CL-HumanEval: A Benchmark for Evaluating Cross-Lingual Transfer through Code Generation

Miyu Sato
Japan Women's University
Tokyo, Japan
m1916038sm@ug.jwu.ac.jp

Yui Obara
Japan Women's University
Tokyo, Japan
m2016026oy@ug.jwu.ac.jp

Nao Souma
Japan Women's University
Tokyo, Japan
m1916045sn@ug.jwu.ac.jp

Kimio Kuramitsu
Japan Women's University
Tokyo, Japan
kuramitsuk@fc.jwu.ac.jp

Abstract

Cross-lingual transfer in large language models (LLMs) has the potential to enhance LLM performance in non-English languages, particularly in specialized fields where it is challenging to gather sufficient non-English data. However, the mechanisms and extent of cross-lingual transfer are not yet fully understood. In this study, we develop a new benchmark dataset called Cross-Lingual HumanEval (CL-HumanEval) to more effectively evaluate cross-lingual transfer. CL-HumanEval is based on the code generation benchmark HumanEval, with careful removal of hints such as function names, variable names, and execution examples to focus on the influence of natural language descriptions. This paper provides an overview of CL-HumanEval and presents experimental results that evaluate cross-lingual transfer at various stages of LLM development, including pre-training, continual pre-training, and instruction tuning. Our findings indicate that CL-HumanEval enables the evaluation of cross-lingual transfer with a focus on natural language differences more than HumanEval.

1 Introduction

In today's global society, many new concepts and ideas are primarily discussed in English. In fields such as advanced science, medical science, and software engineering, where most cutting-edge knowledge is initially provided in English (Guo, 2018). Non-English speakers often require translations to understand these documents, but translation errors can reduce work efficiency.

The emergence of large language models (LLMs) has the potential to significantly improve this situation. As shown in ChatGPT, the LLMs can effectively deal with prompts in non-English languages, even when including cutting-edge knowledge that is considered available only in English.

The phenomenon behind this behavior in LLMs is called *cross-lingual transfer*, where knowledge learned in English is transferred to other languages. However, the cross-lingual transfer isn't always intentional and doesn't always occur (Workshop et al., 2022; Foroutan et al., 2023), depending on training methods and the language makeup of the training data. The mechanisms and extent of cross-lingual transfer are still unclear. To better understand it, we need a benchmark dataset that makes it easy to compare and track the language composition and learning methods.

The goal of this study is to develop a benchmark dataset specifically designed to evaluate cross-lingual transfer in the context of code generation. Our focus on code generation comes from the fact that software engineering is one of the major applications for LLMs, but there has been a notable shortage of non-English data in this domain (Kocetkov et al., 2022). Furthermore, the clear syntactical differences between code and natural language facilitate dataset analysis, making code a suitable choice for benchmarking.

We propose the Cross-Lingual HumanEval (CL-HumanEval), a benchmark dataset to evaluate cross-lingual transfer. CL-HumanEval is based on the code generation benchmark HumanEval, carefully removing hints such as function names, variable names, and execution examples to focus on the influence of natural language descriptions. We switched from the original hand-written descriptions to LLM-generated text to ensure multilingual fairness.

This paper provides an overview of CL-HumanEval and presents experimental results that evaluate cross-lingual transfer at various stages of LLM development, including pre-training, continual pre-training, and instruction tuning. The findings reveal that CL-HumanEval enables a more focused evaluation of how language differences impact code generation capabilities compared to the original HumanEval and JHumanEval benchmarks.

The contributions of this paper are as follows:

- We developed a new benchmark dataset, CL-HumanEval, specifically designed to evaluate cross-lingual transfer in LLMs.

- We evaluated cross-lingual transfer at various stages of development. CL-HumanEval focuses more on natural language differences and captures model differences.

2 Cross-Lingual Transfer and Evaluation

This section clearly defines "cross-lingual transfer" as used in this paper. We consider several multilingual benchmarks and analyze key concerns for evaluating cross-lingual transfer.

2.1 Cross-Lingual Transfer

Here, we consider domain knowledge X that can generate an answer in the LLM. As shown in the next subsection, common sense, mathematics, and programming are examples of such domain knowledge X.

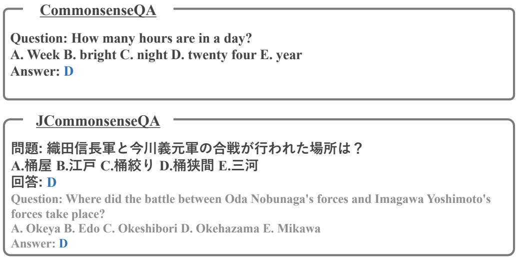
For a given domain knowledge X, we assume the following two points about the English LLM:

- (Assumption 1) The English LLM has been trained on domain knowledge X described in English.

- (Assumption 2) The English LLM has not been trained on domain knowledge X described in Japanese.

We focus on LLMs with a language ratio known in their training data because the training data in current LLMs are often not disclosed. To investigate the occurrence of cross-lingual transfer, we utilize a multilingual benchmark specifically related to domain knowledge X.

Intuitively, if the Japanese benchmark performance in an English LLM is *higher than expected*, it suggests that knowledge transfer from English to Japanese has occurred. However, estimating "higher than expected" is problematic. Because the English LLMs often use large-scale web corpora, they cannot completely exclude multilingual training data.



CommonsenseQA

Question: How many hours are in a day?
A. Week B. bright C. night D. twenty four E. year
Answer: D

JCommonsenseQA

問題: 織田信長軍と今川義元軍の合戦が行われた場所は？
A.桶屋 B.江戸 C.桶絞り D.桶狭間 E.三河
回答: D
Question: Where did the battle between Oda Nobunaga's forces and Imagawa Yoshimoto's forces take place?
A. Okeya B. Edo C. Okeshibori D. Okehazama E. Mikawa
Answer: D

Figure 1: **CommonSenseQA and JCommonSenseQA:** The figure illustrates CommonSenseQA and JCommonSenseQA examples.

We cannot simply estimate the Japanese benchmark scores as zero, nor can we individually measure the impact of small amounts of Japanese training data. This is one of the reasons why tracking cross-lingual transfer has been challenging.

To estimate the extent of cross-lingual transfer, we need to compare the multilingual benchmark performance before and after additional training. Here, additional training refers to any form of training (such as continual pre-training and instruction tuning) applied to the pre-trained English LLM. Note that, at the time of writing, there is no established consensus on what training phase effectively triggers cross-lingual transfer.

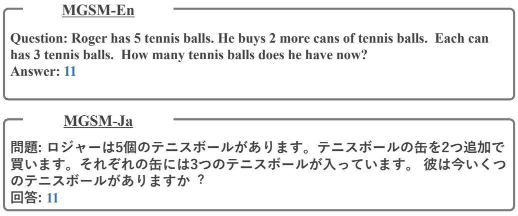
In the additional training, Assumption 2 must still hold. If it does not, then it is difficult to distinguish whether the inference results were obtained from the Japanese additional training or transferred from English. On the other hand, the additional training requires some Japanese training data. Although separating domains (such as X or not) within the same language is not trivial, domains that are easier to separate will be one of the keys to evaluating cross-lingual transfer.

In the remaining sections, we highlight existing major multilingual benchmarks and discuss their suitability for evaluating cross-lingual transfer.

2.2 CommonSenseQA, JCommonSenseQA

CommonSenseQA (Talmor et al., 2018) is a benchmark dataset designed for evaluating the commonsense reasoning abilities of LLMs. An LLM is required to select the correct answer when given a question and multiple-choice options such as Figure 1.

JCommonSenseQA (Kurihara et al., 2022) is the Japanese version of CommonSenseQA. It was constructed separately through crowdsourcing, so the content of the questions is different. For example, JCommonSenseQA includes questions requir-



MGSM-En

Question: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
Answer: 11

MGSM-Ja

問題: ロジャーは5個のテニスボールがあります。テニスボールの缶を2つ追加で買います。それぞれの缶には3つのテニスボールが入っています。 彼は今いくつのテニスボールがありますか ？
回答: 11

Figure 2: **MGSM:** The figure illustrates MGSM examples in English and Japanese.

ing specific knowledge of Japan such as history or place names, as shown in Figure 1.

The knowledge required for JCommonSenseQA is not the same as that needed for CommonSenseQA. For evaluating cross-lingual transfer, it is preferable to use datasets with the same content in different languages, such as those created through translation. Therefore, these datasets with such differences in content may be unsuitable.

2.3 MGSM

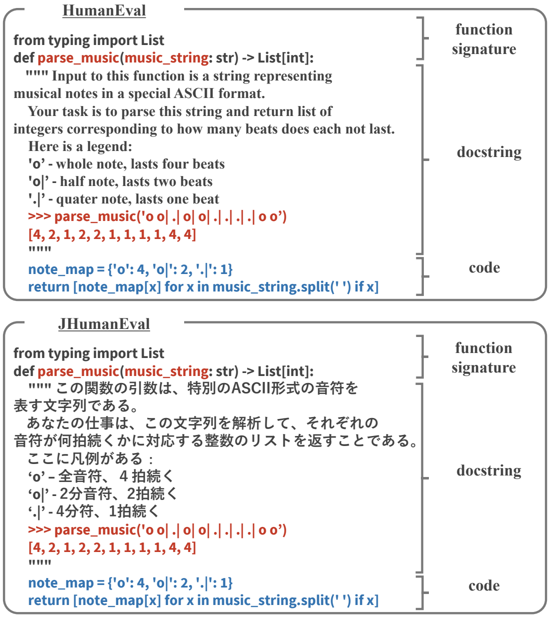
Multilingual Grade School Math (MGSM) (Shi et al., 2022) is a benchmark dataset for evaluating the arithmetic reasoning abilities of LLMs. The LLM is required to generate a numerical answer through multi-step reasoning when given a math problem, such as the one shown in Figure 2.

MGSM presents the same problems across different languages, unlike CommonSenseQA and JCommonSenseQA. It is based on the English dataset GSM8K (Cobbe et al., 2021), which includes grade school level math problems, and has been manually translated into multiple languages, including Japanese. Therefore, the required knowledge is the same across different languages.

Several concerns arise when using MGSM to evaluate cross-lingual transfer. Although mathematics is a distinct domain of knowledge, MGSM makes it difficult to classify as specialized domain knowledge because it consists of elementary-level problems. Additionally, the LLM may require few-shot learning or instruction tuning to generate only numerical answers. Therefore, MGSM is challenging to use directly for evaluations after pre-training. The adjustments needed for evaluation may also unintentionally affect the model's performance.

2.4 HumanEval, JHumanEval

HumanEval (Chen et al., 2021) is a benchmark dataset for evaluating the code generation capabilities of LLMs. Figure 3 shows an example. The LLM is required to complete the function by generating code when given a function signature and a docstring.



HumanEval

```
from typing import List
def parse_music(music_string: str) -> List[int]:
  """ Input to this function is a string representing
musical notes in a special ASCII format.
   Your task is to parse this string and return list of
integers corresponding to how many beats does each not last.
   Here is a legend:
   'o’ - whole note, lasts four beats
   'o|’ - half note, lasts two beats
   '.|’ - quater note, lasts one beat
   >>> parse_music('o o| .| o| o| .| .| .| .| o o’)
   [4, 2, 1, 2, 2, 1, 1, 1, 1, 4, 4]
   """
   note_map = {'o': 4, 'o|': 2, '.|': 1}
   return [note_map[x] for x in music_string.split(' ') if x]
```

JHumanEval

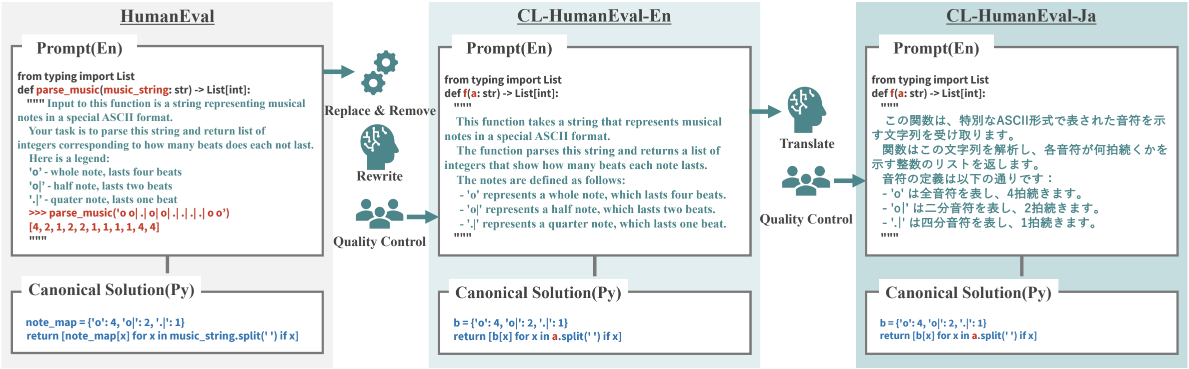
```
from typing import List
def parse_music(music_string: str) -> List[int]:
   """ この関数の引数は、特別のASCII形式の音符を
表す文字列である。
  あなたの仕事は、この文字列を解析して、それぞれの
音符が何拍続くかに対応する整数のリストを返すことである。
  ここに凡例がある：
   ‘o’ – 全音符、４拍続く
   ‘o|’ - 2分音符、2拍続く
   ‘.|’ - 4分符、1拍続く
   >>> parse_music('o o| .| o| o| .| .| .| .| o o’)
   [4, 2, 1, 2, 2, 1, 1, 1, 1, 4, 4]
   """
   note_map = {'o': 4, 'o|': 2, '.|': 1}
   return [note_map[x] for x in music_string.split(' ') if x]
```

Figure 3: **HumanEval and JHumanEval:** The figure illustrates HumanEval and JHumanEval examples.

JHumanEval (Sato et al., 2024) is the Japanese version of HumanEval, with the same function signatures and docstring contents. It was constructed by using both machine translation and manual quality control to translate the English-written docstrings from HumanEval into Japanese.

These datasets offer three beneficial characteristics for evaluating cross-lingual transfer. First, the required programming knowledge is the same in both English and Japanese, making it easy to verify whether code generation capabilities in English can transfer to Japanese. Second, programming is highly specialized domain knowledge, and code is easier to separate from natural language because it is written with strict syntax and structure. Third, HumanEval and JHumanEval can be applied to evaluations before and after pre-training or fine-tuning without the need for adjustments because they are in a code completion format.

Despite these characteristics, there are concerns about directly using these datasets. Function signatures and docstrings contain hints for code generation beyond just the natural language descriptions. The hints include function names and variable names of English origin and execution examples, as highlighted in red in Figure 3. If the LLM generates code based on these hints, it becomes difficult to compare code generation capabilities



```
from typing import List
def parse_music(music_string: str) -> List[int]:
   """ Input to this function is a string representing musical
notes in a special ASCII format.
   Your task is to parse this string and return list of
integers corresponding to how many beats does each not last.
   Here is a legend:
   'o’ - whole note, lasts four beats
   'o|’ - half note, lasts two beats
   '.|’ - quater note, lasts one beat
   >>> parse_music('o o| .| o| o| .| .| .| .| o o’)
   [4, 2, 1, 2, 2, 1, 1, 1, 1, 4, 4]
   """
```

```
note_map = {'o': 4, 'o|': 2, '.|': 1}
return [note_map[x] for x in music_string.split(' ') if x]
```

```
from typing import List
def f(a: str) -> List[int]:
   """
   This function takes a string that represents musical
notes in a special ASCII format.
   The function parses this string and returns a list of
integers that show how many beats each note lasts.
   The notes are defined as follows:
    - 'o' represents a whole note, which lasts four beats.
    - 'o|' represents a half note, which lasts two beats.
    - '.|' represents a quarter note, which lasts one beat.
   """
```

```
b = {'o': 4, 'o|': 2, '.|': 1}
return [b[x] for x in a.split(' ') if x]
```

```
from typing import List
def f(a: str) -> List[int]:
   """
    この関数は、特別なASCII形式で表された音符を示
す文字列を受け取ります。
   関数はこの文字列を解析し、各音符が何拍続くかを
示す整数のリストを返します。
   音符の定義は以下の通りです：
   - 'o' は全音符を表し、4拍続きます。
   - 'o|' は二分音符を表し、2拍続きます。
   - '.|' は四分音符を表し、1拍続きます。
   """
```

```
b = {'o': 4, 'o|': 2, '.|': 1}
return [b[x] for x in a.split(' ') if x]
```

Figure 4: **CL-HumanEval**: This figure illustrates examples of CL-HumanEval and its construction process.

purely based on differences in natural language. Therefore, we develop a new benchmark dataset focused on evaluating cross-lingual transfer.

3 CL-HumanEval

This section presents our benchmark dataset CL-HumanEval.

3.1 Design Principle

CL-HumanEval is a multilingual dataset based on HumanEval and JHumanEval. While inheriting the beneficial characteristics described in Section 2.4, we have made improvements based on the following principles.

- **Purification of Natural Language:** Code generation is a complicated task, and hints such as execution examples are often provided to achieve accurate code. However, these hints are unnecessary when the goal of the benchmark is to accurately measure the impact of natural language alone. Execution examples should be removed and function names or any variable names derived from English or other languages should be replaced.

- **Multilingual Fairness:** Multilingual datasets are often created by using LLM-based machine translation from manually written English text. However, our initial investigation has shown that LLM-generated descriptions usually produce better results than manually written ones. To maintain fairness, we’ve decided to switch the English version from manually written to LLM generated. This will also make it easier to maintain consistency when adding support for more languages in CL-HumanEval.

3.2 Method of Construction

We created an English dataset as the source for the multilingual version by following these three steps:

First, we refined the English version of HumanEval. All function names were replaced with ‘f’ and variable names were shortened to single letters. For example, in the figure, ‘parse_music()’ becomes ‘f()’, and its argument ‘music_string’ is shortened to ‘a’. This transformation makes the identifiers neutral across all languages. If necessary, identifiers in the docstrings were replaced in the same way, and any execution examples were also removed.

Next, we used an LLM to regenerate the English docstrings. The prompt used was: "*Rewrite the docstring in plain English.*" As mentioned later, the multilingual versions are simply translations of this English text.

Finally, we applied human quality control by having multiple reviewers examine the content. Any obvious errors, missed instructions, or unnecessary explanations were corrected and revised.

The multilingual dataset was created from the English version. Identifiers were not changed. For the Japanese version, the docstrings were translated using the following prompt: "*Translate the docstring in plain Japanese.*" Like the English version, human quality control was applied.

The CL-HumanEval dataset follows the same structure as HumanEval, including prompt, canonical_solution, and so on. This enables evaluation using the same script as HumanEval.

The dataset created for this paper (version 1) was generated using GPT-4o mini (version: 2024-07-18) and is available on HuggingFace[1]. The reader may create unsupported multilingual datasets under

[1] https://huggingface.co/datasets/kogi-jwu/cl-humaneval_v1.0

Table 1: **Performance of English LLMs on Benchmark Datasets:** This table shows the performance of various English LLMs on the CommonSenseQA, JCommonSenseQA, MGSM, HumanEval, JHumanEval, and CL-HumanEval datasets. Scores are presented in both English (En) and Japanese (Ja), along with the cross-lingual differences (En-Ja) for each benchmark, as well as the average scores across all models.

Model	Size	CommonSenseQA JCommonSenseQA (0-shot, ExactMatch)			MGSM (4-shot, ExactMatch)			HumanEval JHumanEval (0-shot, pass@1)			CL-HumanEval (0-shot, pass@1)		
		En	Ja	En-Ja	En	Ja	En-Ja	En	Ja	En-Ja	En	Ja	En-Ja
Gemma	2B	41.3	42.3	-1.0	7.6	4.0	3.6	22.0	22.6	-0.6	17.1	14.0	4.3
CodeGemma	2B	29.6	28.0	1.6	4.8	2.0	2.8	34.2	22.6	11.6	20.7	21.3	-0.6
Llama2	7B	41.0	35.9	5.1	6.0	2.8	3.2	12.8	11.6	1.2	11.6	12.8	1.2
CodeLlama	7B	35.5	33.0	2.5	4.4	4.4	0.0	26.8	21.3	5.5	25.0	22.0	5.5
Llama3	8B	44.4	42.8	1.6	14.0	8.4	5.6	37.2	33.5	3.7	34.8	31.7	2.5
Average		38.4	36.4	2.0	7.4	4.3	3.1	26.6	22.3	4.3	21.8	20.4	1.5

similar conditions by using the same LLM. If the LLM updates, the version of CL-HumanEval will be updated to ensure consistency.

3.3 Evaluation Metrics

In CL-HumanEval, the evaluation metric used is the same as in HumanEval, which is $pass@k$ (Chen et al., 2021). The $pass@k$ is defined as the probability that at least one out of the top k code samples passes the unit test for a given problem. In HumanEval, with n: total number of samples, c: number of correct samples, and k: k in $pass@k$, the calculation of $pass@k$ is given by the following:

$$\text{pass@k} := \mathbb{E}_{\text{Problems}}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

In evaluating cross-language transfer, where the goal is to compare the relative code generation capability between languages, it is sufficient to focus on the very first generated sample, setting $n = 1$ and $k = 1$.

4 Experiments on CL-HumanEval

We evaluate the models at various stages, including pre-training, continual pre-training, and instruction tuning.

4.1 English LLMs

To begin, we examined the Japanese language capabilities of several English LLMs. The LLMs examined and their respective training datasets are as follows:

- **Gemma** (Team et al., 2024): Trained on 2 trillion tokens of primarily English data from web documents, mathematics, and code.
- **CodeGemma** (Team, 2024): Trained on an additional 500 billion tokens of primarily English language data from web documents, mathematics, and code, based on the Gemma model.
- **Llama2** (Touvron et al., 2023): Trained on 2 trillion tokens from publicly available sources, with a ratio of 897:1 for English to Japanese.
- **CodeLlama** (Roziere et al., 2023): Trained on 500 billion tokens, primarily code based on the Llama2 model.
- **Llama3** (Dubey et al., 2024): Trained on about 15 trillion tokens, consisting of 50% general knowledge tokens, 25% mathematical and reasoning tokens, 17% code tokens, and 8% multilingual tokens, sourced from curated and filtered web data.

These LLMs are either primarily pre-trained in English or have undergone continual pre-training with source code. Note that these LLMs may include some Japanese content from web corpora. According to the CommonCrawler project, the ratio of English contents to Japanese contents on the Web is approximately 9:1[2]. The Stack project reports that the ratio of English code to Japanese code on GitHub is approximately 94:1 (Kocetkov et al., 2022).

We have compared the performance differences between the English and Japanese versions of CommonSenseQA, MGSM, HumanEval, and CL-HumanEval, as discussed in Section 2. Table 1 summarizes these benchmark scores.

[2] https://commoncrawl.github.io/cc-crawl-statistics/plots/languages

Table 2: **Performance of LLMs after Japanese Additional Training:** This table presents the results of LLMs evaluated on English (En) and Japanese (Ja) benchmarks after continual pre-training and instruction tuning in Japanese. The "Ja-En" column shows the difference between the Japanese scores and the English scores of Llama2.

Model	Continual Pre-training	Instruction Tuning	CommonSenseQA JCommonSenseQA (0-shot, ExactMatch)			MGSM (4-shot, ExactMatch)			HumanEval JHumanEval (0-shot, pass@1)			CL-HumanEval (0-shot, pass@1)		
			En	Ja	Ja-En	En	Ja	Ja-En	En	Ja	Ja-En	En	Ja	Ja-En
Llama2	basemodel		41.0	35.9	-	6.0	2.8	-	12.8	11.6	-	10.4	9.2	-
Swallow	✓(100B)		38.3	56.7	15.7	6.0	5.6	-0.4	3.7	1.8	-11.0	4.3	4.3	-6.1
Swallow-instruct	✓(100B)	✓	36.3	36.7	-4.3	5.6	5.6	-0.4	6.1	1.2	-11.6	1.8	1.2	-9.2
StableLM	✓(100B)		39.2	43.3	2.3	5.2	3.2	-2.8	11.6	13.4	0.6	10.4	9.2	-1.2
StableLM-instruct	✓(100B)	✓	40.1	44.6	3.6	5.6	3.6	-2.4	14.6	11	-1.8	8.5	5.5	-4.9
Youri	✓(40B)		40.1	50.4	9.4	6.0	5.2	-0.8	11.6	10.4	-2.4	7.9	7.3	-3.1
Youri-instruct	✓(40B)	✓	41.5	51.3	10.3	4.0	4.8	-1.2	7.9	5.5	-7.3	4.3	5.5	-4.9

Interestingly, some LLMs show only minor differences in performance between English and Japanese. MGSM captures performance differences; however, because it operates in a lower score range, it may be less effective at distinguishing variations between models. Let us focus on the differences between HumanEval and CL-HumanEval. HumanEval may generate code by leveraging hints such as function names, variable names, and execution examples, whereas CL-HumanEval removes these hints to focus solely on natural language descriptions. As a result, CL-HumanEval scores are lower across all of the LLMs, suggesting that the intended factors were removed. This indicates that CL-HumanEval more accurately measures code generation capabilities from the target language.

In CL-HumanEval, the English version generally outperforms the Japanese version. This is expected given the language makeup of the source code and suggests the possibility of cross-linguistic transfer. However, due to limited details on each LLM's training data, the extent of cross-lingual transfer remains unclear. An ablation study on training data would be beneficial if feasible.

4.2 Japanese Additional Training

Next, we evaluate English LLMs that were subject to additional training, including continual pre-training and instruction tuning, using Japanese datasets. Especially, Japanese continual pre-training is expected to be an effective approach for enhancing the Japanese language understanding and generation capabilities of English LLMs (Fujii et al., 2024). One of the English LLMs Llama2 already exhibits some degree of cross-lingual transfer, as shown by the CL-HumanEval results in Table 2. However, it is interesting to examine how Japanese continual pre-training influences this transfer.

The LLMs examined and their respective Japanese continual pre-training datasets are as follows:

- **Swallow** (Fujii et al., 2024): Trained on 100 billion tokens, with a 1:9 ratio of English sources (The Pile, RefinedWeb) to Japanese sources (Japanese Wikipedia and a curated dataset by Swallow).
- **StableLM** (Lee et al., 2023): Trained on 100 billion tokens, including English sources (English Wikipedia, SlimPajama) and Japanese sources (Japanese Wikipedia, mC4, CC-100, OSCAR).
- **Youri** (Sawada et al., 2024): Trained on 40 billion tokens, from English sources (The Pile) and Japanese sources (CC-100, C4, OSCAR, and a curated dataset by rinna).

The datasets used for Japanese continual pre-training are often proprietary, and details such as the language ratios are frequently not disclosed.

Table 2 show how English and Japanese performance changed the following Japanese continual pre-training. JCommonSenseQA showed a clear improvement in scores; however, it is important to carefully consider whether this is due to newly trained knowledge or cross-lingual transfer. MGSM showed a slight improvement in scores; however, the training dataset includes elementary-level math knowledge.

The cases of HumanEval, JHumanEval, and CL-HumanEval are somewhat different. The continual pre-training dataset contains almost no Japanese code text. Our preliminary investigation confirmed that the Japanese mC4 dataset contains very little source code data. This allows us to focus on cross-lingual transfer; however, Llama2's performance showed little difference between English

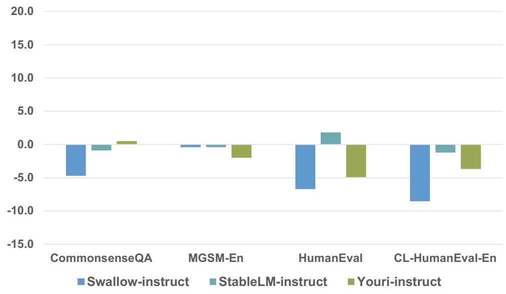


Figure 5: **Impact of Japanese Additional Training on English Tasks:** This chart illustrates how Japanese additional training affects the performance of LLMs on English tasks.

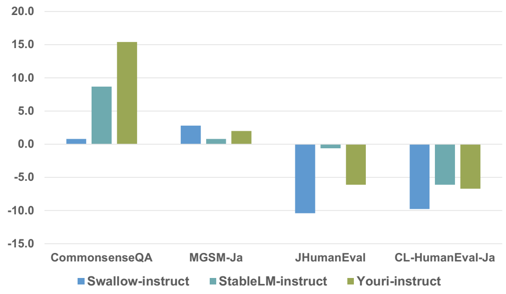


Figure 6: **Impact of Japanese Additional Training on Japanese Tasks:** This chart illustrates how Japanese additional training affects the performance of LLMs on Japanese tasks.

and Japanese. However, catastrophic forgetting was observed instead of promoting cross-lingual transfer.

We also evaluated LLMs after continual pre-training with instruction tuning. Table 2 shows these benchmark scores. The CL-HumanEval results show that scores decreased after instruction tuning compared to before. Figures 5 and 6 illustrate how English and Japanese performance changed the following Japanese additional training including continual pre-training and instruction tuning. CL-HumanEval effectively captures the changes in scores due to Japanese additional training, but all results showed a decline. This confirms that further catastrophic forgetting occurred as a result of the Japanese additional training.

5 Related Work

This study is related to research in cross-lingual transfer and code generation benchmarks.

Cross-Lingual Transfer: Cross-lingual transfer is expected to improve the capabilities of low-resource languages by transferring knowledge learned from high-resource languages. This has been evaluated in tasks such as natural language inference, question answering, and mathematical reasoning (Conneau et al., 2018; Lewis et al., 2019; Shi et al., 2022). The multilingual capabilities of newly released LLMs have been evaluated using independently machine-translated versions of benchmarks like MMLU (Hendrycks et al., 2020; Achiam et al., 2023; Dubey et al., 2024). In this study, we focus on programming knowledge, which requires specialized knowledge and is predominantly available in English. We evaluate cross-lingual transfer through the task of code generation.

Code Generation Benchmarks: Code generation benchmarks exist in multiple datasets to evaluate the capabilities of LLMs (Chen et al., 2021; Austin et al., 2021; Hendrycks et al., 2021). Particularly, HumanEval is widely used as the standard benchmark. Several datasets extending HumanEval have been developed, including those expanded to support multiple natural languages and programming languages (Zheng et al., 2023; Peng et al., 2024). These datasets have highlighted differences in code generation capabilities across languages. We have developed a new benchmark dataset, CL-HumanEval. It is specifically refined to focus on natural language differences for better evaluation of cross-lingual transfer.

6 Conclusion

Cross-lingual transfer in LLMs can enhance performance in non-English languages, especially in fields where non-English data is limited. However, its mechanisms and extent of cross-lingual transfer are not yet fully understood.

We developed CL-HumanEval, a benchmark focused on code generation to more effectively evaluate cross-lingual transfer. CL-HumanEval removes hints such as function names, variable names, and execution examples to isolate the impact of natural language and ensures fairness by using consistent LLM-generated text across languages.

We used CL-HumanEval to evaluate cross-lingual transfer at various stages of LLM development. The results show that CL-HumanEval effectively measures cross-lingual transfer and highlights differences between models. In the future, this could help investigate how differences in the content and language ratios of training datasets impact cross-lingual transfer.

Acknowledgments

This research was supported by joint research with NTT Software Innovation Center and JSPS KAKENHI Grant Number 23K11374.

References

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Negar Foroutan, Mohammadreza Banaei, Karl Aberer, and Antoine Bosselut. 2023. Breaking the language barrier: Improving cross-lingual reasoning with structured self-attention. *arXiv preprint arXiv:2310.15258*.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *arXiv preprint arXiv:2404.17790*.

Philip J Guo. 2018. Non-native english speakers learning computer programming: Barriers, desires, and design opportunities. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–14.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, et al. 2022. The stack: 3 tb of permissively licensed source code. *arXiv preprint arXiv:2211.15533*.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. Jglue: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966.

Meng Lee, Fujiki Nakamura, Makoto Shing, Paul McCann, Takuya Akiba, and Naoki Orii. 2023. Japanese stablelm instruct alpha 7b.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Qiwei Peng, Yekun Chai, and Xuhong Li. 2024. Humaneval-xl: A multilingual code generation benchmark for cross-lingual natural language generalization. *arXiv preprint arXiv:2402.16694*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Miyu Sato, Shiho Takano, Teruno Kajiura, and Kimio Kuramitsu. 2024. Does the llm demonstrate cross-lingual knowledge transfer by additional japanese training? Proceedings of the Thirtieth Annual Meeting of the Association for Natural Language Processing.

Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. Release of pre-trained models for the japanese language. *arXiv preprint arXiv:2404.01657*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

CodeGemma Team. 2024. Codegemma: Open code models based on gemma. *arXiv preprint arXiv:2406.11409*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Zihan Wang, Lei Shen, Andi Wang, Yang Li, et al. 2023. Codegeex: A pre-trained model for code generation with multilingual evaluations on humaneval-x. *arXiv preprint arXiv:2303.17568*.