is in een samenleving?</td>
<td>gemeengoed</td>
<td>gemeenschap</td>
<td>gemeenplaats</td>
</tr>
<tr>
<td>FR</td>
<td>Easy</td>
<td>Quelle unité de temps correspond à une période de vingt-quatre heures ?</td>
<td>jour</td>
<td>décade</td>
<td>siècle</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>Quelle partie du corps utilise-t-on pour saisir des objets de petite taille ?</td>
<td>doigt</td>
<td>annulaire</td>
<td>auriculaire</td>
</tr>
<tr>
<td>RU</td>
<td>Easy</td>
<td>Какое время года обычно связаны с праздниками Нового года и Рождества?</td>
<td>зима</td>
<td>весна</td>
<td>осень</td>
</tr>
<tr>
<td>Hard</td>
<td></td>
<td>Какой звук издает довольный кот?</td>
<td>урчание (purr)</td>
<td>заурчать (rumble)</td>
<td>прорычать (purr)</td>
</tr>
</tbody>
</table>

Table 11: The examples of mCSQA. The English translations are all machine-translated by DeepL. The translated results sometimes are aggregated into one English word due to ignoring source language-specific subtle meaning differences caused by machine translation. This aggregation has also been observed in X-CSQA, which was created using machine translation of CSQA. Hence, X-CSQA could not evaluate fine-grained, language-specific knowledge for each language, but mCSQA can evaluate it because it is created from scratch for each language.
Steps Prompt (English)

Create question sentences Please create a multiple-choice question with the following conditions:
(a) The only correct answer is "[correct]".
(b) The incorrect answers are "[distractor1]", "[distractor2]".
(c) Do not use the words "[correct]", "[distractor1]", "[distractor2]" in the question.
(d) Avoid using superficial information, such as character count.
(e) The question ends with a question mark (?).
(f) It should be an objective question that can be sufficiently answered with common sense knowledge alone.
(g) The question must be a simple and short sentence consisting of only one sentence.

Question:

Refine question sentences If the original sentence is semantically and grammatically correct, repeat it; if it is unnatural, please rewrite it into a correct and fluent sentence.

Add additional distractors Please only add two plausible and natural choices and save them in ['additional_choice':[]}.

Verify Qualities Please select only one alphabet as the answer from the Answer Choices and save it in the format: {'answer': selected_answer}.

Q: {question}
Answer Choices: (A) {choice_a} (B) {choice_b} (C) {choice_c} (D) {choice_d} (E) {choice_e}

Table 12: The prompt templates used to create the mCSQA in the English version.

Steps Prompt (Japanese)

Create question sentences 以下の条件を満たす選択肢付きのクイズ問題を作成してください。
(a) 正解は"[correct]"のみです。
(b) 不正解は"[distractor1]", "[distractor2]"です。
(c) 問題文に"[correct]", "[distractor1]", "[distractor2]"という単語を使わないでください。
(d) 文字数などの表面的な情報の使用を避けてください。
(e) 問題文は疑問符 (?) で終わります。
(f) 一般常識だけでなく、文脈で十分に答えられる客観的な問題である必要があります。
(g) 問題文は一文のみから成る単純で短い文でなければなりません。

問題:

Refine question sentences 元の文が意味的・文法的に正しい場合は繰り返す、不自然な場合は正しい流暢な文へ書き換えてください。

Add additional distractors もっともらしい自然な選択肢を2つだけ追加し、それらを['additional_choice':{}]に保存してください。

Verify Qualities Answer Choicesから解答となるアルファベットを1つだけ選び、次の形式で保存してください：{'answer': selected_answer}。

Q: {question}
Answer Choices: (A) [choice_a] (B) [choice_b] (C) [choice_c] (D) [choice_d] (E) [choice_e]

Table 13: The prompt templates used to create the mCSQA in the Japanese version.
请根据以下条件创建一个多项选择题:
(a) 唯一正确答案是“{correct}”。
(b) 错误答案是“{distractor1}”, “{distractor2}”。
(c) 问题中不得使用“{correct}”, “[distractor1]”, “[distractor2]”这些词。
(d) 避免使用表面信息，如字符数。
(e) 问题以问号(?)结束。
(f) 它应该是一个客观的问题，仅凭常识就能充分回答。
(g) 问题必须是一个简单且短的句子，仅由一句话组成。

问题:

如果原句在语义和语法上正确，请重复它；如果不自然，请将其改写为正确流畅的句子。

请只添加两个合理且自然的选择，并将它们保存在’additional_choice’:[]中，
“[choice1]”, “[choice2]”, “[choice3]”

请从Answer Choices中仅选择一个字母作为答案，并以以下格式保存：’answer’: selected_answer。

Q: {question}
Answer Choices: (A) {choice_a} (B) {choice_b} (C) {choice_c} (D) {choice_d} (E) {choice_e}

Table 14: The prompt templates used to create the mCSQA in the Chinese version.
Table 16: The prompt templates used to create the mCSQA in the Portuguese version.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Prompt (Portuguese)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create question sentences</td>
<td>Por favor, crie uma pergunta de múltipla escolha com as seguintes condições: (a) A única resposta correta é &quot;{correct}&quot;. (b) As respostas incorretas são &quot;{distractor1}&quot;, &quot;{distractor2}&quot;. (c) Não use as palavras &quot;{correct}&quot;, &quot;{distractor1}&quot;, &quot;{distractor2}&quot; na pergunta. (d) Evite usar informações superficiais, como a contagem de caracteres. (e) A pergunta termina com um ponto de interrogação (?). (f) Deve ser uma pergunta objetiva que pode ser suficientemente respondida apenas com conhecimento de senso comum. (g) A pergunta deve ser uma frase simples e curta, consistindo de apenas uma frase. Pergunta:</td>
</tr>
</tbody>
</table>

Refine question sentences | Se a frase original estiver semanticamente e gramaticalmente correta, repita-a; se for pouco natural, por favor, reescreva-a em uma frase correta e fluente. |

Add additional distractors | Por favor, adicione apenas duas escolhas plausíveis e naturais e salve-as em {'additional_choice':[]}.

Verify Qualities | Por favor, selecione apenas uma letra como resposta das Answer Choices e salve no formato: {'answer': selected_answer}. |

Q: {question} Answer Choices: (A) {choice_a} (B) {choice_b} (C) {choice_c} (D) {choice_d} (E) {choice_e} |

Table 17: The prompt templates used to create the mCSQA in the Dutch version.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Prompt (Dutch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create question sentences</td>
<td>Maak alstublieft een meerkeuzevraag met de volgende voorwaarden: (a) Het enige juiste antwoord is &quot;{correct}&quot;. (b) De onjuiste antwoorden zijn &quot;{distractor1}&quot;, &quot;{distractor2}&quot;. (c) Gebruik de woorden &quot;{correct}&quot;, &quot;{distractor1}&quot;, &quot;{distractor2}&quot; niet in de vraag. (d) Vermijd het gebruik van oppervlakkige informatie, zoals het aantal tekens. (e) De vraag eindigt met een vraagteken (?). (f) Het moet een objectieve vraag zijn die alleen met algemene kennis voldoende beantwoord kan worden. (g) De vraag moet een eenvoudige en korte zin zijn die uit slechts één zin bestaat. Vraag:</td>
</tr>
</tbody>
</table>

Refine question sentences | Als de originele zin semantisch en grammaticaal correct is, herhaal deze dan; als het onnatuurlijk is, herschrijf het dan naar een correcte en vloeiende zin. |

Add additional distractors | Voeg alstublieft slechts twee aannemelijke en natuurlijke keuzes toe en sla ze op in {'additional_choice':[]}.

Verify Qualities | Selecteer alstublieft slechts één letter als antwoord uit de Answer Choices en sla het op in het formaat: {'answer': selected_answer}. |

Q: {question} Answer Choices: (A) {choice_a} (B) {choice_b} (C) {choice_c} (D) {choice_d} (E) {choice_e} |
<table>
<thead>
<tr>
<th>Steps</th>
<th>Prompt (French)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create question sentences</td>
<td>Veuillez créer une question à choix multiples avec les conditions suivantes :</td>
</tr>
<tr>
<td></td>
<td>(a) La seule bonne réponse est [&quot;{correct}&quot;].</td>
</tr>
<tr>
<td></td>
<td>(b) Les réponses incorrectes sont [&quot;{distractor1}&quot;, &quot;{distractor2}&quot;].</td>
</tr>
<tr>
<td></td>
<td>(c) Ne pas utiliser les mots [&quot;{correct}&quot;, &quot;{distractor1}&quot;, &quot;{distractor2}&quot;] dans la question.</td>
</tr>
<tr>
<td></td>
<td>(d) Évitez d’utiliser des informations superficielles, telles que le nombre de caractères.</td>
</tr>
<tr>
<td></td>
<td>(e) La question se termine par un point d’interrogation (?).</td>
</tr>
<tr>
<td></td>
<td>(f) Il doit s’agir d’une question objective qui peut être suffisamment répondue avec le seul sens commun.</td>
</tr>
<tr>
<td></td>
<td>(g) La question doit être une phrase simple et courte composée d’une seule phrase.</td>
</tr>
<tr>
<td></td>
<td>Question :</td>
</tr>
<tr>
<td>Refine question sentences</td>
<td>Si la phrase originale est correcte sémantiquement et grammaticalement, répétez-la ;</td>
</tr>
<tr>
<td></td>
<td>si elle est peu naturelle, veuillez la reformuler en une phrase correcte et fluide.</td>
</tr>
<tr>
<td></td>
<td>{question}</td>
</tr>
<tr>
<td>Add additional distractors</td>
<td>Veuillez ajouter seulement deux choix plausibles et naturels et les enregistrer dans <code>{additional_choice</code>:[[]].</td>
</tr>
<tr>
<td></td>
<td>[&quot;{choice1}&quot;, &quot;{choice2}&quot;, &quot;{choice3}&quot; ]</td>
</tr>
<tr>
<td>Verify Qualities</td>
<td>Veuillez sélectionner uniquement une lettre comme réponse parmi les Answer Choices et enregistrez-la dans le format : <code>{answer</code>: selected_answer}.</td>
</tr>
<tr>
<td></td>
<td>Q: {question}</td>
</tr>
<tr>
<td></td>
<td>Answer Choices: (A) {choice_a} (B) {choice_b} (C) {choice_c} (D) {choice_d} (E) {choice_e}</td>
</tr>
</tbody>
</table>

Table 18: The prompt templates used to create the mCSQA in the French version.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Prompt (Russian)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create question sentences</td>
<td>Пожалуйста, создайте вопрос с несколькими вариантами ответа с учетом следующих условий:</td>
</tr>
<tr>
<td></td>
<td>(a) Единственный правильный ответ - [&quot;{correct}&quot;].</td>
</tr>
<tr>
<td></td>
<td>(b) Неправильные ответы - [&quot;{distractor1}&quot;, &quot;{distractor2}&quot;].</td>
</tr>
<tr>
<td></td>
<td>(c) Не используйте слова [&quot;{correct}&quot;, &quot;{distractor1}&quot;, &quot;{distractor2}&quot;] в вопросе.</td>
</tr>
<tr>
<td></td>
<td>(d) Избегайте использования поверхностной информации, такой как количество символов.</td>
</tr>
<tr>
<td></td>
<td>(e) Вопрос заканчивается вопросительным знаком (?).</td>
</tr>
<tr>
<td></td>
<td>(f) Это должен быть объективный вопрос, на который можно достаточно ответить только с помощью здравого смысла.</td>
</tr>
<tr>
<td></td>
<td>(g) Вопрос должен быть простым и коротким, состоящим только из одного предложения.</td>
</tr>
<tr>
<td></td>
<td>Вопрос:</td>
</tr>
<tr>
<td>Refine question sentences</td>
<td>Если исходное предложение семантически и грамматически правильно, повторите его;</td>
</tr>
<tr>
<td></td>
<td>если оно звучит ненатурально, пожалуйста, перепишите его на корректный и свободно звучащий язык.</td>
</tr>
<tr>
<td></td>
<td>{question}</td>
</tr>
<tr>
<td>Add additional distractors</td>
<td>Пожалуйста, добавьте только два правдоподобных и естественных выбора и сохраните их в <code>{additional_choice</code>:[[]].</td>
</tr>
<tr>
<td></td>
<td>[&quot;{choice1}&quot;, &quot;{choice2}&quot;, &quot;{choice3}&quot; ]</td>
</tr>
<tr>
<td>Verify Qualities</td>
<td>Пожалуйста, выберите только одну букву алфавита в качестве ответа из Answer Choices и сохраните её в формате: <code>{answer</code>: selected_answer}.</td>
</tr>
<tr>
<td></td>
<td>Q: {question}</td>
</tr>
<tr>
<td></td>
<td>Answer Choices: (A) {choice_a} (B) {choice_b} (C) {choice_c} (D) {choice_d} (E) {choice_e}</td>
</tr>
</tbody>
</table>

Table 19: The prompt templates used to create the mCSQA in the Russian version.