LLM は日本語追加学習により言語間知識転移を起こすのか？

佐藤 美唯 [1]　髙野 志歩 [1]　梶浦 照乃 [1]　倉光 君郎 [2]
[1] 日本女子大学大学院 理学研究科　[2] 日本女子大学 理学部
m1916038sm@ug.jwu.ac.jp　kuramitsuk@fc.jwu.ac.jp

概要

大規模言語モデル (Large Language Model, LLM) における言語間知識転移は，英語を中心に公開されている情報を日本語でも活用できるようになることから関心が高まっている．現在，英語圏で開発された高性能な LLM に，日本語を追加学習させて，日本語でも英語と同様の能力を目指す動きがある．本研究では，LLM は日本語追加学習により言語間知識転移を起こすのかをコード生成タスクに焦点を当てて分析する．我々は，日本語からコードを生成する能力を評価するために，ベンチマークデータセットである HumanEval を日本語へ拡張した JHumanEval を開発した．そして，日本語追加学習が LLM のコード生成能力にどのような影響を与えるかを検証した．

1　はじめに

現代社会では，先進的な概念やアイディアは英語で表現され，議論されることが多い．特にエンジニアリングの領域では，技術情報を伝える共通言語として英語が定着している．多くの学術論文やチュートリアルが英語で提供されているため，英語が不得意な日本人のエンジニアは機械翻訳を使用するか，日本語での情報提供を待つ必要があった．

しかし，ChatGPT の登場により，状況は大きく変化した．ChatGPT は英語でのみ提供されていると思われる技術情報について，日本語で問い合わせると，日本語で流暢なの回答を生成するようになってきている．これは，大規模言語モデル（Large Language Model，LLM）が英語で習得した知識を日本語に転移し，日本語での応答を可能にしていることを示唆している．このような言語間知識転移（Cross-Lingual Transfer）には，高い関心が集められている．

過去には，多言語のテキストを事前学習した言語モデルが異なる言語間で知識を共有していることを示唆する研究が存在している [1]．しかし，言語間知識転移に関しては，言語による推論能力の不均等さも指摘されており，英語で獲得した知識を他の言語で必ずしも利用できるとは限らない [2]．言語間知識転移の原理への理解を深めることは重要である．

本研究の目的は，英語圏で開発された LLM は日本語追加学習により言語間知識転移を起こすのかを検証することである．英語と比較して日本語テキストデータが不足している背景から，日本語の追加学習は，日本語を強化した LLM の開発において期待される手法である．

我々は，言語間知識転移を分析するために，言語間知識転移の需要が高い技術情報に着目する．具体的には，日本語追加学習により，英語で学習し獲得したコード生成能力が，日本語でのコード生成にどのような影響を与えるかを比較実験する．実験を行うにあたり，コード生成能力の標準ベンチマークである HumanEval の日本語版「JHumanEval[1)]」を開発した．本論文では，JHumanEval の開発方法を紹介し，プログラミングに関する知識転移の発生有無を検証した実験結果を報告する．

本論文の残りの構成は以下の通りである．2 節では，コード生成に着目した理由を述べる．3 節では，JHumanEval の開発について述べる．4 節では，日本語追加学習前後のモデルを使用し，知識転移の発生有無を検証した実験結果を報告する．5 節では，関連研究を概観し，6 節では，本論文をまとめる．

2　問題定義

本節では，我々がコード生成に着目して言語間知識転移を分析する理由を述べる．

2.1　シナリオ

現在，技術情報の多くは英語で提供されている．特にプログラミング分野においては，オープンソース開発が英語圏のコミュニティを中心に行われており，英語での情報アクセスがますます重要になっている．LLM は，ユーザーが必要とする情報を要約した形で提供する能力を持ち，新たな開発手段になり

1) https://huggingface.co/datasets/kogi-jwu/jhumaneval

つつある [3].

日本のエンジニアの間では，プログラミングに関するコードの書き方やエラー分析，バグ修正などの疑問や問題を英語ではなく日本語で LLM に問い合わせる需要がある．英語への翻訳は作業の中断を招くため，直接日本語で問い合わせて，英語でのみ提供されている情報も日本語で回答を得ることが理想的である．

英語圏で開発された LLM は，日本語でのプログラミングに関連する問い合わせに対応可能だが，英語での応答がより正確で質が高いとされることがある．しかし，日本語での情報提供が不十分であると思われる内容にも日本語で答えられることから，言語間知識転移が発生している可能性がある．

本研究では，プログラミング分野の中から，自然言語からのコード生成タスクに焦点を当てる．その理由の一つは，GitHub Copilot[4, 5] など，コード生成は既に実用的なサービスとして提供が始まり，エンジニアが活用しているためである．もう一つの理由は，入力は自然言語で，出力はコードのように，明確に自然言語部分が区別されているからである．コード生成タスクは，オープンエンドな問題で選択問題よりも複雑であるが，ソフトウェアテストベースによる定量評価 [6] が確立されているため，評価がしやすい利点がある．

2.2 言語間知識転移の推定

我々がコード生成タスクを用いて LLM の言語間知識転移をどのように推定するかについて概要を示す．まず，ある LLM が高リソース言語 H と低リソース言語 L から成る訓練データで学習されていると仮定する．

ここで，以下の定義を用いる：

- H_p は，H 言語で指示を与え，コード生成した際の正解率
- L_p は，L 言語で指示を与え，コード生成した際の正解率

高リソース言語 H は低リソース言語 L よりも高い正解率を持つことが期待されるため，$H_p \geq L_p$ となる．そして，H_p と L_p の差，つまり $H_p - L_p$ は，言語間コード生成能力の差分として解釈される．

ここから H から L への知識転移を推定する方法を考える．例えば，LLM が H 言語で H_p 分のコード生成能力を持っているとした際に，この知識が L 言語の訓練データに含まれないとする．この状況で，LLM が L 言語で同様にコード生成に成功した場合，

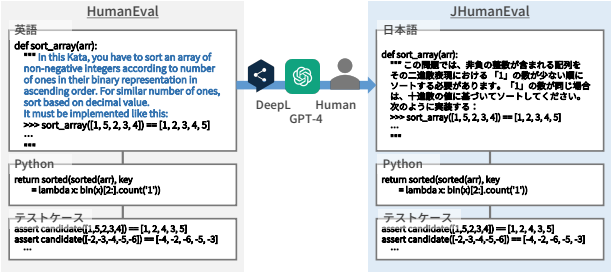

図 1 HumanEval と JHumanEval

H 言語の知識が L 言語に転移したと推定することができる．しかし，LLM の事前学習に使用された訓練データセットの内容は非公開であることが多く，H 言語から L 言語への知識転移によるものか，訓練データに由来する L 言語の知識なのかを区別することは難しい

追加学習やファインチューニングを行った場合，訓練データを自己管理することが可能である．追加学習によって得られた正解率を H'_p, L'_p とすれば，次の式で言語間知識転移率（Cross-Lingual Transfer Rate, CLTR）を計算できる．

$$CLTR = \frac{L'_p - L_p}{H_p - L_p}$$

追加学習やファインチューニングによって破壊的忘却が発生することには注意が必要である．したがって，$H'_p < H_p$ や $L'_p < L_p$ となる場合があり，この場合 $CLTR$ は負の値になる可能性もある．$CLTR$ は-1 に近づくと忘却が発生しており，+1 に近づくと転移が発生していることを示す．

3 JHumanEval

本研究のコード生成能力の評価には，データセット HumanEval と評価指標として pass@k を採用する [7]．また，新たに日本語版 HumanEval の開発を行った．

3.1 HumanEval

HumanEval は，164 個の英語で記述されたプログラミング問題から構成され，それを用いて英語から Python コードの生成能力を評価する．生成されたコードの正しさの評価には評価指標 pass@k が用いられる．pass@k は，k 個のコードサンプルのいずれかがすべてのテストケースをパスした場合に正解のコードが生成されたとみなされる．ポイントは，字句的な類似ではなく，意味的に等しいコードが生成されたかどうかで評価できる点である．つまり，テストケースをパスするコードは，参照コー

ドと異なる形をしていても，正解として扱われる．HumanEval は，LLM のコード生成能力を測定する標準的なベンチマークになっている．

3.2 JHumanEval の開発

我々は，コード生成に着目して，英日間の知識転移の発生有無を検証するために HumanEval を日本語へ拡張した JHumanEval を開発した．JHumanEval は日本語から Python コードの生成能力を測定することを目的としたデータセットである．図 1 に，HumanEval と JHumanEval の一題を示す．

JHumanEval は，HumanEval の英語記述のプログラミング問題を，複数の機械翻訳ツールと著者らによる手作業での翻訳を組み合わせて日本語に翻訳して開発された．まず，DeepL API[2] を使用して英語記述のプログラミング問題を機械翻訳した．機械翻訳では，英日間の文化的な用語・用例の違いや，プログラミング独自の用語により，不自然な意味を持つ文が生成されることがあった．具体例を以下にを示す．原文にある “Kata” を日本語で「型」と直訳すると，意味が通じなくなるなどの誤訳が見られた．

HumanEval 内に含まれる問題文 (抜粋)

In this Kata, you have to sort an array of non-negative integers according to number of ones in their binary representation in ascending order.

機械翻訳による日本語文

この型では，非負の整数の配列を，2 進数表現における “1”の数の昇順でソートしなければならない．その 2 進表現における “1”の数の昇順で並べ替える．

次に，GPT-4 を使用して，機械翻訳による日本語文とコードの関係性を精査しながら用語を修正した．その後，著者らは全問題の訳文を確認し，日本人学生やエンジニアが意味を理解しやすいように，標準的な日本語語彙への置き換え，最終的な日本語文を作成した．

人手翻訳による日本語文

この問題では，非負の整数が含まれる配列をその二進数表現における「1」の数が少ない順にソートする必要があります．

ただし，一部 HumanEval 内に含まれる英語文よりも丁寧・詳細に記述された日本語文になっている可能性があることには注意が必要である．JHumanEval は，LLM が日本語でも高いコード生成能力を発揮するかどうかを評価するための基盤を提供する．

2) https://www.deepl.com/ja/docs-api

表 1 HumanEval と JHumanEval の結果

Model	Size	HumanEval (pass@1)	JHumanEval (pass@1)
GPT-4-0613 [8]	-	56.10	45.12
GPT-3.5-turbo-1106 [3]	-	53.66	48.17
Llama-2-7b-hf [9]	7B	15.24	10.98
CodeLlama-7b-Instruct-hf [10]	7B	30.49	29.27
phi-1 [11]	1.3B	50.00	27.44
llm-jp-1.3b-v1.0 [4]	1.3B	1.83	1.83

3.3 JHumanEval を用いた LLM の評価

我々は，HumanEval と開発した JHumanEval を用いて，LLM のコード生成能力を評価する．表 1 に実験に使用したモデルとそのパラメータ数と，HumanEval と JHumanEval での pass@k のスコアを示す．

結果から，英語圏で開発された LLM は日本語でのコード生成に対応していることが確認された．しかし，5 つのモデルにおいて，英語でのコード生成能力に比べて，日本語でのコード生成能力は低いことが明らかになった．これは，英語圏で開発された LLM は日本語でのプログラミング関連の質問には対応できるものの，英語での応答がより正確で質が高いとされる現状を反映している．

次に，日本語で開発された LLM のコード生成能力に注目した結果，英語と日本語でのコード生成能力に差はないことが確認された．しかし，全体的なコード生成能力は英語圏のモデルよりも低いことが明らかになったため，日本語でも英語と同等以上のコード生成能力を持つモデルの開発が必要になる．

4 実験

本実験では，日本語追加学習前後のモデルを使用して，言語間知識転移の発生有無を検証する．

4.1 実験設定

実験に使用したモデルを以下に示す．

- Llama2 は，Meta 社のオープンソース実装の LLM である．英語が約 9 割を占める 2T トークンのテキストデータで事前学習されている．
- StableLM は，Stability AI Japan 社が Llama2 へ 100B トークンの日本語と一部英語のを追加学習したモデルである．
- ELYZA-Llama2 は，ELYZA 社が Llama2 に対し

3) https://openai.com/blog/new-models-and-developer-products-announced-at-devday

4) https://huggingface.co/llm-jp/llm-jp-1.3b-v1.0

表 2　日本語追加学習前後のモデルにおける HumanEval，JHumanEval および CLTR の実験結果

Model	En	Ja	Code	HumanEval (pass@1)	JHumanEval (pass@1)	CLTR
Llama-2-7b-hf	✓			15.24	10.98	-
japanese-stablelm-instruct-beta-7b [12]	✓	✓		14.02	10.98	0.00
ELYZA-japanese-Llama-2-7b-instruct [13]	✓	✓		12.80	9.76	-0.29
Swallow-7b-instruct-hf [14]	✓	✓		6.71	1.83	-2.14
CodeLlama-7b-Instruct-hf	✓		✓	30.49	29.27	-
ELYZA-japanese-CodeLlama-7b-instruct [15]	✓	✓	✓	31.71	23.17	-5.00

て 180B トークンの日本語テキストの追加学習を加えたモデルである．

- Swallow は，東工大と産総研が共同開発した Llama2 への日本語追加学習モデルである．
- CodeLlama は，Llama2 に対してコードに特化した 500B トークンの追加学習を行ったモデルである．また，16K トークンまでの Long context のファインチューニングを加えて，大きなサイズのコード生成にも対応している．
- ELYZA-Codellama は，ELYZA 社が CodeLlama に対して 180B トークンの日本語テキストの追加学習を加えたモデルである．

これらのモデルに対して，HumanEval と JHumanEval を用いて，コード生成能力を評価し，2.2 節で示した言語間知識転移率 (CLTR) を算出する．

4.2　実験結果

表 2 に実験結果を示す．表 2 は，モデル名，学習データの言語，HumanEval と JHumanEval の pass@k のスコア，および CLTR を左から順に示している．

まず，Llama2 ベースのモデルに日本語追加学習したモデルの結果を確認する．日本語追加学習により，HumanEval のスコアは低下し，JHumanEval のスコアも一致または低下する結果が観察されました．CLTR は，StableLM では JHumanEval のスコアが Llama2 と一致したためスコア 0 となり，その他のモデルでは負の値を示した．

次に，CodeLlama ベースのモデルに日本語追加学習したモデルの結果を確認する．CodeLlama はコードに特化した事前学習がされているため，Llama2 ベースのモデルと比較して HumanEval および JHumanEval の両方で高いスコアを示した．ただし，日本語追加学習により，HumanEval のスコアは若干上昇したものの，JHumanEval のスコアは低下する結果を示し，CLTR も同様に負の値を示している．

以上の結果から，コード生成タスクにおける評価を基にすると，日本語追加学習による言語間知識転移の顕著な促進は確認できなかった．しかし，この評価はコード生成タスクに限定されているため，プログラミング知識の転移に関しては，今後さらなる分析が必要である．

5　関連研究

言語間知識転移の分析はコード生成だけでなく，他の下流タスクで行われている．多言語の評価用データセットを用いて，自然言語推論 [16, 17] や質問応答 [18, 2] や常識推論 [19] タスクなどでも評価されている．また，自然言語の構造的な特徴の影響を分析するために，人工言語 [20] や埋め込み空間の類似度分析 [21, 22] も行われている．本研究では，コード生成タスクに焦点を当てて日本語追加学習による言語間知識転移の分析を行なった．新たに日本語からのコード生成能力の評価が可能な JHumanEval を開発し，HumanEval と JHumanEval を用いた評価を可能にした．これにより，知識編集 [23] などの追加学習以外のアプローチに対しても，実社会での応用を考慮した知見を得ることができる．

6　おわりに

本論文では，LLM は日本語追加学習により言語間知識転移が起こるのかをコード生成タスクに着目して検証実験を行った．検証実験を行うにあたり，新たに日本語からコードを生成する能力を評価するために HumanEval を日本語へ拡張した JHumanEval を開発した．日本語追加学習前後のモデルを用いて実験を実施した結果，コード生成タスクにおいては明確な英語と日本語間のプログラミング知識の転移は確認されなかった．今後は，HumanEval と JHumanEval の結果に加えて，コード生成以外のプログラミングに関する下流タスクの分析を進めるとともに，言語間知識転移についてさらなる理解を深めたい．

謝辞

本研究は JSPS 科研費 JP23K11374 の助成を受けたものです．2023 年度 国立情報学研究所 公募型共同研究「大規模言語モデルの効率良い学習のための訓練データ配信基盤の研究」の一部として実施されたものです．本研究を進めるにあたり，有意義なアドバイスをくださった株式会社 ELYZA の佐々木 彬氏，堀江 伸太朗氏に感謝いたします．

参考文献

[1] Ethan A Chi, John Hewitt, and Christopher D Manning. Finding universal grammatical relations in multilingual bert. *arXiv preprint arXiv:2005.04511*, 2020.

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

[3] Kimio Kuramitsu, Yui Obara, Miyu Sato, and Momoka Obara. Kogi: A seamless integration of chatgpt into jupyter environments for programming education. In *Proceedings of the 2023 ACM SIGPLAN International Symposium on SPLASH-E*, SPLASH-E 2023, p. 50–59, New York, NY, USA, 2023. Association for Computing Machinery.

[4] Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Zhen Ming (Jack) Jiang. Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, Vol. 203, p. 111734, 2023.

[5] Nhan Nguyen and Sarah Nadi. An empirical evaluation of github copilot's code suggestions. In *2022 IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, pp. 1–5, 2022.

[6] Baptiste Roziere, Marie-Anne Lachaux, Lowik Chanussot, and Guillaume Lample. Unsupervised translation of programming languages. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[7] 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. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

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

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

[11] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[12] Meng Lee, Fujiki Nakamura, Makoto Shing, Paul McCann, Takuya Akiba, and Naoki Orii. japanese-stablelm-instruct-beta-7b. https://huggingface.co/stabilityai/japanese-stablelm-instruct-beta-7b. Accessed on January 11, 2024.

[13] Akira Sasaki, Masato Hirakawa, Shintaro Horie, and Tomoaki Nakamura. Elyza-japanese-llama-2-7b. https://huggingface.co/elyza/ELYZA-japanese-Llama-2-7b. Accessed on January 11, 2024.

[14] Naoaki Okazaki, Sakae Mizuki, Hiroki Iida, Mengsay Loem, Shota Hirai, Kakeru Hattori, Masanari Ohi, Rio Yokota, Kazuki Fujii, and Taishi Nakamura. Swallow-7b-instruct-hf. https://huggingface.co/tokyotech-llm/Swallow-7b-instruct-hf. Accessed on January 11, 2024.

[15] Akira Sasaki, Masato Hirakawa, Shintaro Horie, and Tomoaki Nakamura. Elyza-japanese-codellama-7b. https://huggingface.co/elyza/ELYZA-japanese-CodeLlama-7b. Accessed on January 11, 2024.

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

[17] Yujia Qin, Cheng Qian, Xu Han, Yankai Lin, Huadong Wang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Recyclable tuning for continual pre-training. *arXiv preprint arXiv:2305.08702*, 2023.

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

[19] Xianze Wu, Zaixiang Zheng, Hao Zhou, and Yong Yu. Laft: Cross-lingual transfer for text generation by language-agnostic finetuning. In *Proceedings of the 15th International Conference on Natural Language Generation*, pp. 260–266, 2022.

[20] Meryem M'hamdi, Xiang Ren, and Jonathan May. Cross-lingual continual learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3908–3943, 2023.

[21] Yixin Ji, Jikai Wang, Juntao Li, Hai Ye, and Min Zhang. Isotropic representation can improve zero-shot cross-lingual transfer on multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8104–8118, 2023.

[22] Ahtamjan Ahmat, Yating Yang, Bo Ma, Rui Dong, Kaiwen Lu, and Lei Wang. Wad-x: Improving zero-shot cross-lingual transfer via adapter-based word alignment. *ACM Transactions on Asian and Low-Resource Language Information Processing*, Vol. 22, No. 9, pp. 1–23, 2023.

[23] Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, and Jiarong Xu. Cross-lingual knowledge editing in large language models. *arXiv preprint arXiv:2309.08952*, 2023.