大規模言語モデルベースの日本語固有表現抽出における Self-Reflection と Few-Shot 学習による精度改善

久保田崇文
株式会社カカクコム
kubota_takafumi@kakaku.com

概要

本研究では，大規模言語モデル（Large Language Model: LLM）を活用した日本語固有表現抽出（Named Entity Recognition: NER）の精度向上手法として Self-Reflection の有効性を検証した．Self-Reflection は, モデルが自身のタスク結果を評価し，反省点を考慮した再試行を行うことで精度を向上させる枠組みである．GPT-4o を対象とした実験では，Few-Shot 学習と組み合わせることで F1-Score が 1.6 ポイント向上した．しかし，一部の条件下では精度の低下や評価結果の不正確性が観察され, 今後は Few-Shot 設計の最適化が課題となる．本研究の結果は, LLM ベースの日本語 NER 精度向上の指針として，実応用への貢献が期待される．

1 はじめに

1.1 本研究の背景と動機

NER とは，文章の中から特定の名前（組織名，人名など）や属性（日時，金額など）を抽出するタスクである．この技術は Web サービスにおける，記事のタグ付け，検索やレコメンド精度の向上，情報抽出業務の自動化などに応用できる．さらに，セキュリティ分野における個人情報検知や金融業界における市場予測などにも使われている．また，生成 AI の高度化技術である Graph RAG [1] の基盤となる知識グラフの構築にも利用されている．これらのことから，NER の精度向上は産業界全般において大きなインパクトとなると考えられる．

1.2 本研究が解決する問題

高精度の NER 手法の一つに DeBERTa [2] などの Transformer のエンコーダーモデルを用いた手法が挙げられる．この手法では，教師ありファインチューニング（Supervised Fine-Tuning: SFT）を行うことによって高い精度を達成している．しかし，高い精度を出すためには数千程度の潤沢な学習データが必要であり，これは実応用上の大きな課題である．例えば，新しいタイプの固有表現を追加する場合は，その都度 SFT のためにデータ収集を行う必要があり, 時間的，人的コスト上実施できない場合が多い．

一方, デコーダー系の LLM は SFT を用いずに NER タスクに対応可能である．これは，入力テキストに対して実行すべきタスクを自然言語で記述するだけで, モデルが直接的に結果を生成できるためである．そのため，固有表現の種類を追加する際にも柔軟に対応できる利点がある．しかし，その精度は大量の学習データを用いて SFT を施したエンコーダーモデルよりも劣ることが知られている．[3]

そこで, 本研究では SFT を用いずに高精度な LLM ベースの NER を行うべく，Self-Reflection を導入することを考えた．Self-Reflection は LLM が自身の行なったタスクを評価し，悪い結果である場合，反省点を踏まえて再実行するといった枠組みである．[4] これは，SFT による LLM 自体のパラメーター更新なしにタスクの精度向上を狙える手法である．

1.3 本研究の貢献

Self-Reflection を導入することで, 既存のデータセットを用いた GPT-4o による NER タスクにおいて，F1-Score が 77.6%から 78.0%へと 0.4 ポイント改善した．さらに，Few-Shot 学習と組み合わせることで 78.6%から 80.2%へと 1.6 ポイント改善した．これらの結果は，Self-Reflection が日本語 NER タスクに対して有効であることを示唆している．

また本研究は, Self-Reflection を用いた日本語 NER 精度向上の初めての体系的検証である．そのため，4.3 項および 4.4 項に記載した課題および今後の展望は，LLM ベースの日本語 NER の精度向上に向けた指針となり，実応用上においても有用であると考えられる．

2 関連研究

2.1 LLM ベース NER の有望性

Keraghel らの NER に関する包括的な調査 [5] によると，大規模で一般的なドメインのデータセットでは，DeBERTa などの Transformer のエンコーダーモデルが非常に高い性能を示したと言及されている．しかし，小規模なデータセットにおいては過学習によって汎化性能が低下することも指摘されている．一方, Wang らの GPT-3 を用いた NER 手法 [3] は，極めてデータ量の少ない低リソース環境で，BERT ベースの教師あり学習モデルを上回る性能を示している．これらの結果から，大規模データセットではエンコーダーモデルが強力である一方，小規模データセットや低リソース環境下では GPT-3 のようなデコーダーモデルを用いたアプローチが有望であることが示唆される．

2.2 Self-Reflection の有望性

Liu らの Agent Design Pattern Catalogue [4] では，Self-Reflection により自身の応答と推論プロセスを評価し，誤りや不適切な出力を特定して修正できるようになり，推論の確実性と応答の正確性が向上すると言及されている. また, Shinn らの Self-Reflection を構成要素に含むフレームワークである Reflexion [6] では，複数の文書から情報を集めて質問に答える推論タスクでベースラインのアプローチよりも大幅なマージンで性能を上回ったと報告されている．これらのことから，同じ推論タスクである日本語 NER タスクも Self-Reflection を導入することで改善する可能性があると考えられる．

3 実験設定

3.1 LLM の選定

本研究の対象とした LLM を以下に示す.

- gpt-4o-2024-08-06（GPT-4o）
- gemini-1.5-pro-002（Gemini）
- claude-3-5-sonnet-v2@20241022（Claude）

これらはクローズドモデルであるため，具体的なアーキテクチャやパラメーター数は公開されていないが、以下の理由により選定した.

- 既に産業界では広く使われており，検証結果の実応用上の貢献度が高いこと.
- Self-Reflection の有効性は，LLM がブラックボックスであっても検証可能であること.
- ユーザー評価のリーダーボード [7] においてオープンソースのモデルよりも良い結果を記録していること.

3.2 データセットの選定

本研究では，評価データセットとしてストックマーク NER データセット [8] を採用した．採用理由は他の候補と比較して，アノテーションの一貫性およびデータ品質の高さ，シンプルなデータ形式による既存枠組みでの推論の容易さ，さらにデータセットの入手性の高さが挙げられる.

採用したデータセットは，日本語版 Wikipedia [9] を用いて作成されており，固有表現のタイプは「人名」，「法人名」，「政治的組織名」，「その他の組織名」，「地名」，「施設名」，「製品名」，「イベント名」の 8 種類となっている．データ数は，アノテーションされた 4859 文と固有表現が含まれていない 484 文の合計 5343 文である．今回は，ランダムに抽出した 8 割をテストデータとし, 残りの 2 割からさらにランダムに抽出した 10 サンプルを Few-Shot 用のデータとした．ただし，Few-Shot 用の 10 サンプルは，全ての固有表現タイプと固有表現のないデータが含まれるように調整した.

3.3 Self-Reflection の実装

本研究では LangGraph [10] を用いて図 1 のような Self-Reflection の Agent フローを実装した．各 Agent の実装は以下の通りである.

- Prediction Agent（Pred）: 入力テキストの NER を行う．2 周目以降は Reflection Agent から出力された「改善すべき点」を考慮する.
- Evaluation Agent（Eval）: Pred のタスクとその実行結果を受け取り，精度を以下の 5 つの観点から各 20 点満点で評価する.
  - 抽出範囲：抽出範囲が妥当か
  - タイプ：タイプが妥当か
  - 文脈依存性：文脈に適したタイプか
  - 抽出漏れ：抽出漏れがないか
  - 過剰抽出：過剰に抽出していないか

  なお、Eval による評価は「Reflection で改善点を引き出すための指標」に過ぎず，実際の性能評価に用いる F1-Score とは別物である.
- Reflection Agent（Refl）: Pred のタスクとその

実行結果および Eval の評価結果を受け取り，「改善すべき点」を出力する．

また，Eval から Refl に遷移する条件は，評価結果が 100 点未満かつ Reflection 回数が 3 回未満の場合とした．なお，各 Agent は全て同一の LLM となるようにしている．さらに，LLM の出力を構造化するために，LangGraph が提供している機能である .with_structured_output() [11] を利用している．

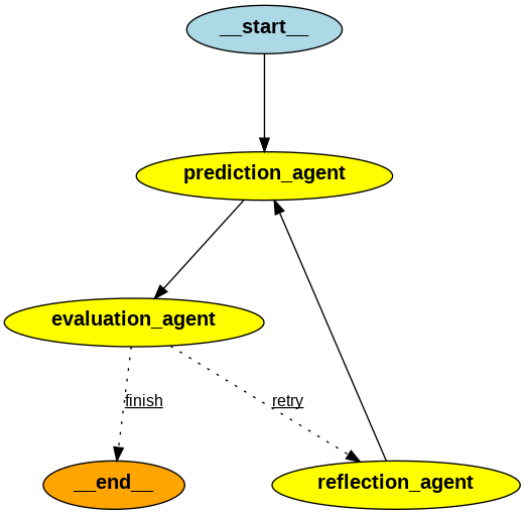

図 1　Self-Reflection の Agent フロー

3.4 Few-Shot Prompt の設計

本研究では Few-Shot Prompt として，Pred にはテキストと答えのセットを与え，Eval にはテキストと答えと満点の評価結果を与えた．その上で，それぞれ Few-Shot 無しの場合と比較した．Refl については，Few-Shot を自動で作成することが困難であったため，今回の調査では Few-Shot 無しで統一した．

4 結果および考察

4.1 定量評価結果

表 1 は，各モデルに対して，Pred および Eval の Few-Shot 有無を組み合わせた 4 通りの条件で得られた NER の F1-Score を示し，さらに Self-Reflection の導入による改善幅（Gain:　Self-Reflection による F1-Score から Baseline の F1-Score を減じた値）を併せて示している．ここで，Baseline は Self-Reflection を行わず，初回出力を評価した結果である．Few-Shot の設定は以下の 4 通りである．

- Pred, Eval の双方が Few-Shot 無し
- Pred のみが Few-Shot 有り
- Eval のみが Few-Shot 有り
- Pred, Eval の双方が Few-Shot 有り

なお，Eval に Few-Shot を与えるか否かは 2 回目以降の予測結果にのみ影響することから，初回予測結果である Baseline は，Pred-Few-Shot の有無が同一である場合，Eval-Few-Shot の有無に関わらず同じ条件となるため，データを流用して実験を行った．

表 1　Self-Reflection による F1-Score の改善幅

LLM	Pred-Few-Shot	Eval-Few-Shot	F1-Score [%] Base-line	F1-Score [%] Self-Reflection	Gain
GPT-4o	×	×	77.6	78.0	0.4
	×	○		78.4	0.8
	○	×	78.6	79.7	1.1
	○	○		80.2	1.6
Gemini	×	×	68.2	70.7	2.5
	×	○		70.7	2.5
	○	×	75.2	70.8	-4.4
	○	○		74.5	-0.7
Claude	×	×	77.8	78.5	0.7
	×	○		78.6	0.8
	○	×	79.6	80.1	0.5
	○	○		80.1	0.5

ほとんどの条件において F1-Score が 0.4〜2.5 ポイント改善したが，Gemini の Pred-Few-Shot 有りの条件では Eval-Few-Shot 無しで -4.4，有りで -0.7 ポイントとなった．Pred または Eval の片方のみ Few-Shot 有りの設定では，双方が無い設定と有る設定の中間的な F1-Score を示しているが，Pred のみの方が高い傾向がある．

4.2 定性評価結果

表 2 には，F1-Score が最も高かった条件（Pred と Eval 双方の Few-Shot を有りに設定した場合の GPT-4o）における NER 出力例を示す．1st Prediction では，Ground Truth に対して過剰抽出であったが，2nd Prediction では軽減されていることが確認できる．一方，1st Evaluation でタイプスコアと過剰抽出スコアが低いことは、name や type の誤り方から理解可能であるものの，抽出範囲スコアと文脈依存性スコアが低い理由は解釈が困難である．

表 2　Pred と Eval 双方の Few-Shot を有りに設定した場合の GPT-4o を用いた NER 出力例

Input Text	そういった中で、オーストラリアで実施された最大の M&A は、ウエストパック銀行がセントジョージ銀行を買収した案件であり、その金額は 191 億オーストラリア・ドルであった。
Ground Truth	[{'name': 'オーストラリア', 'type': '地名'}, {'name': 'ウエストパック銀行', 'type': '法人名'}, {'name': 'セントジョージ銀行', 'type': '法人名'}
1st Prediction	[{'name': 'オーストラリア', 'type': '地名'}, {'name': 'ウエストパック銀行', 'type': '法人名'}, {'name': 'セントジョージ銀行', 'type': '法人名'}, {'name': 'オーストラリア・ドル', 'type': 'その他の組織名'}]
1st Evaluation	抽出範囲: 16, タイプ: 12, 文脈依存性: 18, 抽出漏れ: 20, 過剰抽出: 14
1st Reflection	「オーストラリア・ドル」は固有表現タイプのリストに含まれていないため、抽出しないようにする必要があります。また、固有表現タイプの正確性を向上させるために、抽出する固有表現がリストに含まれているかを確認するプロセスを強化するべきです。
2nd Prediction	[{'name': 'オーストラリア', 'type': '地名'}, {'name': 'ウエストパック銀行', 'type': '法人名'}, {'name': 'セントジョージ銀行', 'type': '法人名'}]
2nd Evaluation	抽出範囲: 20, タイプ: 20, 文脈依存性: 20, 抽出漏れ: 20, 過剰抽出: 20

また，1st Reflection の「オーストラリア・ドル」を抽出しない判断は妥当であるが，その説明は不正確である．正確には“「オーストラリア・ドル」の固有表現タイプは「通貨」であるものの，抽出対象として設定した固有表現タイプのリストに「通貨」がないため抽出すべきでない．”という説明になる．

4.3 考察と課題

以上の結果から, Self-Reflection は全体として有効性が認められるものの，一部の条件下での精度低下や，Eval の出力する評価値および Refl が提示する「改善すべき点」の不正確さに課題がある．改善策としては, Eval に与える Few-Shot 例の質を上げることや, Refl にも Few-Shot を付与することが挙げられる．特に，Eval に与えている Few-Shot の質の低さは，Pred と Eval の片方にのみ Few-Shot を与えた条件を比較した際に Pred のみの方が相対的に高い性能を示したことからも示唆される．具体的な改善案は，満点の事例だけでなく低評価の事例も加えることなどが考えられる．

また，本研究で用いたデータセットに含まれるテキストは平均約 58 文字であったが，実応用を考慮すると，1,000 文字以上の長文から NER を行うことも想定されるため，より長いテキストでの精度評価も求められる．

4.4 今後の展望

本研究で用いたデータセットにおける BERT の精度は 86%と報告されている [8] が, 本研究ではその水準に達していない．しかし，性能向上に向けて検討すべき要点は以下の通り数多く残されており，今後のさらなる手法改良が期待できる．

- GPT-NER [3] で示されている手法のように，Few-Shot 文を入力文の埋め込み類似度に基づいて選定するほか，特殊トークンを用いることでエンティティを明示的にマークさせながらテキストを生成する戦略を導入する．
- Agent Design Pattern Catalogue [4] において提示されている Self-Reflection 以外の手法（役割ベース，投票ベースなど）を組み込む．
- 固有表現タイプの拡張性を損なわない範囲で，各 Agent に特化した SFT を施した LLM を使用する．

5 結論

本研究では，LLM を用いた日本語 NER の精度向上手法として Self-Reflection が有効であることが示唆された．一方で，一部条件での精度低下や一部 Agent の不正確な出力といった課題も存在する．これらへの対策には，Few-Shot 設計の見直しが有効だと考えられる．今後は，エンティティマーキング戦略 [3] や，他の Agent Design Pattern（役割ベース，投票ベースなど）の導入 [4], および各 Agent に対する専用 SFT の実施を通じ，LLM ベース NER のさらなる性能向上を目指すことが可能である．以上の知見は，LLM ベースの日本語 NER タスクにおける精度向上手法の設計指針として，今後の進展にも寄与できると考えられる．

参考文献

[1] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Jonathan Larson. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. **arXiv preprint arXiv:2404.16130,** 2024.

[2] Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. **Proceedings of the 9th International Conference on Learning Representations,** 2021.

[3] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. GPT-NER: Named entity recognition via large language models. **arXiv preprint arXiv:2304.10428,** 2023.

[4] Yue Liu, Sin Kit Lo, Qinghua Lu, Liming Zhu, Dehai Zhao, Xiwei Xu, Stefan Harrer, and Jon Whittle. Agent Design Pattern Catalogue: A Collection of Architectural Patterns for Foundation Model based Agents. **arXiv preprint arXiv:2405.10467,** 2024.

[5] Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study. **arXiv preprint arXiv:2401.10825,** 2024.

[6] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, Shunyu Yao. Reflexion: language Agents with verbal reinforcement learning. **Proceedings of the 37th Conference on Neural Information Processing Systems,** 2023.

[7] Chatbot Arena. （オンライン）（引用日: 2024 年 12 月 11 日.）https://lmarena.ai/?leaderboard.

[8] 近江崇宏. Wikipedia を用いた日本語の固有表現抽出のデータセットの構築. **言語処理学会第 27 回年次大会発表論文集,** 2021.

[9] Wikipedia. （オンライン）（引用日: 2024 年 12 月 18 日.）https://ja.wikipedia.org/wiki/メインページ.

[10] LangGraph v0.2.22. Available at: https://github.com/langchain-ai/langgraph.

[11] How to return structured data from a model. （オンライン）（引用日: 2024 年 12 月 12 日.）https://python.langchain.com/v0.2/docs/how_to/structured_output/.