

# 継続事前学習による LLM の知識獲得

高橋洸丞<sup>1</sup> 近江崇宏<sup>1</sup> 有馬幸介<sup>1</sup> Benjamin Heinzerling<sup>2,3</sup> Qin Dai<sup>3</sup> 乾健太郎<sup>4,3,2</sup>

<sup>1</sup> ストックマーク株式会社 <sup>2</sup> 理化学研究所 <sup>3</sup> 東北大学 <sup>4</sup> MBZUA dI  
{kosuke.takahashi, takahiro.omi, kosuke.arima}@stockmark.co.jp

benjamin.heinzerling@riken.jp qin.dai.b8@tohoku.ac.jp kentaro.inui@mbzuai.ac.ae

## 概要

継続学習は、モデルが新しい情報を取り入れつつ、既存の知識を保持する能力を向上させる技術であり、本研究では大規模言語モデル (LLM) における継続学習を通じた知識の埋め込みの手法について検討する。本研究では、LLM に特定のドメインデータの知識を覚えさせるために、継続学習時に対象ドメインデータから作成した複数の NLP タスクを混ぜ、さらに指示学習時においても、ドメイン特化した指示データで学習を行う。

QA タスクによる実験の結果、人手評価において、対象のドメインで指示学習データを作ったモデルが最も正答率が高く、ドメイン特化させる場合は指示学習データもドメイン特化させる方がよいと考えられる。また本研究では、継続学習時のコンテキスト長が与える QA タスクの性能変化についても調査した。

## 1 はじめに

近年、大規模言語モデル (LLM) は、企業のデジタル変革を加速させる重要な技術として、業務効率化、意思決定支援、検索サービスへの活用など、さまざまなビジネス分野で活用されている。しかし、これらのモデルがビジネス環境で最大限に機能するためには、単に一般的な知識や言語能力を持つだけでは不十分である。企業ごとに異なる理念や専門用語、業界特有の文化的背景が存在するため、LLM はこれらの企業特有の特徴を反映し、それに適応することが求められる。

ドメイン知識を LLM に反映させる手法は、大きく 4 つに分類される。第一の手法は、対象ドメインに特化したデータを用いて、LLM をフルスクラッチで事前学習する方法である [1, 2, 3]。このアプローチは、対象ドメインの知識やルールを直接モデルに学習させる点で効果的である一方、事前学習に必要

な大規模なデータの収集と高い計算コストが課題となる。第二の手法は、対象ドメインにおける下流タスクの教師データを用いてファインチューニングを行う方法である [4, 5]。しかし、教師ありファインチューニングは、タスク固有のパフォーマンス向上を目的とするため、ドメイン知識そのものの獲得には限界があると報告されている [6]。第三の手法として、RAG (Retrieved Augmented Generation) のように、入力クエリに対して関連する外部知識を検索し、それをプロンプトとして LLM に与えるアプローチが挙げられる [7, 5]。この手法は柔軟性が高いものの、回答が検索精度に依存するため一貫した応答生成が難しい場合がある。最後に、対象ドメインのデータを用いて LLM を継続学習させ、ドメイン知識をモデル内部に直接埋め込む方法がある [8, 9]。この手法は、検索に依存せず、フルスクラッチの事前学習と同様に LLM に知識を獲得させられる点で有効とされている。

本研究では、ビジネス分野におけるデータを使用した実験を行う前段階として、言語処理年次大会 (ANLP) のコーパス<sup>1)</sup>で継続学習による実験を行った。ANLP は、言語処理における最新の研究成果を集約したコーパスであり、NLP 特有の専門用語や分野固有の表現が多く含まれる。本研究の目的は、継続学習を用いて LLM が NLP 特有の知識を獲得する場合、その知識獲得がどれだけ起きているか明らかにすることである。将来的にはビジネス分野における企業固有の知識獲得の応用を目指し、LLM のビジネス環境における個社や個人に特化させることを意識している。

## 2 継続学習

継続学習は、事前学習後の大規模言語モデル (LLM) に追加的な知識を学習させる手法であり、

1) <https://huggingface.co/datasets/kunishou/J-ResearchCorpus>

特に特定のコーパスやタスクに関連する知識を深めるために使用される。本研究では、継続学習を行うモデルとして、事前学習後に instruction tuning された LLM 「llm-jp-3-13b-instruct」 [10]<sup>2)</sup> を使用し、継続学習を行う。ここで instruction-tuned モデルを使用する理由は、多様な指示に対する出力をするためには事前学習時に指示データで学習を行うことが有効だとされているからである [11]。継続学習の目的は、ANLP コーパスに含まれる専門用語や特有の表現をモデルに覚えさせ、ANLP に関するタスクをより適切に処理できるようにすることである。

## 2.1 データ拡張

[9] では、事前学習時に指示データのような合成データを混ぜることで、モデルが下流タスクで優れた性能を示すことが報告されている。本研究でも、このアプローチを踏襲し、データ拡張を通じて ANLP コーパスに多様な表現を加え、データ内部の知識をより確実に LLM が覚えられるようにする。

データ拡張には、Qwen2.5-72B-Instruct[12]<sup>3)</sup> を用いて、以下の観点で全論文テキストに対して合成データを作成した。データ作成時のプロンプトは [9] を参考に日本語で記述し、1 回のデータ作成推論を行う毎に 1 つずつのデータ拡張カテゴリを指定して、1 論文に対して合計 16 個の合成データを準備する。

- paraphrase: 同義の言い換え
- summary: 要約
- generalized: 専門用語を一般的な用語に置き換え
- elaboration: 詳細な説明の追加
- kg\_extraction: 知識トリプレットの抽出
- translation\_en: 英語への翻訳
- translation\_zh: 中国語への翻訳
- qa\_difficult: 専門知識が必要な難しい質問 (QA)
- qa\_easy: 専門知識が不要な簡単な質問 (QA)
- qa\_numbers\_formula: 数字や数式に関連する質問 (QA)
- qa\_methods: 手法に関する質問 (QA)
- qa\_people\_names: 人物名に関連する質問 (QA)
- positive\_negative: ポジティブ・ネガティブな要素のまとめ

2) <https://huggingface.co/llm-jp/llm-jp-3-13b-instruct/tree/main>

3) <https://huggingface.co/Qwen/Qwen2.5-72B-Instruct>

- keywords: 重要なキーワードの抽出とその説明
- reference: 引用論文との関係性の記述
- discussion: 研究的理解を深めるための 5 ターンまでの議論形式の会話

これらの合成データは、元のテキストと共に指示文と出力文のペアを順番に連結して学習させる。

## 2.2 継続学習時のコンテキスト長の変化

上記のように、ANLP の論文データに合成指示データを混ぜた場合、1 つの論文テキストに付随する合成テキストが長くなり、llm-jp-3-13b-instruct の最大コンテキスト長である 4k トークン長では、1 論文に対して全ての合成データをモデルが一度に扱うことができない。そこで、モデルのコンテキスト長を拡張させることで、より元論文と合成データを統一的に学習できるかどうか検証する。コンテキスト長の拡張は Llama3[13] の論文とモデルを参考に、rope の theta を変更することで対応した。

また継続学習は、全体のデータに対して 2epoch 分の学習を行う。コンテキスト長の変動に伴い、学習時の 1 ステップで読む分量が変化するため、4k コンテキスト長を基準としてバッチサイズを 128 で設定し、その他のコンテキスト長の際はバッチサイズの調整を行い、全てのモデルで同量の合計ステップ数となるようにした。

## 3 事後学習

継続学習だけでは、多様な入力に対して柔軟な出力ができないので、事後学習として再度指示学習を行う。指示学習のデータとして、以下の 4 種類を用意した。

- ichikara : ichikara データ [14] のシングルターンだけのデータ
- one-shot : ichikara をランダムサンプリングして与え、ANLP テキストから Qwen で合成した指示データ
- zero-shot : 指示データの例を提示せず、ANLP テキストから Qwen で合成した指示データ
- all included : 上記 3 つを全て含むデータセット

また、all included 以外は全てサンプル数は合計 10000 になるように調整し、all included は 30000 のサンプル数となった。指示データを作る際のプロンプトは QA のデータ拡張時と同様である。

**表 1** 継続学習により知識埋め込みがどれだけできているかを QA で自動評価した結果。**Sent** は、モデル出力と正解回答として準備した文単位の回答に対する BERTscore の F 値で、**Doc** は、モデル出力と元論文の文章全体に対する BERTscore の F 値。**train** は継続学習時にも出現している QA ペアに対するスコア、**test** は、継続学習次に出現していない QA ペアに対するスコア。

モデル	train			test		
	Sent	Doc	GPT-4o-mini	Sent	Doc	GPT-4o-mini
llm-jp-3-13b-instruct	0.59	0.795	0.273	0.397	0.543	0.308
コンテキスト長 1024						
+ 継続学習, ichikara	<b>0.682</b>	<b>0.778</b>	0.494	0.398	0.536	0.408
+ 継続学習, one-shot	0.665	0.773	0.404	0.627	0.807	0.365
+ 継続学習, zero-shot	0.450	0.522	0.363	0.299	0.554	0.352
+ 継続学習, all-included	0.454	0.449	<b>0.517</b>	<b>0.689</b>	<b>0.816</b>	<b>0.420</b>
コンテキスト長 2048						
+ 継続学習, ichikara	0.455	<b>0.544</b>	0.344	0.376	0.540	<b>0.377</b>
+ 継続学習, one-shot	0.468	0.433	<b>0.465</b>	0.373	<b>0.555</b>	0.359
+ 継続学習, zero-shot	0.446	0.438	0.346	0.340	0.492	0.341
+ 継続学習, all-included	<b>0.507</b>	0.427	0.412	<b>0.378</b>	0.531	<b>0.377</b>
コンテキスト長 4096						
+ 継続学習, ichikara	0.467	<b>0.552</b>	0.403	0.371	0.513	0.387
+ 継続学習, one-shot	<b>0.490</b>	0.376	0.387	<b>0.700</b>	<b>0.861</b>	0.364
+ 継続学習, zero-shot	0.437	0.500	0.353	0.556	0.846	0.348
+ 継続学習, all-included	0.460	0.547	<b>0.511</b>	0.546	0.821	<b>0.400</b>
コンテキスト長 8192						
+ 継続学習, ichikara	0.436	0.492	0.357	0.350	0.548	0.340
+ 継続学習, one-shot	0.419	0.374	<b>0.408</b>	<b>0.661</b>	<b>0.852</b>	0.314
+ 継続学習, zero-shot	0.424	0.498	0.319	0.363	0.549	0.324
+ 継続学習, all-included	<b>0.439</b>	<b>0.502</b>	0.378	0.616	0.833	<b>0.354</b>
コンテキスト長 16384						
+ 継続学習, ichikara	0.477	0.527	0.481	<b>0.377</b>	<b>0.556</b>	<b>0.394</b>
+ 継続学習, one-shot	0.454	0.388	0.372	0.372	0.535	0.361
+ 継続学習, zero-shot	<b>0.478</b>	<b>0.541</b>	0.456	0.346	0.538	0.367
+ 継続学習, all-included	0.472	0.448	<b>0.477</b>	0.348	0.513	0.387

表 2 継続学習により知識埋め込みがどれだけできているかをテストデータの手法やツール名、実験結果などに関する質問からランダムサンプリングした 100 件の QA で人手評価した結果

モデル	正答率
llm-jp-3-13b-instruct	0.38
コンテキスト長 1024	
+ 継続学習, ichikara	0.45
+ 継続学習, one-shot	0.48
+ 継続学習, zero-shot	0.65
+ 継続学習, all-included	<b>0.72</b>
コンテキスト長 4096	
+ 継続学習, ichikara	0.46
+ 継続学習, one-shot	0.46
+ 継続学習, zero-shot	0.69
+ 継続学習, all-included	<b>0.73</b>
コンテキスト長 16384	
+ 継続学習, ichikara	0.44
+ 継続学習, one-shot	0.45
+ 継続学習, zero-shot	0.72
+ 継続学習, all-included	<b>0.76</b>

## 4 QA による知識定着の評価

継続学習により LLM の知識定着がどの程度進んだかを評価するため、質問応答 (QA) を各モデルで行う。QA は継続学習向けのデータ拡張で作成したデータ群から、QA に該当するものと keywords の指示データを用いる。100 件のランダムな論文に付随するこれらの QA 拡張データは継続学習時に除外されており、その除外されたデータの QA を test データ、継続学習時にも出現している QA データを train データとした評価セットを組む。

また、QA データの中には「この論文の提案手法は何ですか？」というような特定の論文を指し示していて、その論文を参照しないと答えられない質問が存在している。継続学習時においては、これは元論文が共に与えられるため問題にならないが、評価時には元論文を与えないため元論文の参照が必要な質問は排除すべきである。そこで、OpenAI の GPT-4o-mini を用いて、評価用の QA データの質問のフィルタリングを行った。

**自動評価** 自動評価は、BERTscore[15]<sup>4)</sup> と GPT-4o-mini を用いた。BERTscore は自動的に作成された

4) モデルは <https://huggingface.co/tohoku-nlp/tohokunlp-bert-500m-sq8192-alpha> を使用

回答文を reference とする場合と、元論文を reference にする場合でそれぞれ **Sent**、**Doc** と表記している。GPT-4o-mini での評価は、正解回答文と元論文を提示しつつ、モデル出力が事実に基づくかどうかを 1、0 出力した。表 1 に、QA タスクにおける自動評価の結果を示す。全体的にどのモデルでも、継続学習によりスコアは上昇しており、継続学習後の指示学習データは ichikara, one-shot, all-included のスコアが高い。コンテキスト長に関しては、各モデルを通してあまり差が見られなかった。

**人手評価** 人手評価では、正解文と元論文に加えてコーパス全体の論文を読みながら、モデルの出力が事実に基づくかどうかという評価軸で 1、0 のバイナリー評価をテストデータからランダムにサンプリングした 100 件に対して行った。人手評価の結果を表 2 に示す。人手評価の場合でも、自動評価と同様にコンテキスト長による QA の性能はあまり変化はしていないが、コンテキスト長が伸びるにつれて正答率は微増した。また、指示学習データは ichikara や ichikara を例としてプロンプトに含む one-shot はスコアが低く、zero-shot や all-included が最も正答率が高くなった。付録にて一部事例分析を紹介している (表 3) が、ichikara や one-shot は具体的な数値や団体名などのハルシネーションが多く、それが原因で正答率が下がっている。指示学習データ毎の性能差については、自動評価ではあまり ichikara, one-shot, all-included に差がなかったが、人手評価では明確なスコア差があり、本実験の自動評価の設計はより人手評価に沿うような改善が必要である。

## 5 おわりに

本研究では、継続学習を用いて特定ドメインの知識を LLM に対して効率的に埋め込む手法を提案した。実験の結果、継続学習後のモデルは指示学習データをドメイン特化させることで最も高い正答率を示し、適切なデータ設計が重要であることが確認された。一方で、コンテキスト長の変化による性能差は限定的であった。

また、プロンプト形式による性能の違いを調査した結果、zero-shot や all-included が最も高い正答率を示し、特定の数値や団体名を含む形式では誤りが多発する課題が明らかになった。今後は、ハルシネーション抑制やデータ選定基準の最適化、そしてより高性能な自動評価に取り組む予定である。



## A 人手評価で用いたテストデータ での事例

表3 人手評価で用いた100件のデータセットの例1

質問	正解回答
JParaCrawl とは何ですか？	JParaCrawl は約 4400 万文を超える日英対訳データで、機械翻訳の研究や開発に使用される大規模なデータコーパスです。
llm-jp-3-13b-instruct	JParaCrawl (Japanese ParaCrawl) は、日本のインターネット上の情報を収集するためのクロールツールです。このツールは、ウェブページやブログ、フォーラム、ニュースサイトなど、さまざまなオンラインリソースからデータを取得するために使用されます。...
コンテキスト長 16384 + 継続学習, ichikara	JParaCrawl は、日本語と英語の対訳文を収集し、公開しているプロジェクトです。国立研究開発法人情報通信研究機構 (NICT) のグローバルコミュニケーション計画の一環として行われています。...
+ 継続学習, one-shot	JParaCrawl は、日英対訳文を含む大規模な対訳コーパスであり、約 20 万文対の対訳文が収集されています。このコーパスは、機械翻訳や自然言語処理の研究に利用され...
+ 継続学習, zero-shot	JParaCrawl は、ウェブから大規模に日英対訳コーパスを継続的に構築するプロジェクトです。このプロジェクトでは、機械翻訳モデルの精度向上を目的とし、2014 年から日英対訳データを継続的に収集・公開しています。
+ 継続学習, all-included	JParaCrawl は、クラウドソーシングを活用して作成された日英対訳コーパスであり、約 2100 万文対の対訳文を含んでいます。このコーパスは、機械翻訳モデルの学習に利用され...

図3の事例では、「JParaCrawl とは何ですか？」という質問に対して答える QA になっている。JParaCrawl は NTT が収集している対訳コーパスで、複数回 ANLP のコーパスに出現している。その中でも JParaCrawl v3.0 では、対訳文数が 2100 万越えと記載されており、正解回答文とは異なるものの、all-included の出力は正解と見做すことができる。ichikara と one-shot のモデルについては、JParaCrawl を公開している機関名と文対数が誤っている。