特集論文

製造業ドメイン特化の言語モデル

Domain－Specific Language Models for Manufacturing Industry

内出隼人* *Hayato Uchide*
平森将裕† *Masahiro Hiramori*
斉藤辰彦* *Tatsuhiko Saito*
田口進也* *Shinya Taguchi*
追立真吾† *Shingo Oidate*

*情報技術総合研究所
†設計技術開発センター

要　旨

大規模言語モデル(Large Language Model：LLM)の進展によって，自然言語処理技術(Natural Language Processing：NLP)の実用化が急速に進んでいる。一方，モデルの大規模化に伴う計算コストやエネルギー消費の増大，データプライバシーの懸念，リアルタイム応答性の制約など，製造現場で運用する上で様々な問題が顕在化している。特に，生産ライン停止時の原因特定や対処手順の即時提示といった，生産性に直結する緊急性の高い場面では，限られた計算環境でのモデルの安定動作と高い応答性が求められる。

そこで三菱電機は，FA分野をはじめとする製造業ドメインに特化した小規模言語モデル(Small Language Model：SLM)を開発した。今回，継続事前学習，指示チューニングに加えて，限定されたデータ環境下でも効果的に学習できるアライメントを新たに検討した。その結果，パラメーター数18億個という，エッジデバイス上で動作可能なコンパクトなモデルでありながら，FA分野の知識の正誤を問うタスクで，正解率77.24％を実現できた。これによって，製造現場などの制約がある環境での生成AI活用範囲の拡大が期待できる。

1. ま え が き

LLMの進展によって，NLPの実用化が急速に進んでいる。特に，OpenAIのChatGPT(注1)をはじめとする，クラウド上で動作する汎用LLMは，幅広い言語処理タスクで高い性能を発揮し，製造業を含む多くの産業領域での活用が期待されている。一方，LLMの活用で，モデルの大規模化に伴う計算コストやエネルギー消費の増大が問題になっている。

また，運用の場面では，クラウド上の汎用LLMを単純に利用するだけでは解決が難しい場合がある。例えば，製造現場では，設備マニュアルやトラブル対応記録などの機密情報を外部に送信することが困難な場合や，現場の作業者や設備と即時性の高いやり取りが求められる場合が多く存在する。具体的には，生産ライン停止時の原因特定や対処手順の即時提示，保守作業中の確認支援，異常発生時の対応判断が挙げられる。これらの場合では，設備マニュアルやトラブル対応記録などの機密情報に基づいた，即時性の高い情報取得・応答が，生産性や安全性に直結する。こうした背景から，クラウド上の汎用LLMに依存せずに運用できる仕組みが求められている。特に，セキュアで応答性の高い処理を現場端末で完結させるためには，エッジデバイス上で動作する高性能なSLMの技術開発が重要である。

SLMの開発・運用で，単にモデルサイズを抑えるだけでは十分な性能を得ることは難しい。特に製造業のように専門性の高いドメインでは，汎用モデルには含まれない技術用語や表現が多い。専門性の高いドメインで，ユーザーからの問合せや現場の状況に応じた適切な応答を生成するためには，適切なドメイン適応が不可欠である。そのため，特定ドメインのデータを活用した継続事前学習(1)や指示チューニング(2)，さらにユーザーにとって自然かつ安全な応答を実現するためのアライメント(3)といったファインチューニング手法を適切に組み合わせる必要がある。

また，製造現場などの制約がある環境で言語モデルを実用的に動作させるためには，推論精度を確保しつつ，応答速度やリソース使用量とのバランスを考慮した最適化も重要になる。特に即時性が求められるユースケースでは，現場での応答遅延を最小限に抑える設計や，限られた計算環境で安定動作を実現する工夫が，モデルの性能と実用性を両立する上で重要である。

本稿では，製造業の中でも特にFA分野に特化し，当社FA製品に関する各種ドキュメントを用いてSLM(**図1**)を構築した取組みを述べる。**2章**では，代表的なファインチューニング手法と，それらを用いたドメイン特化の実践例を述べる。**3章**では，OSSを活用した，エッジデバイス上でSLMを動作させる実践例を述べる。最後に**4章**では，今回の成果と今後の展望についてまとめる。

(注1)　ChatGPTは，OpenAI OpCo, LLCの登録商標である。

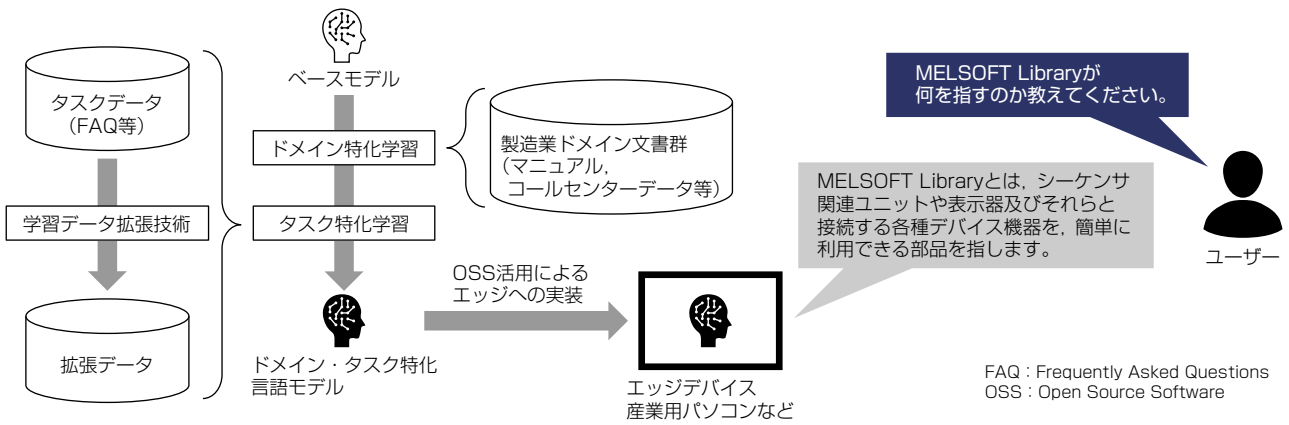

図1−エッジデバイスで動作する製造業ドメインに特化したSLM

2. 製造業ドメイン特化のSLMの開発

LLMは，大量の一般的なテキストデータを用いて事前学習されることによって，幅広い言語処理タスクへの高い汎用性を獲得する。一方，実際の応用では，特定の言語，ドメイン，利用目的に適合するようにモデルを追加で調整する必要がある。こうした目的のための事後学習には，幾つかの代表的な手法があり，それぞれの目的や適用対象に応じて使い分けられる。

この章では，言語モデルの性能を最大限に引き出すための代表的なファインチューニング手法を述べて，今回の開発で取り組んだ製造業ドメイン特化手法を述べる。

2.1　継続事前学習

継続事前学習は，既存の事前学習済みモデルに対して，新たなコーパスを用いて再度事前学習する手法である。これによって，特定言語(例：日本語)やドメイン(例：製造業，医療，法律)での語彙や表現，文脈の知識をモデルに追加的に学習させる。

継続事前学習は自己教師あり学習に基づくため，明示的なタスク定義やアノテーションが不要であるという利点があり，社内文書や設備マニュアル，トラブル対応記録のような非構造データを活用しやすい。

特に日本語や専門性の高いドメインでは，汎用LLMが十分にカバーしていない言語現象や語彙，表記揺れが多数存在するため，継続事前学習は有効な手段になる。一方，既存モデルの性能を劣化させないよう，事前学習データとのバランス設計やトークナイザーの調整が重要になる。

2.2　指示チューニング

指示チューニングは，モデルに“タスクを理解し，指示に従って応答する能力”を学習させるための手法である。具体的には，“命令(instruction)”とそれに対応する“応答(response)”のペアを用いて教師あり学習を行う。例えば，“次の文章を敬語に書き換えてください”といった指示に対して，意味は変えずに自然な敬語の文章を生成するように学習させる。

この手法は，チャット形式やプロンプト形式のインターフェースで，ユーザーが直感的にモデルを活用するために不可欠であり，LLMの対話的な利用を可能にする上で大きな役割を果たす。また，多様なタスクへの汎用性を高めるために，様々な形式の命令データを収集・構築する工夫が求められる。

指示チューニングは，英語で豊富なデータが存在する一方，日本語では高品質なデータセットが限られており，翻訳データの活用や日本語特有のタスク設計が重要になる。

2.3　アライメント

アライメントは，モデルの応答をユーザーの価値観や社会的規範，安全性の観点に整合させるための手法群を指して，単に“正しい”応答を出力するだけでなく，“ユーザーにとって望ましく適切な”応答を生成することを目指す。また，応答

の品質だけでなく，安全性や倫理的な配慮が重要視される。アライメントの実現には，主に人間の選好に基づいた学習と安全性チューニングという二つのアプローチがある。

人間の選好に基づいた学習では，人間が評価した“より好ましい応答”と“望ましくない応答”のペアを用いることで，モデルがユーザーにとって適切な応答を選択する傾向を学習する。この方法は，モデルの出力を評価し，より高い品質と整合性を持つ応答を強化するために利用される。例えば，DPO（Direct Preference Optimization）[4]やPPO（Proximal Policy Optimization）[5]といった手法が広く用いられており，これらは報酬信号を通じてモデルの出力を調整することで，応答品質を向上させることを可能にする。

また，安全性チューニングは，モデルが攻撃的，不適切，又は危険な内容を含む出力を抑制することを目的とする。特に，日本語を対象とする場合，英語とは異なる文化的背景や社会的規範を考慮した安全性の確保が必要になる。例えば，日本語の言語的特性や社会的慣習を考慮したフィルタリング基準を設けることで，意図しない誤解や不適切な応答を回避することが求められる。このような取組みは，モデルを業務システムや公共のサービスに組み込む際の信頼性向上に直結するものであり，特に製造業のように安全性が重視されるドメインでは欠かせない要素である。

このように，アライメントのプロセスは，モデルの応答をユーザーの期待や要求に適合させるための重要な手段である。適切なアライメントを実現することによって，モデルの信頼性と実用性を高めることができて，より安全で効果的なシステムの構築が可能になる。

2.4　製造業ドメイン特化手法

ここまで述べた手法を用いた，製造業ドメイン特化手法を述べる。製造業ドメイン特化手法のフローを**図２**に示す。今回の取組みでは，当社FA製品に関する各種ドキュメントを使用し，継続事前学習，指示チューニング，アライメントの三つの手法を適用した。これによって，製造業特有の専門知識や用語を効率的に学習させて，モデル性能の向上を図った。

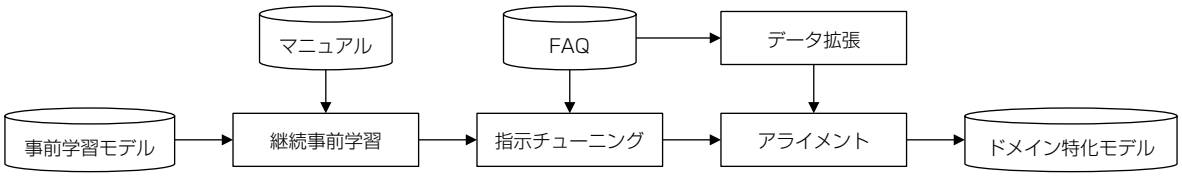

図２−製造業ドメイン特化手法のフロー

ベースとなる事前学習モデルには，18億個のパラメーター数のモデル[6]（モデルアーキテクチャー：Llama2[7]）を採用した。モデルサイズの選定では，製造業ドメインに特化した学習を効率的に行うために，計算コストと性能のバランスを考慮した。特に製造業では，リソース制約を踏まえつつ高精度なモデルを構築することが求められるため，この規模のモデルを選定した。また，当社ではモデルの学習データを適切に扱って，将来的にモデルの出力結果に対して説明できるように，モデルの透明性を高めることを重要視している。この観点から，学習データも併せて公開されているこのベースモデルを採用するに至った。それ以降の処理では，このベースモデルを用いて各種ドメイン特化手法を適用した。

継続事前学習では，FA製品に関するマニュアルデータを使用した。このマニュアルデータは，製品の仕様，操作手順，トラブルシューティングなど幅広い内容を含む。事前学習済みモデルに対して，この専門的なコーパスを再学習させることで，FA製品に特有の語彙や表現，構造化された情報を効率的に取り込むことを目指した。

指示チューニングでは，FA製品に関するFAQデータを利用した。このデータは製品の使用方法やトラブル対応に関する質問とその回答を含む形式で構成されている。指示チューニングによって，モデルはユーザーからの質問に対して適切な応答を生成する能力を向上させることができる。特に，質問形式での入力に対して自然な応答を提供するためには，この段階でのチューニングが不可欠である。さらに，製造業に特有の問合せに対応するため，FAQデータの内容も精査し，適切な形式に整備した。

アライメントでは，指示チューニングと同様に，FA製品に関するFAQデータを活用した。アライメントの目的は，モデルがユーザーにとって望ましい応答を生成する能力を高めることである。しかしながら，FAQデータは質問とそれに対応する“適切な回答”が含まれている一方，“望ましくない応答”が存在しない点に対応する必要があった。このため，既存のFAQデータ内の回答テキスト群から，テキストの類似度が高い，似て非なる回答テキストを抽出し，これを同一質問に対する“望ましくない応答”とみなすデータ拡張手法を新たに検討した。この手法によって，人手による追加アノテー

ションや“望ましくない応答”の設計を行わずに，既存のドメインデータから効率的にアライメント用のデータペアを用意できるようになった。

こうした継続事前学習，指示チューニング，アライメントの手法を組み合わせることで，製造業ドメインに特化したモデルの構築を実行可能にした。モデルの性能評価は，評価モデルにAnthropic Claude-3.7 Sonnetを用いたLLM-as-a-judge[8]による参照回答付きの自動評価を採用した。評価データには，当社FA製品に関する知識の正誤を問う質問群を使用し，比較対象には代表的なクラウド上の汎用LLMであるOpenAI GPT-4oを選定した。評価データに対する，今回開発したモデルの応答と，比較対象であるGPT-4oの応答について，参照回答と比較し正誤を判定させた。この比較によって，開発したモデルの有効性を，一定の客観性をもって評価できる。

評価結果として，今回開発したモデルは，正解率77.24%になることを確認した。比較対象であるGPT-4oは正解率52.03%であった。この評価は当社FA製品に関する知識の正誤を答える２択問題であるため，ランダムに答えた場合の正解率(チャンスレート)は50%が期待される。GPT-4oの正解率はチャンスレートに近い結果であった。一方，今回開発したモデルはチャンスレートを27.24%上回る結果であり，この開発で検討したドメイン特化手法の有効性を示すことができた。

3. OSS活用によるエッジAIへの実装

ドメイン特化SLMは，クラウド上でLLMを動作させる従来のアプローチとは異なり，軽量で，かつ高い応答性が求められる現場のニーズに応えるため，エッジデバイス上での実行(エッジAI)が期待されている。エッジAIとしてSLMを動作させることで，低遅延かつプライバシーを考慮した処理が可能になり，スマートファクトリーやエッジロボティクス，エネルギー制御など多様な分野で，その価値は大きい。

エッジデバイス上でドメイン特化SLMを効率的に動作させるには，限られたハードウエアリソースの中で性能・精度を両立させる最適化が不可欠である。最適化の手段として，柔軟かつ高機能なOSSの活用が重要になる。当社は，AI開発の基盤となるOSSのコミュニティーで，世界トップクラスの技術者とともに開発に参画し，技術力の向上と社会への貢献を同時に果たしている。具体的には，AIコンパイラOSSであるApache TVM[注2)][9]で，当社のエンジニアがソースコードの編集権限を持つ“コミッター”として中核的な立場で活動しており，またAIフレームワークとして事実上の業界標準となっているPyTorch[注3)]でも当社の貢献が評価されて，PyTorch Contributor Awards 2024[10]の最終候補者として選出された。

この章では，これらOSSに関する知見を生かして，当社が取り組んだSLMをエッジデバイス上で動作させる実証について述べる。対象デバイスは，GPUを搭載したJetson[注4)] Orin Nano 8GBと，NPU(Neural Processing Unit)を搭載したRadxa ROCK 5B 16GBの２種であり，どちらも産業用途での応用を視野に入れた選定である。今回は，この両デバイス上で37億個のパラメーター数のSLM(モデルアーキテクチャー：Llama2)をOSS活用によって省メモリーかつ高速に実行する方式を検討した。検討に先立って，PyTorchのEagerモード実行といった一般的な実行方法では対象デバイスのメモリー不足(必要メモリー量は14.9GB)のため前記SLMが実行不可能であることを確認し，この取組みの重要性を明確にした。

エッジ上でSLMを実行するに当たって，まず適切なLLM推論用OSSを選定する必要がある。OSS選定に当たっては，**２章**で述べたSLM(モデルアーキテクチャー：Llama2)に対応していること，対象デバイスに対応していること，の２点を満たすOSSを候補として，OSS活用の有用性を各々評価した。

Jetson Orin Nano 8GBでは，MLC-LLM(Apache TVMベース)，ExecuTorch，ollama，vLLM，IREE，llama.cpp[11]が選定候補になり，最終的にllama.cppを採用した。llama.cppは，推論の実装方針として“handcrafting方式”すなわちハードウエアごとに手動でコードを最適化するスタイルを取っており，開発初期段階で柔軟性が高く，導入が比較的容易である点を重視して採用した。これによって，Jetson Orin Nano 8GB上でSLM実行が可能になり，実用的な推論速度を確認できた。実行時の生成速度とメモリー使用量については**表１**に示す。

Radxa ROCK 5B 16GBでは，MLC-LLM(Apache TVMベース)，rkllmが選定候補になったが，MLC-LLMは対象デバイスへの対応が十分ではなかったため，rkllmを採用した。実行時の生成速度とメモリー使用量については**表２**に示す。

このように，二つの対象デバイスに対して最適なOSSを活用することで，一般的なAI実行方法ではメモリー不足によって実行不可能であった37億個のパラメーター数のSLMが実行可能であることを確認した。

表１−エッジデバイスでのドメイン特化SLM実行結果（Jetson Orin Nano ８GB）

対象デバイス	Jetson Orin Nano ８GB（GPU搭載）
実行対象SLM	3.7BクラスSLM（アーキテクチャー：Llama2）
OSS選定候補	MLC−LLM（Apache TVM），ExecuTorch，ollama，vLLM，IREE，llama.cpp
採用OSS	llama.cpp
生成速度	22トークン/秒
メモリー使用量	2.4GB

表２−エッジデバイスでのドメイン特化SLM実行結果（Radxa ROCK 5B 16GB）

対象デバイス	Radxa ROCK 5B 16GB（NPU搭載）
実行対象SLM	3.7BクラスSLM（アーキテクチャー：Llama2）
OSS選定候補	MLC−LLM（Apache TVMベース），rkllm
採用OSS	rkllm
生成速度	6.8トークン/秒
メモリー使用量	4.5GB

なお，LLM推論フレームワークの設計方針には大きく分けて二つの方式が存在する。一つは，今回採用したllama.cppのような“handcrafting方式”で，ソースコードを直接調整することで高い柔軟性を実現する方式である。もう一つは，Apache TVMに代表される“compiler方式”で，対象デバイスに合わせた最適化コードを自動生成する手法である。

handcrafting方式は，LoRA（Low−Rank Adaptation）のような効率的なファインチューニング手法への対応で柔軟であり，導入も比較的容易である。一方，対応可能なプラットフォームやデバイスは限定的であり，対象デバイスごとの実装・最適化が必要になるため，拡張性や保守性の面で課題が残る。

これに対してcompiler方式は，現時点でLoRA対応に制約があるものの，MLC−LLMのようにモバイル環境やブラウザー環境を含む幅広いプラットフォームに対応しており，将来的な拡張性やスケーラビリティーの観点で優れている。また，コード生成によって環境ごとの最適化を自動化できるため，長期的には保守性や開発生産性の向上が期待できる。

当社は，今後更に多様化するデバイス環境やユースケースへの対応力を強化するため，compiler方式への移行を進める方針である。その上で，LoRAをはじめとする学習最適化機能の実装については，OSSコミュニティーと連携しながら対応機能の拡充を目指す。これまでに培ってきたOSS活用技術を生かして，エッジAI領域でのドメイン特化SLMの実用化と普及を加速し，更に持続可能で拡張性の高い社会の実現に取り組んでいく。

（注２）　Apache TVMは，Apache Software Foundationの登録商標である。
（注３）　PyTorchは，The Linux Foundationの登録商標である。
（注４）　Jetsonは，NVIDIA Corp.の登録商標である。

4. む　す　び

今回の結果は，製造業のように専門的かつ限定的なドメインに特化したSLMの開発で，継続事前学習，指示チューニング，アライメントの組合せが有効であることを示している。また，この手法はほかの製造業関連ドメインや異なる業種に対しても応用可能であり，適用範囲を広げることで更なる性能向上が期待される。

今後の課題として，モデル性能を更に向上させるための追加データの収集や，アライメント手法の改善が挙げられる。特に，望ましくない応答の自動生成手法の精緻化や，応答の多様性と一貫性を両立させるための工夫が求められる。また，モデル評価方法についても，より多角的な指標を用いて実用性を定量的に測定する手法を検討する必要がある。それに加えて，多岐にわたるユースケースに対応するための，エッジデバイス上での最適化を検討する必要がある。ユースケースによっては，現状の想定よりも高い応答性や少ないリソース使用量が求められる可能性がある。そのため，限られた計算環境でのモデルの安定動作と高い応答性を実現するための，実装面の工夫が必要である。これらの課題に継続的に取り組むことで，製造業ドメインでのSLMの更なる高度化と実用化を実現したいと考えている。

参 考 文 献

(1) Gupta, K., et al.：Continual Pre-Training of Large Language Models: How to (re)warm your model?, The Thirty-Seventh Annual Conference on Neural Information Processing Systems (2023)
(2) Wei, J., et al.：Finetuned Language Models are Zero-Shot Learners, The Tenth International Conference on Learning Representations (2022)
(3) Bai, Y., et al.：Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, arXiv, 2204.05862 (2022)
(4) Rafailov, R., et al.：Direct Preference Optimization: Your Language Model is Secretly a Reward Model, Advances in Neural Information Processing Systems, **36**, 53728～53741 (2023)
(5) Schulman, J., et al.：Proximal Policy Optimization Algorithms, arXiv, 1707.0634 7(2017)
(6) Hugging Face：LLM-jp
https://huggingface.co/llm-jp
(7) Touvron, H., et al.：Llama 2: Open Foundation and Fine-Tuned Chat Models, arXiv, 2307.09288 (2023)
(8) Zheng, L., et al.：Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, Advances in Neural Information Processing Systems, **36**, 46595～46623 (2023)
(9) Chen, T., et al.：TVM: An Automated End-to-End Optimizing Compiler for Deep Learning, 13th USENIX Symposium on Operating Systems Design and Implementation, 578～594 (2018)
(10) PyTorch：Announcing the 2024 PyTorch Contributor Awards
https://pytorch.org/ecosystem/contributor-awards-2024
(11) GitHub：ggml-org/llama.cpp
https://github.com/ggml-org/llama.cpp