一般発表 | Track 1: 自然言語処理・機械学習基礎

2026年3月1日(日) 13:00 ~ 15:10 | A会場

[5A] 言語モデル応用

座長:秋岡 明香(明治大学)　コメンテータ:杉山 一成(大阪成蹊大学)　ジュニアコメンテータ:飛岡 憲(兵庫県立大学)

13:00 ~ 13:25

[5A-01] 文脈内学習に基づくテキストスタイル変換における例示組合せ最適化

*大庭 知也[1]、渡邊 千紘[1] (1. NTT株式会社)

13:25 ~ 13:50

[5A-02] 個人情報を活用する小規模言語モデルによる実装の検討

*川村 碧葵[1]、丸 千尋[2]、中野 美由紀[3]、小口 正人[1] (1. お茶の水女子大学、2. 中央大学、3. 津田塾大学)

13:50 ~ 14:15

[5A-03] 大規模言語モデルを用いたVADER感情分析手法の構造的拡張

*劉 浩東[1]、酒井 哲也[1] (1. 早稲田大学)

14:15 ~ 14:40

[5A-04] LLMに基づく説明可能なソーシャルネットワークの将来予測

*方 俊翔[1]、伊藤 寛祥[1]、徐 哲林[1]、森嶋 厚行[1] (1. 筑波大学)

© 2025-2026 DEIM2026実行委員会

文脈内学習に基づくテキストスタイル変換における例示組合せ最適化

大庭　知也† 渡邊　千紘†

† NTT コンピュータ&データサイエンス研究所　〒 180–8585 東京都武蔵野市緑町 3-9-11
E-mail: †{tomoya.ohba,ch.watanabe}@ntt.com

あらまし　大規模言語モデルの発展に伴い，事前学習済みの大規模言語モデルを用いてテキストスタイル変換を行うアプローチが提案されている．このようなアプローチにおいて，文脈内学習の枠組みに基づき例示を含むプロンプトを入力とする場合，例示組合せの選択がタスク性能に影響を与えることが指摘されている．テキストスタイル変換においては，単一のスタイルを持つテキストからなる非パラレルデータと，異なるスタイルを持つテキストの組からなるパラレルデータの 2 種類の形式を例示に含めることが可能である．しかし，例示に含める際に，これらの形式の違いがタスク性能に及ぼす影響は明らかになっていない．そこで，本研究では，パラレル・非パラレル両データを含みうるデータセットから最適な例示組合せを選択するアルゴリズムを提案し，異なる形式のデータセットに適用することで，パラレルデータにおける変換関係の情報の有無がテキストスタイル変換のタスク性能に与える影響を検証する．実験の結果，本研究における設定の下では，高精度なテキストスタイル変換を実現するために変換関係の情報が重要であることが示された．

キーワード　テキストスタイル変換，文脈内学習，プロンプト最適化，大規模言語モデル

1 はじめに

近年，大規模言語モデル（Large Language Model; LLM）の発展に伴い，質問応答 [22] や文章要約 [32]，機械翻訳 [13] など，様々な応用において高精度なテキスト処理が可能となっている [15]．本研究では，特にテキストスタイル変換（Text Style Transfer; TST）タスクに着目する．これは，テキストの（スタイルに非依存な）内容を保持したままスタイルのみを変換するタスクであり，チャットボットによる対話システム [16] やテキスト作成支援システム [1] [14] などへの応用が可能である．

TST を実現する主なアプローチとして，TST タスクに特化したモデルを設計し学習する方法と，事前学習済み LLM による推論に基づく方法が存在する．前者のアプローチにおいては，モデルの TST タスクに関する性能を直接最適化することが可能である一方，学習の計算コストが高いという問題がある [26]．後者のアプローチでは，事前学習済みの汎用 LLM を追加学習なしで用いることにより，推論時の計算コストのみで TST を実現することができる．また，パラメータ情報にアクセスできないブラックボックスなモデルを活用できるという利点があり，近年多くの手法が提案されている [27]．

上記のように，事前学習済み LLM を用いて与えられたタスクを解くアプローチにおいて，高精度なタスク性能を達成するための方法として，自動プロンプト最適化（Automatic Prompt Optimization; APO）が提案されている [34]．これは，LLM に入力するプロンプトをタスクに合わせて最適化することにより，モデルパラメータの追加学習なしでタスク精度を向上することを目的とした手法である．特に，タスク記述を含む指示，タスクの問題文とその回答例を記述する例示，クエリからプロンプトを構成する文脈内学習（In-Context Learning; ICL）[4] の枠組みにおいて，例示を最適化することの重要性が近年指摘されている [28]．

(a) 非パラレルデータ

スタイル	テキスト
positive	あの映画は素晴らしかった.
positive	ここのレストランの料理は美味しい.
negative	この部屋は狭くて汚い.
⋮	⋮

(b) パラレルデータ

ソース スタイル	ソーステキスト	ターゲット スタイル	ターゲットテキスト
positive	あの映画は素晴らしかった.	negative	あの映画はひどかった.
positive	ここのレストランの料理は美味しい.	negative	ここのレストランの料理はまずい.
negative	この部屋は狭くて汚い.	positive	この部屋は広くて綺麗だ.
⋮	⋮	⋮	⋮

図 1　(a) 非パラレルデータと (b) パラレルデータの例.

TST タスクにおいてプロンプトの例示最適化を行う際，タスク固有の特徴として，単一のテキストからなる非パラレルデータと，スタイル変換前後のテキストの組からなるパラレルデータの 2 種類の形式が存在する（図 1）[5] [9] [10]．パラレルデータは非パラレルデータと比べ，スタイル変換前後のテキストを比較可能な形の情報を含むため，例示として用いることでより高精度な TST を達成できることが期待される一方，データ収集のコストが高い．そのため，高精度な TST を実現するためにパラレルデータ形式の例示が必要であるのか，もしくは非パラレルデータ形式の例示のみで同程度のタスク性能を実現できるのかについて検証することは重要な研究課題である．このような検証を行うためには，パラレルデータにおける変換関係の情報を保持する設定と保持しない設定の両ケースについて，統一的な枠組みに基づき例示選択を行うことが可能な手法を構築し，それらの結果を比較する必要がある（図 2）．

そこで，本研究では，パラレル・非パラレル両形式のデータ

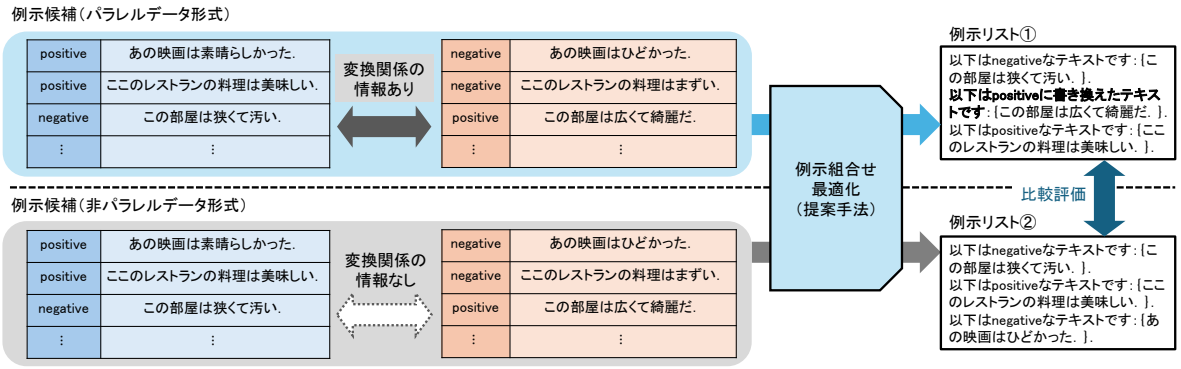

図 2　パラレルデータにおける変換関係の情報を保持する設定と保持しない設定における例示選択結果の比較評価の枠組み.

セットに関して得られる例示選択の結果を統一的に評価する枠組みを提案する．ここで，形式上パラレルデータは 2 つの，非パラレルデータは 1 つのテキストを含むことに注意する．そのため，それぞれを例示の 1 サンプルとして扱った場合，変換関係の情報の有無に加え，例示テキストの長さの違いがタスク性能に影響する可能性がある．したがって，提案法ではパラレルデータにおける変換前後のテキストをそれぞれ個別のサンプルとして扱ったうえで，その変換関係の情報に基づき貪欲法による例示最適化を行う．さらに，提案手法をパラレルデータあり・なしのデータセットに適用し，両ケースで選択された例示組合せのテスト性能を比較することにより，パラレルデータにおける変換関係の情報の有無が TST のタスク性能に与える影響について検証を行う．

2　関 連 研 究

2.1　テキストスタイル変換

TST を実現するためのアプローチとして，従来，TST タスクに特化した独自のモデルを設計し，学習する方法が提案されてきた．特に，1 節に述べた通り，異なるスタイルを持つテキストの組を含むパラレルデータは収集コストが高いため，非パラレルなデータから教師なし学習に基づく TST が可能なエンコーダ・デコーダ型のモデルが提案されている [6] [12] [23]. TST において，変換前（後）のスタイルとテキストをそれぞれソース（ターゲット）スタイル，ソース（ターゲット）テキストと呼ぶ．上記のモデルでは，まずエンコーダにおいて入力テキストのスタイル情報とスタイルに非依存な情報を分離した上で，デコーダにおいてターゲットテキストへの変換を行う．このアプローチにおいては，TST タスクに関する性能を直接最適化したモデルを獲得できる一方，学習時に高い計算コストがかかるという問題がある．

近年，上記の学習時における計算コストの問題を回避することが可能なアプローチとして，事前学習済み LLM を用いて TST を実現する手法が提案されている [17] [20] [24]．これらはデコーダ型の事前学習済み LLM に対し，ソーステキストのターゲットスタイルへの変換を指示するプロンプトを入力することで，ターゲットテキストの予測を行わせる方法である．これにより，推論時の計算コストのみで TST を実現でき，ブラックボックスなモデルも活用可能であるという利点がある．このようなアプローチにおいては，LLM に入力するプロンプトの設定により TST の性能が変わりうることが知られており，特に ICL の枠組みに基づき適切な例示の組合せをプロンプトに含めることで，タスク性能が向上することが報告されている [20].

2.2　自動プロンプト最適化

2.1 節に述べた通り，事前学習済み LLM の推論を通して TST を実現する手法においては，LLM に入力するプロンプトを適切に定義することが重要となる．一般に，事前学習済み LLM を用いて与えられたタスクを解くアプローチにおいて，タスク性能を最大化するプロンプトを推定する APO の手法が提案されている [2] [19] [31] [34]．特に，APO の性能を検証することを目的として，TST タスクに手法を適用した例も存在している [3] [8] [29]．ただし，これらの研究においては，プロンプトにおける例示組合せの最適化を実現する手法は提案されていない.

近年，ICL の枠組みに基づき指示と例示からなるプロンプト構成を想定する場合において，例示を最適化することの重要性が指摘されている [28]．一般のタスクに対し，プロンプト内で提示する例示組合せの最適化を行う手法はすでに提案されているが [25] [30]，これらの研究においては TST タスク固有の特徴であるパラレル・非パラレル両データの存在を考慮した手法は提案されていない.

2.3　本研究の位置づけ

2.1 節に述べたように，近年，事前学習済み汎用 LLM を用いて TST タスクを解くアプローチが提案されている．一方で，2.2 節に述べたように，LLM を用いて一般のタスクを解く際の性能向上を目的として APO が提案されている．特に，プロンプトに含まれる例示組合せを適切に設定することにより，TST のタスク性能が向上することが期待されるが，TST タスク固有の特徴であるパラレル・非パラレル両データの存在を考慮した上で最適な組合せを推定する手法は提案されていない.

本研究では，パラレル・非パラレル両データを含みうるデータセットから TST タスクにおいて最適な例示組合せを推定す

る手法を提案する．提案手法は，事前学習済み LLM を用いて TST タスクを解く設定における APO の手法の一種であり，提案手法により選択された例示組合せをプロンプトに含めることでタスク性能の向上を見込むことができる．

3 提案手法

本研究では，ICL の枠組みに基づき事前学習済み LLM を用いて TST タスクを解く設定において，プロンプト内で提示する例示組合せの最適化を行うアルゴリズムを提案する．提案手法の概要を図 3 に示す．まず，提案手法で用いるデータセットについての説明を述べた後（3.1 節），提案手法のアルゴリズムについて説明する（3.2 節）．

3.1 データセット

本研究では，パラレル・非パラレル両データが含まれうるデータセットの中から TST タスクにおいて最適な例示組合せを選択する問題を考える．与えられた TST タスクにおいて扱うスタイルの集合を $\mathcal{S} = \{s_1, \ldots, s_K\}$ とする（ただし，K をスタイルの数，$\mathcal{T}$ を存在しうるすべてのテキストの集合とし，$i = 1, \ldots, K$ について $s_i \in \mathcal{T}$ とする）．本稿においては，スタイルとして肯定的（positive），否定的（negative）の 2 種類のみを考える（$\mathcal{S} = \{\text{“positive”}, \text{“negative”}\}$）．TST とは，あるスタイル $s \in \mathcal{S}$ を持つテキスト $t \in \mathcal{T}$ とターゲットスタイル $\tilde{s} \in \mathcal{S}$ の組合せが与えられたとき，テキスト t が持つスタイルに非依存な内容を保持したままスタイル $\tilde{s}$ に変換するタスクとして定義することができる．

本研究では，TST のデータセットとして，以下の 2 種類の形式を想定する．

- **非パラレルデータセット**：非パラレルデータセット $\mathcal{D}_1 = \{(s_1, t_1), \ldots, (s_{n_1}, t_{n_1})\}$ は，スタイル $s \in \mathcal{S}$ と，スタイル s を持つテキスト $t \in \mathcal{T}$ の組 (s, t) の集合で表される．ただし，n_1 はサンプルサイズを表す．
- **パラレルデータセット**：パラレルデータセット $\mathcal{D}_2 = \{(s_1, t_1, \tilde{s}_1, \tilde{t}_1), \ldots, (s_{n_2}, t_{n_2}, \tilde{s}_{n_2}, \tilde{t}_{n_2})\}$ は，ソーススタイル $s \in \mathcal{S}$ と，スタイル s を持つソーステキスト $t \in \mathcal{T}$ と，ターゲットスタイル $\tilde{s} \in \mathcal{S}$ と，テキスト t をスタイル $\tilde{s}$ に変換したターゲットテキスト $\tilde{t} \in \mathcal{T}$ の組 $(s, t, \tilde{s}, \tilde{t})$ の集合で表される．ただし，n_2 はサンプルサイズを表す．

上記の形式で与えられる非パラレルデータセット $\mathcal{D}_1$ とパラレルデータセット $\mathcal{D}_2$ に基づき [1]，例示組合せの最適化を行うために，まずこれらのデータセットを学習データセット $\mathcal{D}_{\text{train}} \subset \mathcal{D}_2$, テストデータセット $\mathcal{D}_{\text{test}} \subset \mathcal{D}_2$, 例示データセット $\mathcal{D}_{\text{exemplar}} \subset \mathcal{D}_1 \cup \mathcal{D}_2$ に分割する．ただし，いずれのデータセットも少なくとも 1 つの要素を含み，かついずれの異なる 2 つのデータセットの組も互いに素であるように定義しておく．

さらに，例示データセット $\mathcal{D}_{\text{exemplar}}$ のうち非パラレルデータの集合を $\mathcal{D}_{\text{NP}} = \{(s_1^{\text{NP}}, t_1^{\text{NP}}), \ldots, (s_{n_{\text{NP}}}^{\text{NP}}, t_{n_{\text{NP}}}^{\text{NP}})\}$, パラレルデータの集合を $\mathcal{D}_{\text{P}} = \{(s_1^{\text{P}}, t_1^{\text{P}}, \tilde{s}_1^{\text{P}}, \tilde{t}_1^{\text{P}}), \ldots, (s_{n_{\text{P}}}^{\text{P}}, t_{n_{\text{P}}}^{\text{P}}, \tilde{s}_{n_{\text{P}}}^{\text{P}}, \tilde{t}_{n_{\text{P}}}^{\text{P}})\}$ とし（$\mathcal{D}_{\text{NP}} \subset \mathcal{D}_1$, $\mathcal{D}_{\text{P}} \subset \mathcal{D}_2$, $\mathcal{D}_{\text{exemplar}} = \mathcal{D}_{\text{NP}} \cup \mathcal{D}_{\text{P}}$），これらに基づき以下の例示集合 $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$ を定義する．ただし，$(s_i^{\text{P}}, t_i^{\text{P}}, \tilde{s}_i^{\text{P}}, \tilde{t}_i^{\text{P}})$ を $\mathcal{D}_{\text{P}}$ の i 番目の要素とし，$(s_i^{\text{NP}}, t_i^{\text{NP}})$ を $\mathcal{D}_{\text{NP}}$ の i 番目の要素とする．まず，ソーステキストからなる例示の集合として $\mathcal{X}_1$ を定義する．これは，スタイルに非依存な内容が同一であるようなテキストの連なりにおいて，1 番目に提示されるテキストの集合とも言い換えることができる．

$$\mathcal{X}_1 = \{x_1^{(1)}, \ldots, x_{m1}^{(1)}\} = u(\{\tilde{x}_1^{(1)}, \ldots, \tilde{x}_{n_{\text{P}}+n_{\text{NP}}}^{(1)}\}),$$

$$\tilde{x}_i^{(1)} = \begin{cases} h_1(s_i^{\text{P}}, t_i^{\text{P}}), \text{for } i = 1, \ldots, n_{\text{P}}, \\ h_1(s_{i-n_{\text{P}}}^{\text{NP}}, t_{i-n_{\text{P}}}^{\text{NP}}), \text{for } i = n_{\text{P}}+1, \ldots, n_{\text{P}}+n_{\text{NP}}, \end{cases}$$

$$h_1(s, t) = \text{“Here is a text, which is } s\text{: } \{t\}. \text{”} \quad (1)$$

ここで，u は入力のテキスト集合から重複を除いた集合を出力する関数とする．次に，ターゲットテキストからなる例示の集合として $\mathcal{X}_2$ を定義する．これは，上記のテキストの連なりにおいて，2 番目に提示されるテキストの集合となる．

$$\mathcal{X}_2 = \{x_1^{(2)}, \ldots, x_{m2}^{(2)}\} = u(\{\tilde{x}_1^{(2)}, \ldots, \tilde{x}_{n_{\text{P}}}^{(2)}\}),$$

$$\tilde{x}_i^{(2)} = h_2(\tilde{s}_i^{\text{P}}, \tilde{t}_i^{\text{P}}), \text{for } i = 1, \ldots, n_{\text{P}},$$

$$h_2(s, t) =$$

$$\text{“Here is a rewrite of the text, which is } s\text{: } \{t\}. \text{”} \quad (2)$$

さらに，ターゲットテキストからなる新たな例示の集合として $\mathcal{X}_3$ を定義する．これは，上記のテキストの連なりにおいて，3 番目以降に提示されるテキストの集合となる．

$$\mathcal{X}_3 = \{x_1^{(3)}, \ldots, x_{m3}^{(3)}\} = u(\{\tilde{x}_1^{(3)}, \ldots, \tilde{x}_{n_{\text{P}}}^{(3)}\}),$$

$$\tilde{x}_i^{(3)} = h_3(\tilde{s}_i^{\text{P}}, \tilde{t}_i^{\text{P}}), \text{ for } i = 1, \ldots, n_{\text{P}},$$

$$h_3(s, t) =$$

$$\text{“Here is another rewrite of the text, which is } s\text{: } \{t\}. \text{”} \quad (3)$$

最後に，$\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ の要素同士の変換関係の情報を参照可能にしておくため，これらの集合の各要素に対応するソーステキストの番号を出力する関数 $\phi : \mathcal{T} \mapsto \mathbb{N}$ を以下のように定義する．まず，$\mathcal{X}_1$ の各要素 $x_i^{(1)}$ $(i = 1, \ldots, m_1)$ について，$\phi(x_i^{(1)}) = i$ とする．次に，$\mathcal{X}_2$ の各要素 $x_i^{(2)}$ $(i = 1, \ldots, m_2)$ について，対応するソーステキストが含まれる $\mathcal{X}_1$ の要素が $x_j^{(1)}$ であるとき，$\phi(x_i^{(2)}) = j$ とする．さらに，$\mathcal{X}_3$ の各要素 $x_i^{(3)}$ $(i = 1, \ldots, m_3)$ について，対応するソーステキストが含まれる $\mathcal{X}_1$ の要素が $x_j^{(1)}$ であるとき，$\phi(x_i^{(3)}) = j$ とする．

3.2 貪欲法に基づく例示組合せ最適化

本研究の目的は，3.1 節で定義した学習データセット $\mathcal{D}_{\text{train}}$ について，TST のタスク性能を最大化する例示組合せを例示集合 $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ から選択することである．すべての可能な例示組合せの中から学習損失を最小化する組合せを選択することは計算的に困難であるため，本研究では貪欲法に基づく例示組合せ最適化手法を提案する（アルゴリズム 1）．これは，具

1：ここで，非パラレルデータセット $\mathcal{D}_1$ については空集合でもよい．

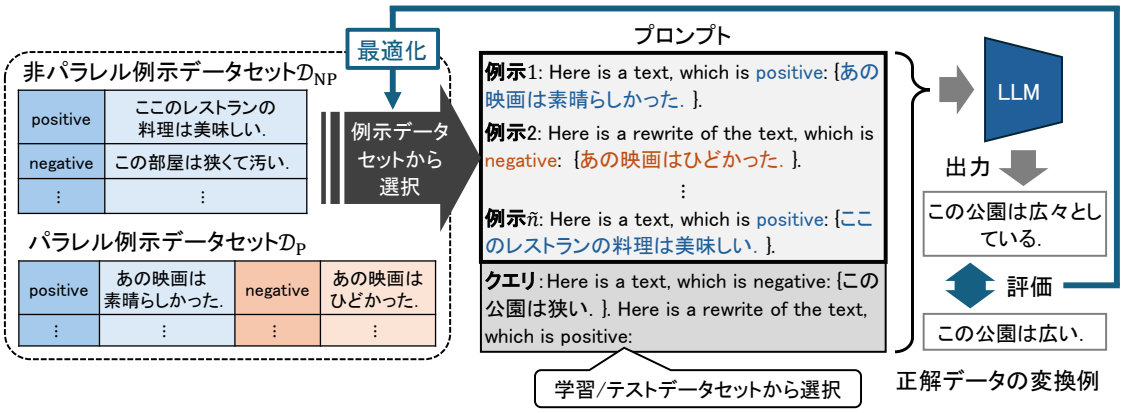

図 3　文脈内学習に基づくテキストスタイル変換における例示組合せ最適化の枠組み.

体的には例示数 $\tilde{n} \in \mathbb{N}$ が与えられたとき，学習データセット $\mathcal{D}_{\mathrm{train}}$ における損失を最小化するようなサイズ $\tilde{n}$ の例示リスト $\hat{\mathcal{Y}} = [y_1, \ldots, y_{\tilde{n}}]$ を選択する手法である．

貪欲法に基づくナイーブな例示選択方法として，1 番目の例示から順に学習損失を最小化する候補を選択していく方法が考えられるが，この方法では例示の選択順が固定されているため，局所最適解に陥りやすい．そのため，本研究では，例示の選択順をランダムに変えて T 回の例示組合せ選択を行い，それらの結果のうち学習損失を最小化する組合せを例示リスト $\hat{\mathcal{Y}}$ として出力することを提案する．ただし，T は任意の自然数とする．

提案手法においては，3.1 節に述べたスタイルに非依存な内容が同一であるようなテキストの連なりを表す例示リスト $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_C$ を要素とする例示グループリスト $\mathcal{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_C]$ を扱う．ただし，C を $\mathcal{G}$ の要素数とする．これにより，変換関係の矛盾が生じないように，位置に応じた例示集合の中から候補を選択することが可能になる．具体的には，各ステップ $i \in \{1, \ldots, T\}$ において，例示グループリスト $\mathcal{G}^{(i)}$ の初期値を空リストとする．各 $j \in \{1, \ldots, \tilde{n}\}$ について，一様分布から位置 r を生成し，現在の例示グループリスト $\mathcal{G}^{(i)}$ における位置 r に新たな例示を追加することを考える．ここで，位置 r に応じて，変換関係の矛盾なく追加することが可能な例示候補の集合として $\mathcal{X}$ を定義し（アルゴリズム 1 の 6–16 行目），その要素から最適な例示候補の選択を行う．

最適な例示候補の選択にあたり，各候補を現在の例示グループリスト $\mathcal{G}^{(i)}$ に追加した結果得られる例示リストについて，それを用いた場合の学習損失を計算する必要がある．このため，以下の関数 α, ζ を定義しておく．まず，例示グループリスト $\mathcal{G}$ の位置 $i \in \{1, \ldots, C+1\}$ に新たな例示 x を追加する操作を表す関数 α を以下のように定義する．(1) $i = 1$ の場合，$\alpha(\mathcal{G}, i, x) = [[x], \boldsymbol{g}_1, \ldots, \boldsymbol{g}_C]$．(2) $i \neq 1$ の場合，$\boldsymbol{g}_{i-1} = [x_1, \ldots, x_m]$ とする．(2a) $\phi(x) = \phi(x_1)$ かつ $x \neq x_1$ の場合，$\alpha(\mathcal{G}, i, x) = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{i-2}, [x_1, \ldots, x_m, x], \boldsymbol{g}_i, \ldots, \boldsymbol{g}_C]$．(2b) $\phi(x) \neq \phi(x_1)$ もしくは $x = x_1$ の場合，$\alpha(\mathcal{G}, i, x) = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{i-1}, [x], \boldsymbol{g}_i, \ldots, \boldsymbol{g}_C]$．次に，与えられた例示グループリスト $\mathcal{G}$ の各要素に含まれる例示を昇順に並べることで，例示リストに変換する関数 ζ を定義する．これは，$i = 1, \ldots, C$ について $\boldsymbol{g}_i = [x_1^{(i)}, \ldots, x_{m_i}^{(i)}]$ とするとき，以下のように定義される．

$$\zeta(\mathcal{G}) = [x_1^{(1)}, \ldots, x_{m_1}^{(1)}, \ldots, x_1^{(C)}, \ldots, x_{m_C}^{(C)}] \tag{4}$$

これらの関数に基づき，現在の例示グループリスト $\mathcal{G}^{(i)}$ の位置 r に j 個目の例示 x を追加することで得られる例示リストを $\mathcal{Y} = [y_1, \ldots, y_j] = \zeta(\alpha(\mathcal{G}^{(i)}, r, x))$ と表すことができる．

上記の例示リスト $\mathcal{Y}$ に対応する学習損失を計算するため，与えられた学習サンプル $\boldsymbol{q} = (s_{\mathrm{Q}}, t_{\mathrm{Q}}, \tilde{s}_{\mathrm{Q}}, \tilde{t}_{\mathrm{Q}}) \in \mathcal{D}_{\mathrm{train}}$ と例示リスト $\mathcal{Y}$ からプロンプト $p \in \mathcal{T}$ を出力する関数 ρ を定義する．一般的に，LLM に対するプロンプトは指示，例示，クエリの 3 点から構成されるが，本稿では簡単のため，例示とクエリのみに基づく以下の定義を用いる．

$$\begin{aligned} p = \rho(\boldsymbol{q}, \mathcal{Y}) &= y_1 + \cdots + y_j + h_{\mathrm{Q}}(s_{\mathrm{Q}}, t_{\mathrm{Q}}, \tilde{s}_{\mathrm{Q}}), \\ h_{\mathrm{Q}}(s, t, \tilde{s}) &= \text{“Here is some } s \text{ text: } \{t\}. \\ &\quad \text{Here is a rewrite of the text, which is } \tilde{s}\text{: } \{\text{”} \end{aligned} \tag{5}$$

ここで，$+$ は 2 つのテキストを結合し 1 つのテキストとする操作を表す．学習データセットを

$$\begin{aligned} \mathcal{D}_{\mathrm{train}} &= \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|\mathcal{D}_{\mathrm{train}}|}\}, \\ \boldsymbol{x}_i &= (s_i, t_i, \tilde{s}_i, \tilde{t}_i) \text{ for } i = 1, \ldots, |\mathcal{D}_{\mathrm{train}}| \end{aligned} \tag{6}$$

としたとき，与えられた例示リスト $\mathcal{Y}$ の学習損失は以下で定義される．

$$\mathcal{L}_{\mathrm{train}}(\mathcal{Y}) = \frac{1}{|\mathcal{D}_{\mathrm{train}}|} \sum_{i=1}^{|\mathcal{D}_{\mathrm{train}}|} d(\tilde{t}_i, f_{\mathrm{LLM}}(\rho(\boldsymbol{x}_i, \mathcal{Y}))) \tag{7}$$

ここで，$f_{\mathrm{LLM}} : \mathcal{T} \mapsto \mathcal{T}$ はテキストを入力とし，テキストを出力する任意の言語モデルを表す関数とする．また，$d(t_1, t_2)$ は 2 つのテキスト $t_1, t_2 \in \mathcal{T}$ 間の非類似度を評価する関数を表す．本稿を通し，テキスト t_1, t_2 をそれぞれ Sentence Transformer [21] に入力して得られる埋め込みベクトル $\boldsymbol{v}_1, \boldsymbol{v}_2$ を用いて，以下のように定義する．

$$d(t_1, t_2) = (1 - \cos(\boldsymbol{v}_1, \boldsymbol{v}_2))/2 \tag{8}$$

Algorithm 1 貪欲法に基づく例示組合せ最適化アルゴリズム

Input: 学習データセット $\mathcal{D}_{\text{train}}$, 例示集合 $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, 例示数 $\tilde{n}$, ステップ数 T

Output: 例示リスト $\hat{\mathcal{Y}}$

1: **for** $i = 1, \ldots, T$ **do**
2: $\quad \mathcal{G}^{(i)} = []$
3: $\quad$ **for** $j = 1, \ldots, \tilde{n}$ **do**
4: $\quad\quad r \sim \text{Uniform}(1, \ldots, |\mathcal{G}^{(i)}| + 1)$
5: $\quad\quad$ {$\mathcal{G}^{(i)}$ の r 番目の位置に追加する例示候補の集合 $\mathcal{X}$ を定義}
6: $\quad\quad$ **if** $r = 1$ **then**
7: $\quad\quad\quad \mathcal{X} = \mathcal{X}_1$
8: $\quad\quad$ **else**
9: $\quad\quad\quad \boldsymbol{g}_{r-1} =: [x_1, \ldots, x_m]$
10: $\quad\quad\quad$ **if** $|\boldsymbol{g}_{r-1}| = 1$ **then**
11: $\quad\quad\quad\quad \mathcal{X}' = \{x \in \mathcal{X}_2 | \phi(x) = \phi(x_1)\}$
12: $\quad\quad\quad$ **else**
13: $\quad\quad\quad\quad \mathcal{X}' = \{x \in \mathcal{X}_3 | \phi(x) = \phi(x_1)\}$
14: $\quad\quad\quad$ **end if**
15: $\quad\quad\quad \mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}'$
16: $\quad\quad$ **end if**
17: $\quad\quad \hat{x} = \arg\min_{x \in \mathcal{X}} \mathcal{L}_{\text{train}}(\zeta(\alpha(\mathcal{G}^{(i)}, r, x)))$
18: $\quad\quad \mathcal{G}^{(i)} \leftarrow \alpha(\mathcal{G}^{(i)}, r, \hat{x})$
19: $\quad$ **end for**
20: **end for**
21: {T 回の例示選択結果のうち，学習損失を最小化する組合せを選択}
22: $\hat{\mathcal{G}} = \arg\min_{\mathcal{G} \in \{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(T)}\}} \mathcal{L}_{\text{train}}(\zeta(\mathcal{G}))$
23: $\hat{\mathcal{Y}} = \zeta(\hat{\mathcal{G}})$

式 (8) の定義に基づく学習損失の最小化は，負の BERTScore [33] を用いることと等価である．式 (7) の学習損失は全学習サンプルに基づき計算されるが，本研究では計算量削減のため，アルゴリズム 1 の各 (i, j) についてランダムに選択した n_{train} 個の学習サンプルを代わりに用いて損失の計算を行うこととした．このようにして計算される学習損失を最小化する例示候補 $\hat{x}$ を集合 $\mathcal{X}$ から選択し，例示グループリスト $\mathcal{G}^{(i)}$ に追加することを繰り返すことで，$\tilde{n}$ 個の例示組合せの選択を行う．最終的に，T 回の試行で得られた例示グループリスト $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(T)}$ から学習損失を最小化する結果 $\hat{\mathcal{G}}$ を選び，例示リスト $\hat{\mathcal{Y}}$ に変換して出力する（アルゴリズム 1 の 22–23 行目）．

上記のアルゴリズム 1 に基づき選択された例示リスト $\hat{\mathcal{Y}}$ のテスト損失を評価することで，未知のデータに対する TST のタスク性能を測ることができる．テストデータセットを

$$\begin{aligned} &\mathcal{D}_{\text{test}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|\mathcal{D}_{\text{test}}|}\}, \\ &\boldsymbol{x}_i = (s_i, t_i, \tilde{s}_i, \tilde{t}_i) \text{ for } i = 1, \ldots, |\mathcal{D}_{\text{test}}| \end{aligned} \tag{9}$$

としたとき，与えられた例示リスト $\mathcal{Y}$ のテスト損失は以下で定義される．

$$\mathcal{L}_{\text{test}}(\mathcal{Y}) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{i=1}^{|\mathcal{D}_{\text{test}}|} d(\tilde{t}_i, f_{\text{LLM}}(\rho(\boldsymbol{x}_i, \mathcal{Y}))) \tag{10}$$

4 実　　験

本節では，パラレルデータにおける変換関係の情報の有無が TST のタスク性能に与える影響を検証することを目的として，変換関係の情報を保持する設定と保持しない設定の両ケースについて実データを用いて提案手法による例示選択を行い，それらの結果の比較を行う．

4.1 実験設定

TST の実データセットとして，Yelp データセット [11] を用いて実験を行った．本データセットに含まれるサンプルは全てパラレルデータの形式を持ち，スタイル “positive” からスタイル “negative” への変換例と，スタイル “negative” からスタイル “positive” への変換例がそれぞれ 500 個ずつ含まれる．本データセットをランダムに分割することで，3.1 節に述べたように学習データセット $\mathcal{D}_{\text{train}}$, テストデータセット $\mathcal{D}_{\text{test}}$, 例示データセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ を定義した．ここで，スタイル “positive” からスタイル “negative” への変換例と，スタイル “negative” からスタイル “positive” への変換例のそれぞれについて，データセット $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{test}}$, $\mathcal{D}^{(0)}_{\text{exemplar}}$ のサンプルサイズをそれぞれ 225, 225, 50 とした．また，$n^{(0)} = |\mathcal{D}^{(0)}_{\text{exemplar}}|$ として，ここで得られた例示データセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ を以下のように定義しておく．

$$\begin{aligned} &\mathcal{D}^{(0)}_{\text{exemplar}} = \{\boldsymbol{x}^{(0)}_1, \ldots, \boldsymbol{x}^{(0)}_{n^{(0)}}\}, \\ &\boldsymbol{x}^{(0)}_i = (s^{(0)}_i, t^{(0)}_i, \tilde{s}^{(0)}_i, \tilde{t}^{(0)}_i) \text{ for } i = 1, \ldots, n^{(0)} \end{aligned} \tag{11}$$

本研究では，例示における変換関係の情報の有無が TST の性能に与える影響を検証するため，上記の例示データセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ から以下の 4 つの例示データセット $\mathcal{D}_{\text{exemplar}}$ を設定し，実験を行った（図 4）．

- **設定 ST-R**：$\mathcal{D}_{\text{exemplar}} = \mathcal{D}^{(0)}_{\text{exemplar}}$ とする．これは，元のデータセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ に含まれるソース・ターゲットの両データを用いた上で，それらの変換関係の情報も考慮する（つまり，パラレルデータとして扱う）設定に相当する．
- **設定 ST**：以下の定義を用いる．これは，元のデータセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ に含まれるソース・ターゲットの両データを用いるが，それらの間の変換関係の情報を考慮しない（つまり，非パラレルデータとして扱う）設定に相当する．

$$\begin{aligned} &\mathcal{D}_{\text{exemplar}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{2n^{(0)}}\}, \\ &\boldsymbol{x}_i = \begin{cases} (s^{(0)}_i, t^{(0)}_i) & \text{for } i = 1, \ldots, n^{(0)} \\ (\tilde{s}^{(0)}_i, \tilde{t}^{(0)}_i) & \text{for } i = n^{(0)} + 1, \ldots, 2n^{(0)} \end{cases} \end{aligned} \tag{12}$$

- **設定 S**：以下の定義を用いる．これは，元のデータセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ に含まれるソーステキストのデータのみを非パラレルデータとして用いる設定に相当する．

$$\begin{aligned} &\mathcal{D}_{\text{exemplar}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n^{(0)}}\}, \\ &\boldsymbol{x}_i = (s^{(0)}_i, t^{(0)}_i) \text{ for } i = 1, \ldots, n^{(0)} \end{aligned} \tag{13}$$

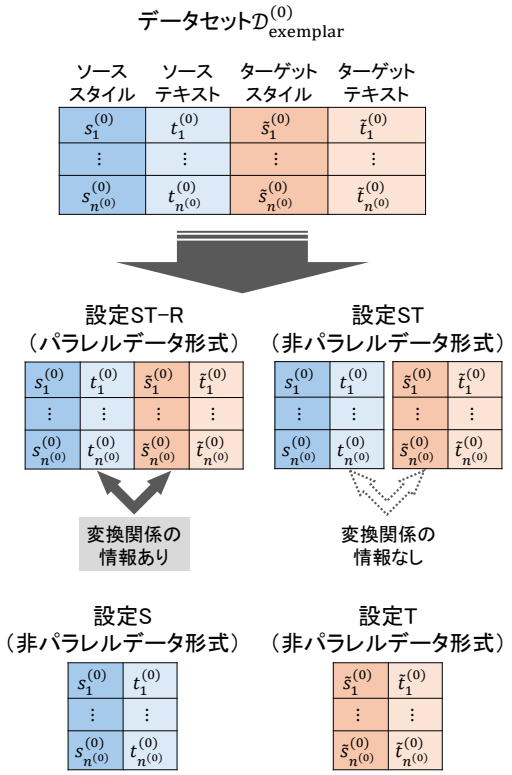

図 4　設定 ST-R, ST, S, T における例示データセットの構成.

- **設定 T**：以下の定義を用いる．これは，元のデータセット $\mathcal{D}^{(0)}_{\text{exemplar}}$ に含まれるターゲットテキストのデータのみを非パラレルデータとして用いる設定に相当する．

$$\mathcal{D}_{\text{exemplar}} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n^{(0)}}\},$$
$$\boldsymbol{x}_i = (\tilde{s}_i^{(0)}, \tilde{t}_i^{(0)}) \text{ for } i = 1, \ldots, n^{(0)} \tag{14}$$

上記の 4 つの設定においてそれぞれ定義された例示データセット $\mathcal{D}_{\text{exemplar}}$ から，式 (1), (2), (3) に基づき例示集合 $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$ を定義した [2].

上記の例示データセット $\mathcal{D}_{\text{exemplar}}$ を用いて，アルゴリズム 1 に基づき例示組合せの最適化を行った．ここで，関数 f_{LLM} の表す言語モデルとして温度パラメータを 0 とした GPT-4o mini [7] を用い，式 (8) におけるテキストの埋め込みベクトルの計算には paraphrase-MiniLM-L6-v2 [21] を用いた．また，アルゴリズム 1 の入力における例示数を $\tilde{n} = 5$, ステップ数を $T = 5$ とし，アルゴリズム 1 の各 (i, j) において学習損失の計算に用いるサンプルサイズを $n_{\text{train}} = 20$ とした．さらに，実験では計算量削減のため，アルゴリズム 1 の各 (i, j) においてそれまでに選ばれた位置 r_{ij} の情報を軌跡 $\boldsymbol{r}^{(ij)} = [r_{i1}, \ldots, r_{ij}]$ として保持しておき，過去のステップにおいて一致する軌跡が存在する場合はその時の選択結果 $\hat{x}$ を複製して用いることとした．このようにして設定 ST-R, ST, S, T の下で選ばれた各例示組合せについて，式 (10) に基づきテスト性能の評価を行った．

2：ただし，設定 ST, S, T においては，例示データセット $\mathcal{D}_{\text{exemplar}}$ は非パラレルデータのみから構成されるため，$\mathcal{X}_2$, $\mathcal{X}_3$ は空集合とした．

4.2 実験結果

4.1 節に述べた各設定において，アルゴリズム 1 を用いて選択された例示リスト $\hat{\mathcal{Y}}$ とその学習/テスト損失をそれぞれ表 1, 2 に示す．ここで，ベースラインとして，例示なし（zero-shot）の場合の学習/テスト損失も記載した．また，表 3 に，アルゴリズム 1 における各ステップで得られた例示グループリスト $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(T)}$ から構成される例示リスト $\zeta(\mathcal{G}^{(1)}), \ldots, \zeta(\mathcal{G}^{(T)})$ の学習/テスト損失の平均値と標準偏差を示す．

表 2, 3 より，ソース・ターゲットの両データを用いた上で，それらの変換関係の情報も考慮する設定 ST-R が最良のテスト性能を達成することが示された．また，同じくソース・ターゲットの両データを用いた場合でも，それらの変換関係の情報を考慮しない設定 ST においては，ソーステキストのみ/ターゲットテキストのみのデータを用いた場合（設定 S/T）と同程度のテスト性能しか達成されなかった．これらの結果から，本稿における実験設定の下では，高精度な TST を実現するためにパラレルデータ形式の例示が必要であることが示された．

また，表 1 から，実際に各設定の下で選択された例示組合せを比較することができる．例えば，設定 ST-R の下で選択された例示組合せにおいては，3, 4 番目にパラレルデータ形式の例示の組が選択されていることが分かる．また，ソーステキストのデータを含む設定 ST-R, ST, S のすべての場合において，例示として共通のテキスト “Here is a text, which is negative: {it did n’t get finished .}.” が選択されている．この他にも，テキスト “Here is a text, which is negative: {worst chicken parmesan i have ever had.}.,” “Here is a text, which is negative: {not one of my regular spots in scottsdale}.” は複数の設定下で例示として選択されている．これらの結果から，例示データセットの中でも学習損失の最小化に寄与するような特定の例示の存在が示唆された．

5 おわりに

本研究では，事前学習済み LLM を用いて ICL の枠組みに基づき TST タスクを解くアプローチにおいて，例示組合せの最適化を行うアルゴリズムを提案した．特に，TST タスクにおいてはパラレル・非パラレルの 2 種類の形式に基づくデータが存在することに着目し，これらの形式の違いがタスク性能に与える影響を検証するため，両形式のデータセットに関して得られる例示選択の結果を統一的に評価可能な枠組みを提案した．本研究における実験設定の下では，高精度な TST を実現するためにパラレルデータにおけるテキストの変換関係の情報が必要であることが示された．

本研究においては，例示数 $\tilde{n}$ を固定した上で式 (10) の評価規準に基づき異なる設定下でのタスク性能を比較したが，異なる例示数の設定や，BLEU [18] などの異なる評価規準を用いた場合についても同様の検証を行うことは，今後の課題である．また，TST タスクにおいて最良の性能を達成可能な例示数を選択することも，重要な課題である．

表 1　各設定において選択された例示組合せの結果.

設定	選択された例示リスト
ST-R	Here is a text, which is negative: {safeway has officially lost my business to sprouts , & fresh & easy .}.
	Here is a text, which is negative: {it did n't get finished .}.
	Here is a text, which is positive: {my husband and i enjoyed our 3rd anniversary here .}.
	Here is a rewrite of the text, which is negative: {my husband and i didn't enjoy our 3rd anniversary hear .}.
	Here is a text, which is negative: {as soon as they delivered i was like ugh .}.
ST	Here is a text, which is negative: {it did n't get finished .}.
	Here is a text, which is negative: {worst chicken parmesan i have ever had.}.
	Here is a text, which is negative: {not one of my regular spots in scottsdale}.
	Here is a text, which is negative: {fish tacos were the worst I had}.
	Here is a text, which is positive: {Ra was a chain, wow im impressed}.
S	Here is a text, which is negative: {it did n't get finished .}.
	Here is a text, which is negative: {i 'm sure they must get it right some days but not this day .}.
	Here is a text, which is positive: {i loved the ribs more than the chicken .}.
	Here is a text, which is negative: {in turn my legs are burnt from the noodles and all over the floor .}.
	Here is a text, which is positive: {the lunch and dinner items are very good as well .}.
T	Here is a text, which is positive: {I'm one of the corn people. }.
	Here is a text, which is negative: {not one of my regular spots in scottsdale}.
	Here is a text, which is negative: {worst chicken parmesan i have ever had.}.
	Here is a text, which is negative: {came here without my family .}.
	Here is a text, which is positive: {The short rib hash was perfectly cooked and juicy.}.

表 2　選択された例示リスト $\hat{\mathcal{Y}}$ の学習/テストデータにおける損失.

設定	学習損失	テスト損失
ST-R	**0.1635**	**0.1680**
ST	0.1840	0.1979
S	0.1894	0.1945
T	0.1920	0.2058
Zero-shot	0.2012	0.2050

表 3　T 回の例示選択結果の学習/テストデータにおける損失 ($T = 5$).

設定	学習損失	テスト損失
ST-R	$\mathbf{0.1722 \pm 0.0094}$	$\mathbf{0.1685 \pm 0.0004}$
ST	0.1888 ± 0.0032	0.2001 ± 0.0014
S	0.1928 ± 0.0038	0.1930 ± 0.0009
T	0.1963 ± 0.0034	0.2051 ± 0.0006

文　献

[1] Magdalena Badura, Michał Lampert, and Rafał Dreżewski. System supporting poetry generation using text generation and style transfer methods. *Procedia Computer Science*, Vol. 207, pp. 3310–3319, 2022.

[2] Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3201–3219, 2024.

[3] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3369–3391, 2022.

[4] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1128, 2024.

[5] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: exploration and evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, No. 1, 2018.

[6] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, pp. 1587–1596, 2017.

[7] Aaron Hurst, et al. GPT-4o system card. arXiv:2410.21276, 2024.

[8] Yasaman Jafari, Dheeraj Mekala, Rose Yu, and Taylor Berg-Kirkpatrick. MORL-prompt: An empirical analysis of multi-objective reinforcement learning for discrete prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9878–9889, 2024.

[9] Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pp. 10–19, 2017.

[10] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 424–434, 2019.

[11] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1865–1874, 2018.

[12] Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. Towards fine-

grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2013–2022, 2019.

[13] Chenyang Lyu, Zefeng Du, Jitao Xu, Yitao Duan, Minghao Wu, Teresa Lynn, Alham Fikri Aji, Derek F. Wong, and Longyue Wang. A paradigm shift: The future of machine translation lies with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 1339–1352, 2024.

[14] Do June Min, Veronica Perez-Rosas, Ken Resnicow, and Rada Mihalcea. VERVE: Template-based ReflectiVE rewriting for MotiVational IntErviewing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10289–10302, 2023.

[15] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. arXiv:2402.06196, 2024.

[16] Sourabrata Mukherjee, Vojtěch Hudeček, and Ondřej Dušek. Polite chatbot: A text style transfer application. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 87–93, 2023.

[17] Sourabrata Mukherjee, Atul Kr. Ojha, and Ondrej Dusek. Are large language models actually good at text style transfer? In *Proceedings of the 17th International Natural Language Generation Conference*, pp. 523–539, 2024.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[19] Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968, 2023.

[20] Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 837–848, 2022.

[21] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

[22] Pasi Shailendra, Rudra Chandra Ghosh, Rajdeep Kumar, and Nitin Sharma. Survey of large language models for answering questions across various fields. In *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Vol. 1, pp. 520–527, 2024.

[23] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, Vol. 30, pp. 6833–6844, 2017.

[24] Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2195–2222, 2022.

[25] Jonathan Tonglet, Manon Reusens, Philipp Borchert, and Bart Baesens. SEER : A knapsack approach to exemplar selection for in-context HybridQA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13569–13583, 2023.

[26] Martina Toshevska and Sonja Gievska. A review of text style transfer using deep learning. *IEEE Transactions on Artificial Intelligence*, Vol. 3, No. 5, pp. 669–684, 2022.

[27] Martina Toshevska and Sonja Gievska. LLM-based text style transfer: Have we taken a step forward? *IEEE Access*, Vol. 13, pp. 44707–44721, 2025.

[28] Xingchen Wan, Ruoxi Sun, Hootan Nakhost, and Sercan Ö. Arık. Teach better or show smarter? on instructions and exemplars in automatic prompt optimization. In *Advances in Neural Information Processing Systems*, Vol. 37, pp. 58174–58244, 2024.

[29] Jianyu Wang, Zhiqiang Hu, and Lidong Bing. Evolving prompts in-context: An open-ended, self-replicating perspective. In *Forty-second International Conference on Machine Learning*, 2025.

[30] Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Prompt optimization with EASE? efficient ordering-aware automated selection of exemplars. In *Advances in Neural Information Processing Systems*, Vol. 37, pp. 122706–122740, 2024.

[31] Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. Prompt engineering a prompt engineer. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 355–385, 2024.

[32] Haopeng Zhang, Philip S. Yu, and Jiawei Zhang. A systematic survey of text summarization: From statistical methods to large language models. *ACM Computing Surveys*, Vol. 57, No. 11, 2025.

[33] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020.

[34] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023.

個人情報を活用する小規模言語モデルによる実装の検討

川村　碧葵† 丸　千尋†,†† 中野美由紀†††,†††† 小口　正人†

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2–1–1
†† 中央大学　〒 112–8551 東京都文京区春日 1–13–27
††† 津田塾大学　〒 187–8577 東京都小平市津田町 2–1–1
†††† 情報・システム研究機構　〒 105–0001 東京都港区虎ノ門 4–13–13 ヒューリック神谷町ビル 2 階
E-mail: †{tamaki,maru.chihiro,oguchi}@ogl.is.ocha.ac.jp, ††g1120536@is.ocha.ac.jp, †††miyuki@tsuda.ac.jp

あらまし　近年，生成 AI，特に大規模言語モデル（LLM）を用いた情報検索，要約，機械翻訳などの技術が大きく進展している。また，IT 技術の進展により小型のセンサーデバイスが様々な場所で利用されており，スマートフォンの GPS 記録からヘルスケアに関する体温，脈拍，あるいは，医療デバイスによる心電図情報の収集など，個人情報を含むデータ利用が急速に進んでいる。現在の LLM の利用ではクラウド上でサービスが提供されることが一般的であるが，上記のような個人情報を含むデータの扱いについては，漏洩リスクのある外部サーバへ送信せずに端末内で安全な処理をすることが求められている。つまり，エッジデバイスの性能が向上しつつある状況において，デバイス上のデータを用いた分野特化型の小型生成 AI モデル（SLM:Small Language Model) が期待されている。つまり，エッジデバイス上の計算リソースには限界があり，大規模言語モデルを動作させることは難しいため，用途に適応した SLM の開発が求められている。本稿では，デバイスで動作可能な小規模言語モデルの構築を目指し，まずは，分野特化のための手法として，言語モデルの中に特定分野の情報も学習するファインチューニングと，特定分野の情報をプロンプティング時にその一部をモデルに渡すことで出力精度を向上させる RAG の二つについて検討を行う。まずは，小さな特殊データに対する精度向上について実験を行い，エッジデバイスで特定のデータを活用する手法について考察を行った。

キーワード　SLM，RAG，ファインチューニング

1 はじめに

近年，エッジデバイスが広く普及し，個人利用が進んでいる。例えばヘルス・スポーツ分野等で個人の体調や記録データが日常的に記録・蓄積されている。これらの個人情報が含まれるデータは漏洩のリスクがある外部サーバに送信するのではなく，端末内で処理することが求められている。情報漏洩対策やデータ主権の観点から，海外のストレージサービスを利用せずに国内でデータ保存をする動きが進んでいる。

また，IT 技術の発展によりエッジデバイスの性能が向上し，AI モデルの利用が可能となっており，大規模言語モデルを利用することが期待されている。つまり，エッジデバイス上の計算リソースには限界があり，大規模言語モデルを動作させることは難しいため，用途に適応した SLM の開発が求められている。本稿では，デバイスで動作可能な小規模言語モデルの構築を目指し，まずは，分野特化のための手法として，言語モデルの中に特定分野の情報も学習するファインチューニングと，特定分野の情報をプロンプティング時にその一部をモデルに渡すことで出力精度を向上させる RAG の二つについて検討を行う。まずは，小さな特定データに対する精度向上について実験を行い，エッジデバイスで特定のデータを活用する手法について考察を行った。

2 関連研究

2.1 機密情報を扱う大規模言語モデルの課題と関連研究

近年，大規模言語モデル（LLM）は文書要約，質問応答，文章生成など多様なタスクにおいて高い性能を示している。しかしながら，ChatGPT などのサービス運用においては，膨大なデータを大規模計算機資源を使って学習を行い，モデルの運用においても高性能な計算機環境が要求される。このような汎用的な LLM サービスに対し，特定の分野，ある限定された機能に特化した小規模言語モデルを構築することで，学習コストも運用コストも低減することが可能でありながら，フロントエンド側で動作することでリアルタイム性に優れ，特定の用途における回答精度は十分に担保できるという提案がなされている [3]，[2]。小規模言語モデルとして，Phi-3 （Microsoft），OpenELM（Apple），Llama3（Meta），Gemma2（Google）等が提供されているが，いずれもエッジデバイス上で動作するには大きなモデルと考えられる。

また，ChatGPT 等の商用クラウド型 LLM を利用する場合，入力データを外部サーバへ送信する必要があり，機密情報や個人情報を含む組織内文書の活用には情報漏洩のリスクが伴う。このため，機密文書を安全に扱うためのローカル環境で動作する LLM の構築手法が重要な課題となっている。柳原ら [1] は，組織内機密文書の利用を前提として，Local な利用を前提とし

た大規模言語モデルの構築を目指している。汎用の大規模言語モデルに機密文書の学習を行う手法として，ファインチューニング（LoRA）および RAG に用い，その精度の評価を行っている。

3 特定データの学習手法の検討

大規模言語モデルにおける特定データの学習手法として，ファインチューニングが用いられる。また，個人情報を含むデータがセンサーデバイス上での取得となることを考慮すると，データは大規模言語モデルで学習するような大規模パラメタにはならないと想定される。今回の実験では，パラメタ数を抑えられる LoRA によるファインチューニングとモデルの外で関連文書群を提供する RAG の二つの方法を検討する。以下，二つの手法の特徴を述べる。

3.1 ファインチューニング

ファインチューニングは，大規模言語モデルに特定分野の情報をさらに学習させることによって，モデルを構築する。従って，モデルを学習させるための環境が必要となり，デバイス上などで動作させる場合には外部からのモデル更新等が必要になると考えられる。一方で，特定分野の専門知識をモデル内に取り込むことでより専門的な問合せへの精度向上が期待できる。

3.2 RAG による文章知識の活用

RAG（Retrieval-Augmented Generation）は，外部の知識源から関連文書を検索し，その情報を基に LLM が応答を生成する手法である。LLM 単体では対応が難しい最新情報や非公開文書に基づく質問応答を可能とし，LLM が事実に基づかない情報を生成するハルシネーションの抑制にも効果がある。ローカル環境において RAG を構築することで，機密文書を外部に送信することなく，安全に情報検索・質問応答を実現できる。一方で，チャンク分割方法や検索精度に応答品質が大きく依存するため，ベクトルデータベース設計や検索戦略の最適化が重要な課題となる。

4 ローカルデータに対するファインチューニングと RAG の評価実験

4.1 使用したデータ

鈴鹿工業高等専門学校令和 6 年度の 5 学科分 pdf 版シラバスの授業内容が記述されている 7 ページ以降を使用し，合計 614 科目の情報を用いた。シラバスの情報と担当教員名の CSV ファイルに整形した。

データ項目を絞っていないシラバスデータの項目は,「学校名開校年度，授業科目，科目番号，科目区分，授業形態，単位の種別と単位数，開設学科，対象学年，開設期，週時間数，教科書/教材，担当教員，到達目標，ルーブリック，概要，授業の数目方・方法，注意点，授業の属性・履修上の区分，授業計画，モデルコアカリキュラムの学習内容と到達目標，評価割合」である。

データ項目を絞ったシラバスデータの項目は,「授業科目，開設学科，教科書，担当教員，到達目標」である。

Accuracy(正解率) を求める為に,「OO の担当教員名は誰ですか？」という問い合わせと担当教員名のファイルを作成し，正解率を求めた。

4.2 実 験 環 境

CPU が 20 コアの Intel Xeon，GPU が NVIDIA の GeForce GTX 1080Ti である，HPC5000-xsl サーバ上で実験を行った。実験では主に CPU を利用した。

4.3 ファインチューニングの評価実験環境

4.3.1 使用したツール

a） 事前学習言語モデル

文書分類モデルの基盤として，日本語事前学習モデルである cl-tohoku/bert-base-japanese-v3 を使用した。モデルの実装および学習には HuggingFace Transformers および PyTorch を用いた。

b） 生成モデル

分類結果を基に自然言語出力を行うため，Ollama を用いてローカル環境で日本語 LLM， lucas2024/gemma-2-2b-jpn-it:q8_0 を実行した。

c） データセット構築

分類チューニングでは入力をシラバス本文とし，出力を担当教員名とする教師あり分類タスクとして定式化した。教員名は LabelEncoder により整数ラベルへ変換し，分類問題としてファインチューニングを行った。

インストラクションチューニングでは，各シラバス本文の先頭に「次のシラバスから担当教員名を答えてください。」という自然言語の指示文を付与し，指示付き入力としてモデルに与えた。教員名は LabelEncoder により整数ラベルへ変換し，分類問題としてファインチューニングを行った。

訓練・検証・テストデータを統合した全てのデータを用いて学習を行った。

4.3.2 モデル構成

本研究では，BERT の `CLS` トークン表現を用いた多クラス分類モデルを構築した。出力層には教員数に対応する線形層を配置し，クロスエントロピー損失を用いて学習を行った。

a） 学 習 設 定

モデルの学習には AdamW オプティマイザを使用した。バッチサイズは 8，エポック数は 3 とした。再現性確保のため，乱数シードを固定して学習を行った。

b） 推論及び生成

推論時には，学習済み BERT 分類モデルを用いて，入力されたシラバス文書から担当教員を推定する。得られた出力を基に，Ollama 上の日本語 LLM へ指示文を入力し，担当教員名のみを生成させた。

生成時には，出力形式を人名のみに限定するプロンプトを設計し，不要な説明文の生成を抑制した。

4.3.3 ファインチューニング手法

a） 分類チューニング

入力データをあらかじめ定義されたクラスラベルに割り当てることを目的としたファインチューニング手法。訓練中に遭遇したモデルの予測に限定されるため，データを事前に定義されたクラスを正確に分類しなければならないプロジェクトに適している。

b） インストラクションチューニング

特定の指示を使った一連のタスクで言語モデルを訓練するファインチューニング手法。自然言語のプロンプトで表されたタスクを理解して実行する能力を向上させる。モデルの柔軟性や対話の品質を向上，ユーザからの複雑な指示に基づいて様々なタスクを処理する必要があるモデルに適している。

4.4 RAG の評価実験環境

4.4.1 使用したツール

a） RAG 基盤

検索器，および生成モデルの統合には LangChain フレームワークを使用した。

b） 埋め込み表現（Embedding）

文書および検索クエリの意味表現を獲得するため，日本語事前学習モデルである `cl-tohoku/bert-base-japanese-v3` を使用した。モデルの推論処理には PyTorch を用い，トークン埋め込みの平均プーリングによって文書ベクトルを生成した。また，類似度計算の安定性を向上させるため，生成されたベクトルに対して L2 正規化を施した。

c） 生成モデル

検索結果を基に，Ollama を用いてローカル環境で日本語 LLM，lucas2024/gemma-2-2b-jpn-it:q8_0 で回答を生成した。検索で取得した文書の内容のみに基づいて質問に対し，担当教員名のみを出力するようプロンプト設計を行った。

4.4.2 システム設計

本研究で構築したシステムは，検索と生成を分離して実装した RAG 構成を採用している。処理の流れを以下に示す。

1. CSV ファイルからシラバス本文を読み込む
2. 各文書を埋め込み表現に変換する
3. 検索器により関連文書を取得する
4. 取得文書をコンテキストとして LLM に入力する
5. 担当教員名を生成し，正解データと比較して評価する

検索手法として，以下の 3 種類を実装し，比較評価を行った。

4.4.3 検 索 手 法

a） セマンティック検索

キーワードの一致だけでなく，ユーザの検索クエリの背後にあるコンテキスト上の意味と意図を理解し，その解釈に基づいた関連性の高い情報を検索する手法。セマンティック検索を実現する手法の一つにベクトル検索があり，文章や単語をベクトルに変換しベクトル間の距離を計算して，意味が近い単語や文章を検索する。

調べたい情報が汎用的な場合は関連情報が妨げになり，求めている情報とは異なる情報となる可能性がある。

b） キーワード検索

データベースの中にある全ての文字を対象としてキーワードや文字列を検索する手法。文字列と一致する内容を探しているだけで，言葉を理解している訳ではない。

c） ハイブリッド検索

複数の検索方法を組み合わせることで，検索結果の精度や関連性を向上させる手法。本稿ではベクトル検索とキーワード検索を用いた。ハイブリッド検索スコアリング (RRF) はベクトル検索とキーワード検索の結果を重みなしで統合して最終ランキングを作成した場合と、重みを与えたうえで統合して最終ランキングを作成した場合を実験した。

4.5 結　　果

データ項目を絞ったシラバスデータの項目は,「授業科目，開設学科，教科書，担当教員，到達目標」である。シラバスデータの csv ファイルサイズが 5,400KB に対して，データ項目を絞った csv のファイルサイズは 383KB である。

検索器が返す上位 k 件の文書をコンテキストとして用いた。

表 1: シラバスデータに対するファインチューニングによる担当教員推定の Accuracy

手法	Accuracy(正解数/総数)
学習をしていない結果	0.0000 (0/124)
分類チューニング シラバスデータ (5,400KB)	0.0000 (0/124)
分類チューニング 絞ったシラバスデータ (383KB)	0.0000 (0/124)
インストラクションチューニング シラバスデータ (5,400KB)	0.0323 (4/124)
インストラクションチューニング 絞ったシラバスデータ (383KB)	0.0565 (7/124)

表 2: RAG によるシラバスデータに対する検索手法毎の担当教員推定の Accuracy と実行時間

k	検索手法	Accuracy(正解数/総数)	実行時間 [s]
3	ベクトル検索	0.0161 (2/124)	298.32
	キーワード検索	0.0242 (3/124)	297.71
	ハイブリッド検索	0.0081 (1/124)	303.91
5	ベクトル検索	0.0242 (3/124)	304.71
	キーワード検索	0.0081 (1/124)	296.57
	ハイブリッド検索	0.0081 (1/124)	305.42
10	ベクトル検索	0.0323 (4/124)	309.46
	キーワード検索	0.0000 (0/124)	300.94
	ハイブリッド検索	0.0000 (0/124)	315.47
15	ベクトル検索	0.0484 (6/124)	317.66
	キーワード検索	0.0000 (0/124)	309.95
	ハイブリッド検索	0.0403 (5/124)	330.08

表 3: データ項目を絞ったシラバスデータに対する検索手法毎の担当教員推定の Accuracy と実行時間

k	検索手法	Accuracy(正解数/総数)	実行時間 [s]
3	ベクトル検索	0.2258 (28/124)	74.80
	キーワード検索	0.0323 (4/124)	37.89
	ハイブリッド検索	0.2177 (27/124)	82.77
5	ベクトル検索	0.2581 (32/124)	94.35
	キーワード検索	0.0403 (5/124)	41.45
	ハイブリッド検索	0.2984 (37/124)	113.90
10	ベクトル検索	0.3145 (39/124)	140.72
	キーワード検索	0.0565 (7/124)	42.04
	ハイブリッド検索	0.3065 (38/124)	194.68
15	ベクトル検索	0.3145 (39/124)	192.42
	キーワード検索	0.1532 (19/124)	49.03
	ハイブリッド検索	0.2016 (25/124)	280.30

表 4: ベクトル検索とキーワード検索の重みを 0.7:0.3 としたハイブリッド検索による担当教員推定の Accuracy と実行時間

k	検索手法	シラバスデータの Accuracy(正解数/総数)	データ項目を絞ったシラバスデータの Accuracy(正解数/総数)
3	ベクトル検索	0.0161 (2/124)	0.2258 (28/124)
	キーワード検索	0.0242 (3/124)	0.0323 (4/124)
	ハイブリッド検索	0.0161 (2/124)	0.2258 (28/124)
5	ベクトル検索	0.0242 (3/124)	0.2581 (32/124)
	キーワード検索	0.0081 (1/124)	0.0403 (5/124)
	ハイブリッド検索	0.0242 (3/124)	0.2581 (32/124)
10	ベクトル検索	0.0323 (4/124)	0.3145 (39/124)
	キーワード検索	0.0000 (0/124)	0.0565 (7/124)
	ハイブリッド検索	0.0323 (4/124)	0.3145 (39/124)
15	ベクトル検索	0.0484 (6/124)	0.3145 (39/124)
	キーワード検索	0.0000 (0/124)	0.1532 (19/124)
	ハイブリッド検索	0.0484 (6/124)	0.3226 (40/124)

シラバスデータに対し，ファインチューニングを行った際の手法による担当教員推定の正解率の違いを表 1 に示す。シラバスデータに対し、RAG による検索手法と検索器が返す文章の上位 k 件を採用した場合の担当教員推定の正解率と実行時間の違いを表 2 に示す。データ項目を絞ったシラバスデータに対し、RAG による検索手法と検索器が返す文章の上位 k 件を採用した場合の担当教員推定の正解率と実行時間の違いを表 3 に示す。ベクトル検索：キーワード検索=0.7:0.3 で重みをつけて、ハイブリッド検索スコアリングを定めた結果を表 4 に示す。

表 1 より，ファインチューニングのみを行った場合，担当教員推定の正解率は最大でも 0.0565 に留まり，全体として低い結果となった。特に分類チューニングでは正解率が 0.0000 となっており，本タスクにおいては単純な分類学習だけでは有効な特徴を獲得できていないことが分かる。一方，インストラクションチューニングではわずかながら精度が向上しており，自然言語形式での学習が一定の効果を持つことが考えれる。

表 2 および表 3 より，RAG を用いた場合の結果を比較すると，シラバス全体のデータでは正解率は最大 0.0484 に留まり，ファインチューニング単体と大きな差は見られなかった。一方でデータ項目を絞ったシラバスデータでは，正解率が大きく向上し，ベクトル検索では最大 0.3145，ハイブリッド検索では最大 0.3065 を達成した。また，実行時間も全体データでは約 300 秒前後であったのに対し，データ項目を絞ったシラバスデータでは最大でも約 280 秒，多くの条件で 100 秒前後となっており，大幅な短縮が確認された。このことから，担当教員推定に関係の薄い情報を削減することで，検索精度と処理効率の双方が向上したと考えられる。

検索件数 k に着目すると，データ項目を絞ったシラバスデータでは，k の増加に伴い正解率が向上する傾向が見られた。特にベクトル検索では，$k = 3$ の 0.2258 から $k = 10$ の 0.3145 まで上昇した。しかし，$k = 10$ から $k = 15$ では正解率の向上が見られず，実行時間のみが増加している。このことから，本実験条件では $k = 10$ 程度が精度と実行時間のバランスが取れた設定であると考えられる。

表 2 および表 3 より，ハイブリッド検索は必ずしもベクトル検索より高い正解率を示すとは限らず，精度が低下する場合が見られた。これは，ハイブリッド検索スコアリング（RRF）において，ベクトル検索とキーワード検索の結果を重みなしで統合しているためであると考えられる。そこで，ベクトル検索の方が多くの条件で高い正解率を示していたことから，ベクトル検索：キーワード検索＝ 0.7：0.3 の重みを設定し，ハイブリッド検索スコアリングを再定義した結果を表 4 に示す。表 4 と表 2，表 3 を比較すると，重み付けを行ったハイブリッド検索では，多くの条件で正解率が向上していることが確認できた。しかし，もともとハイブリッド検索の方が正解率が高かった条件では，重み付けによりわずかに正解率が低下する場合も見られた。

5 まとめと今後の課題

本実験では，シラバスデータを用いた担当教員推定において，ファインチューニング手法と RAG 手法の性能を比較した。その結果，シラバスデータ全体及び，データ項目を絞ったシラバスデータのいずれの場合においても，ファインチューニングより RAG を用いた手法の方が正解率が高いことを確認した。

ファインチューニングでは，分類チューニングにおいて正解率が 0.0000 となり有効な予測を行うことが出来なかった。これは学習時にはシラバス本文を与えたのに対して，推論時には「OO の担当教員名は誰ですか？」という問い合わせを利用したため，入力形式が大きく異なり，モデルが適切に対応出来なかった為であると考える。

一方，インストラクションチューニングでは僅かではあるが正解率の向上を確認できた。これは自然言語の指示形式で学習を行ったことにより，推論時の問い合わせ文との形式差が小さくなり，モデルの汎用性が向上した為であると考えられる。

RAG 手法では，検索によって関連するシラバス情報を直接

参照出来るため，ファインチューニングに比べ高い正解率を得ることが出来たと考えられる。特にデータ項目を絞ったシラバスデータは，そのままのシラバスデータを用いたサイト比較して，性能が大きく向上した。これは不要な情報を除去し，担当教員に関係する情報のみを検索対象としたことで，検索精度が向上したためであると考えられる。

また，検索時に使用する上位文書数 k の増加に伴い，正解率が向上する傾向がみられたが，同時に実行時間も増加することを確認した。このことから，RAG では精度と計算コストのトレードオフが存在すると考えられる。

ハイブリッド検索スコアリングの重みを変化させることで、ハイブリッド検索の正解率が変化することが確認した。データ特性に応じた適切な重みの設定が重要であると考えられる。

以上より，本実験のようにデータ量が限られている情報を活用する際には，モデル内部に知識を学習させるファインチューニングよりも，外部知識を検索して利用する RAG の方が有効であると考えられる。

文　　献

[1] 柳原皓之介，伊藤栄典 “LoRA による機密情報を扱う LLM 実現の試み，“ 研究報告データベースとデータサイエンス. 2025-DBS-181 (10)， pp.1-6，情処， 2025-09-09.

[2] Peter Belcak, Greg Heinrich, et al., "Small Language Models are the Future of Agentic AI, " https://arxiv.org/pdf/2506.02153, 2025.9

[3] Zhenyan Lu, Xiang Li, et al., "SMALL LANGUAGE MODELS:SURVEY, MEASUREMENTS, AND INSIGHTS" https://arxiv.org/pdf/2409.15790, 2025.2

[4] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" https://arxiv.org/pdf/2005.11401, 2020

[5] Hyung Won Chung et al., "Scaling Instruction-Finetuned Language Models" Journal of Machine Learning Research (JMLR), https://arxiv.org/abs/2210.11416, 2024

[6] Hiroki Watanabe, Motonobu Uchikoshi "Generating Privacy-Preserving Personalized Advice with Zero-Knowledge Proofs and LLMs" https://arxiv.org/abs/2502.06425, 2025-4

[7] S.Xu, W.Xie, L.Zhao, and P.He,“Chain of Draft: Thinking Faster by Writing Less,”https://arxiv.org/pdf/2502.18600, 2025.

[8] Noveen Sachdeva, Benjamin Coleman, et al., "How to Train Data-Efficient LLMs" https://arxiv.org/abs/2402.09668, 2024

[9] "Curriculum Reinforcement Learning from Easy to Hard Tasks Improves LLM Reasoning" https://arxiv.org/abs/2506.06632, 2025

[10] Yun Zhu, Jia-Chen Gu, Caitlin Sikora, et al., "ACCELERATING INFERENCE OF RETRIEVAL- AUGMENTED GENERATION VIA SPARSE CONTEXT SELECTION" https://arxiv.org/abs/2405.16178, 2024

大規模言語モデルを用いたVADER感情分析手法の構造的拡張

劉　浩東†　酒井　哲也†

† 早稲田大学基幹理工研究科情報理工・情報通信専攻　〒 169–8555 東京都新宿区大久保 3 丁目 4 番 1 号
E-mail: †lhd1565703257@akane.waseda.jp, ††tetsuyasakai@acm.org

あらまし　本研究は，商品レビューにおける感情評価の定量化を目的とし，ルールベース感情分析手法である VADER（Valence Aware Dictionary and sEntiment Reasoner）の拡張フレームワークを提案する．VADER は高い計算効率と可解釈性を有する一方，ルール適用における固定長の参照範囲に依存するため，修飾語や否定表現などの文脈依存関係を正確に捉えられない場合がある．本研究では，大規模言語モデル（Large Language Model）を感情スコアの算出器としてではなく，感情語と修飾表現の関係構造を判定する補助的判断器として用い，その判定結果を VADER の既存ルールの適用条件に反映する手法を提案する．実験的検討の結果，複数の商品カテゴリおよび異なる評価分布条件において，誤差指標および相関指標の双方で一貫した性能向上が確認された．

キーワード　感情分析，大規模言語モデル

1　はじめに

近年，電子商取引プラットフォームや SNS の普及に伴い，商品レビューやユーザ投稿といった自然言語テキストは，消費者の意見や感情を反映する重要な情報源となっている．これらのテキストデータは，広告配信，商品推薦，顧客満足度分析など，多様な実務的応用において活用されており，レビュー内容に含まれる感情的傾向を適切に把握することは，意思決定支援の基盤として重要である．特に商品レビューには，ユーザが自由記述によって表現する主観的な感情や評価理由が含まれており，数値評価のみからは把握できない情報が豊富に存在する．

商品レビューの感情分析においては，テキストが肯定的か否定的かを分類する定性的分析が広く用いられてきた．しかし，実務的な分析やシステム応用を考えると，「ポジティブ」「ネガティブ」「ニュートラル」といった離散的なラベルのみでは十分でない場合が多い．例えば，複数の商品を比較する場面や，レビュー全体の傾向を集約する場面では，感情の強さや度合いを連続的な尺度として扱えることが望ましい．また，広告推薦やランキング生成においては，感情評価を用いた重み付けや順位付けが必要となるため，レビュー本文に含まれる感情表現を定量的に評価できることは有用である．このような背景から，商品レビューに対する感情分析においては，定性的分類に加え，感情の度合いを数値として表現する定量的分析が重要な役割を果たす．

短文テキストの定量的感情分析手法として，VADER は現在も広く利用されている代表的手法の一つである [1]．VADER は，ソーシャルメディアや商品レビューなどの短文テキストを対象として設計されたルールベース感情分析手法であり，高い計算効率と可解釈性を有することから，実務的分析において現在も用いられている．VADER は，各語の感情極性を表す語彙辞書と，修飾語，否定表現，強調表現などを考慮する一連のルールに基づいて感情スコアを算出するため，モデルの振る舞いを人手で理解・検証しやすいという利点を持つ．

一方で，VADER は典型的なルールベース手法であり，その振る舞いは事前に人手で設計された規則集合によって完全に規定されている．ルールベース手法では，語彙の極性や修飾規則，否定規則といった処理が明示的に定義されているため，可解釈性や安定性に優れる反面，ルールの適用条件や作用範囲は固定的であり，入力文の文脈構造に応じて動的に調整されることはない．VADER においても，固定的な語彙辞書および固定長の文脈 window に基づいて規則が適用される設計となっており，語の出現位置や距離といった局所的な情報に依存して修飾関係や否定関係が判断される．このような設計は，短文テキストを高速かつ一貫した基準で処理できるという利点を持つ一方で，文中に複数の修飾語や節構造が存在する場合には，文脈的な関係を十分に反映できない場合がある．例えば，修飾語が意図しない感情語に付着する誤判定や，否定表現の作用範囲が曖昧になるといった構造的な問題が生じることがある．これらの問題は，特定のルール実装の不備というよりも，固定的な規則に基づいて処理を行うルールベース手法に共通する構造的制約に起因するものであり，感情評価を定量的に行う際の安定性や妥当性に影響を及ぼす可能性がある．

近年，大規模言語モデル（LLM）は，高度な文脈理解能力を示し，さまざまな自然言語処理タスクにおいて有効性が報告されている．感情分析においても，LLM を用いた end-to-end の分類手法が提案されているが，これらの手法はモデル内部の判断過程が不透明である点や，計算コストの高さ，出力の安定性といった観点から，既存のルールベース手法をそのまま置き換えることには課題が残る．特に，実務利用において重視される可解釈性や軽量性を維持したまま感情評価の精度を向上させるためには，LLM を単純なスコア算出器として用いるのではなく，その能力を限定的かつ補助的に活用する設計が求められる．

そこで本研究では，LLM を感情極性や数値スコアを直接決定するモデルとしてではなく，文中の感情語と修飾表現との関

係構造を判定する補助的な判断器として位置付ける．具体的には，LLM により推定された修飾関係や否定範囲といった構造情報を，VADER の既存ルールの適用条件に統合することで，数値計算規則および可解釈性を保持したまま，文脈に基づくルール適用を可能にする拡張フレームワークを提案する．本論文では，提案フレームワークが従来手法と比較して予測精度および順位相関の観点からどの程度改善に寄与しうるかを，定量的評価を通じて明らかにする．

2 関 連 研 究

商品レビューに含まれる感情情報の分析は，ユーザ意見の理解，推薦システム，広告分析など多様な応用において重要な研究課題である．本研究では，特にレビュー文における感情をどのように分析するかかを，既存研究を代表的なアプローチごとに整理する．

2.1 語彙・ルールベースによる感情定量化

短文テキストに対する感情分析手法として，語彙辞書と手作業で設計されたルールに基づくアプローチは現在も広く利用されている．その代表例が，Hutto らにより提案された VADER である [1]．VADER は，感情語彙に付与された極性および強度情報を基礎とし，否定表現，強調表現，句読点，大文字表記，接続詞による転換などを考慮する一連のルールを適用することで，テキスト全体の感情を感情価（valence）として算出する手法である．VADER は学習を必要とせず，計算効率が高いことに加え，感情スコアの算出過程が明示的であるという高い可解釈性を有する．そのため，商品レビューや SNS 投稿などの短文テキスト分析において，ベースライン手法として現在も広く採用されている．一方で，語彙およびルールが静的に定義されているため，文脈に依存した修辞的表現や複雑な修飾関係を十分に捉えられない場合があることも指摘されている．

2.2 構造化によるレビュー理解の拡張

全体的な感情極性のみではレビュー内容を十分に捉えられないという観点から，レビュー文をより構造化して分析する研究が数多く提案されている．

アスペクトベースの感情分析（Aspect-Based Sentiment Analysis; ABSA）では，レビュー文から製品やサービスの属性（aspect）を抽出し，各 aspect に対する感情を推定することで，評価対象を明確化することを目的としている [2]．

また，推薦システム分野においては，レビュー文から抽出された感情情報や情緒的特徴をユーザ・アイテムの特徴量として利用し，推薦性能の向上を図る研究が報告されている [3]．

これらの研究は，レビュー理解の粒度を高める点で有用である一方，感情表現を多くの場合，離散的なカテゴリとして扱っており，文脈に依存した感情強度を連続的に定量化する枠組みは必ずしも対象としていない．

2.3 大規模言語モデルを用いたレビュー文脈理解

近年では，Transformer 系モデルや大規模言語モデル（LLM）を用いたレビュー理解が注目されている．LLM は高い言語理解能力を有しており，レビュー文に含まれる文脈や修辞的表現を柔軟に解釈できる点が特徴である．例えば，Zuhir らの研究 [4] では，LLM を用いてレビュー文から重要な側面や意見を抽出し，ユーザ評価値とテキスト感情の乖離を分析する枠組みが提案されている．しかし，これらのアプローチでは，LLM が直接的に解釈結果や要約を生成することが多く，感情を明示的かつ再利用可能な数値として定義・算出する枠組みは必ずしも提供されていない．また，モデル内部の判断過程が不透明であるため，語彙・ルールベース手法が有する可解釈性との整合は課題として残されている．

3 提 案 手 法

3.1 VADER における感情スコアリング機構と制限

VADER は，語彙辞書と手続き的ルールに基づいて短文テキストの感情強度を定量化するルールベース感情分析手法である．特に，ソーシャルメディアや商品レビューといった非定型・短文テキストを対象とし，高い計算効率と可解釈性を重視した設計がなされている．

VADER の処理は図 1 に示すように，入力テキストをトークン列として逐次処理することから始まる．分かち書きは主として空白に基づいて行われ，単語に加えて顔文字や絵文字，感嘆符などの記号も保持される．また，don’t のような縮約形や，!!! といった強調表現が後続のルールで利用できるよう，過度な正規化は行われない．このような設計は，感情表現が語彙および簡単な記号に集約されやすいというソーシャルメディア言語の特性を反映している．

次に，各トークンに対して感情語彙辞書の照合が行われる．VADER の語彙辞書は，クラウドソーシングにより人手評価された感情語彙から構成されており，各語には $[-4, +4]$ の範囲で定義された連続値の感情価（valence）が付与されている．辞書に含まれないトークンは中立（valence $= 0$）として扱われ，語彙辞書に基づく感情価が後続の文脈補正の基礎スコアとなる．

例えば，文“I absolutely love this phone”は以下のようにトークン列と初期感情価列として表現される：tokens $=$ [I, absolutely, love, this, phone]，valence $=$ [0, 0, +v, 0, 0]

感情語が検出されると，その語を基準として文脈的修飾を考慮するためのルールが適用される．VADER では，感情語の直前に出現する最大 3 トークンを参照範囲（window）とし，この範囲内に含まれる語に基づいて修正を行う．まず，強調・減衰を表す程度副詞（booster / dampener）が検出された場合，対応するスカラー値により感情価が増幅または減衰される．booster の影響は感情語との距離に応じて減衰するよう設計されており，距離が近い修飾語ほど強く作用する．

否定表現についても，同様に感情語前方の window 内で検出される．否定語が存在する場合，感情語の感情価は固定の縮放係数により反転または減衰される．一方で，“never so X” や “without doubt” のように，単純な否定として扱うと誤判定を招く高頻度表現については，例外規則が明示的に定義されてい

る．これにより，否定規則による過剰な反転を回避している．

これらの文脈補正ルールは，感情語ごとに定められた順序で逐次的に適用される．すなわち，語彙辞書による基礎感情価に対して，booster / dampener，negation，idiom や special case の判定が同一 window 内で交錯的に実行される構造となっている．この順序性は，自然言語における修飾と否定の重なりを近似的に再現することを意図している．

さらに，文全体の構造を考慮するため，対比接続詞 “but” が検出された場合には特別な処理が行われる．具体的には，but より前の部分に含まれる感情価を減衰させ，but 以降の部分を相対的に強調することで，英語における評価の重心が後半に置かれやすいという談話的特性を反映している．

また，表記上の強調も感情価に反映される．感情語が全て大文字で記述されている場合や，感嘆符・疑問符が連続して用いられている場合には，感情の極性を変えずに強度のみを増幅する．ただし，過剰な影響を避けるため，標点による増幅効果には上限が設けられている．

以上の処理により得られた各感情語の感情価は集約され，文全体の感情強度を表す指標として compound score が算出される．compound score は，感情語ごとの感情価の総和を，その二乗和に平滑項を加えた値の平方根で除算する非線形正規化により定義され，$[-1, +1]$ の範囲に収まる連続値として出力される．この正規化により，文長や感情語数の違いによる影響が抑制され，異なる文間での感情強度の比較が可能となる．

このように VADER は，語順，距離，例外規則といった人手設計のヒューリスティクスを組み合わせることで，高速かつ可解釈な感情スコアリングを実現している．

しかしその一方で，感情修正の判断は，事前に定義された固定的なルール集合に強く依存している．

VADER では，修飾語や否定表現，転換表現に対して，多くの特例が個別に設計されているが，自然言語における表現の多様性を網羅的に列挙することは本質的に困難である．そのため，既存ルールに含まれない表現や，複数のルールが同時に関与する状況においては，誤った感情修正が生じる可能性がある．

特に，否定表現に関しては，二重否定や否定と修飾語の組み合わせなど，ルール間の相互作用が複雑化する場合，固定長 window に基づく判定では十分に対応できない．同様に，転換表現についても，VADER は主に “but” のみを対象としており，however などの他の転換表現や，although など文頭に現れる転換構造に対する作用範囲の違いは考慮されていない．

これらの問題は，VADER の数値スコアリング規則そのものではなく，ルールが適用される対象語および作用関係を，語彙と語順のみに基づいて決定している点に起因する．

3.2 LLM-assisted VADER

前節で述べたように，VADER は語彙辞書と明示的なルールに基づく設計により，高い可解釈性と計算効率を実現している一方で，修飾語や否定表現，転換表現の作用対象および作用範囲を，語順と固定長 window のみに基づいて決定している点に構造的な制約を有する．本研究では，これらの制約を緩和するため，VADER の数値スコアリング規則および処理順序を変更することなく，文脈に基づく関係判定を導入する手法を提案する．

Algorithm 1: Original VADER Sentiment Scoring

Require: input text T
Ensure: compound score $s \in [-1, 1]$

$(w_1, \ldots, w_n) \leftarrow \texttt{tokenize}(T)$
$V \leftarrow \emptyset$
for $i = 1$ **to** n **do**
  if w_i in sentiment lexicon **then**
    $v \leftarrow \texttt{lexicon\_valence}(w_i)$
    for $k = 1$ **to** 3 **do**
      if w_{i-k} is booster/dampener **then**
        $v \leftarrow \texttt{apply\_booster\_vader}(v, i-k, i)$
      if w_{i-k} is negation **then**
        $v \leftarrow \texttt{apply\_negation\_vader}(v, i-k, i)$
    end for
    $v \leftarrow \texttt{apply\_caps\_emphasis\_vader}(v, w_i)$
    $v \leftarrow \texttt{apply\_idiom\_override\_vader}(v, w_{1:n}, i)$
    $V \leftarrow V \cup \{v\}$
  end if
end for
$V \leftarrow \texttt{apply\_contrast\_vader}(V, w_{1:n})$
$V \leftarrow \texttt{apply\_punctuation\_emphasis\_vader}(V, w_{1:n})$
$s \leftarrow \dfrac{\sum V}{\sqrt{\sum V^2 + 15}}$

図 1 Procedure of original VADER sentiment scoring [1]

提案手法の基本的な考え方は，VADER のルールやスコア計算を置き換えるのではなく，各ルールが適用されるか否かの判断を補助する機構として大規模言語モデル（LLM）を利用する点にある．すなわち，LLM は感情極性やスコアを直接推定するのではなく，感情語と修飾語との意味的関係を構造的に判定する役割のみを担う．

具体的には，図 1 に示す．入力文をトークン列として処理し，語彙辞書に基づいて各トークンに初期感情価を付与した上で，その情報を LLM に入力する．LLM は，文全体の文脈を考慮し，感情語（sentiment targets）の位置，booster や negation として作用する語の対応関係，および転換表現（contrast）の作用範囲を推定し，構造化された形式で出力する．

提案手法では，VADER の window 構造およびルール適用順序を維持したまま，修飾関係の判定のみを LLM に委ねる．LLM によって推定された感情語と修飾語の関係に基づき，booster や negation などの各ルールがどの感情語に対して適用されるかを制御する．数値的な調整方法，強調表現（大文字表記や句読点），およびスコアの正規化処理については，原版 VADER の定義をそのまま用いる．これにより，VADER の可解釈性と既存実装との互換性を保ったまま，修飾関係の誤付与や作用範囲の誤判定を抑制することを目的とする．

提案する LLM-assisted VADER の処理手順を示す．まず，入力文をトークン化し，語彙辞書に基づいて各トークンに初

期感情価を割り当てる．次に，トークン列と初期感情価情報を LLM に入力し，感情語の位置および修飾関係を含む構造的な判定結果を取得する．

LLM は，感情語（target）と修飾語（booster，negation，contrast など）との意味的関係を判定し，その結果を構造化された形式で出力する．例文“I absolutely love this phone”例文に対する出力の一例を以下に示す：

```
targets={3}, booster={3:[2]}
negation={}, contrast={}
```

この出力は，感情語 love（位置 3）が，直前の修飾語 absolutely（位置 2）によって強調されていることを示している．

その後，VADER のスコアリング処理において，LLM によって指定された感情語のみを対象としてループ処理を行い，対応する修飾語が存在する場合に限って，booster や negation による調整を適用する．強調表現については原版 VADER のルールを適用し，最後に，LLM によって判定された転換表現の作用範囲に基づいて contrast 処理を行い，文全体の compound score を算出する．

以上のように，本手法は，VADER の数値的なスコアリング機構を保持したまま，ルール適用の判断部分に文脈情報を導入することで，ルールベース感情分析手法の柔軟性を向上させることを目指している．

ここでは，原版 VADER と提案手法の挙動の違いを，代表的な文例を用いて定性的に比較する．表 1 は，修飾関係や転換表現の扱いに起因する感情評価の差異を示した例である．

表 1 に示すように，原版 VADER では，固定長 window と語彙ベースのルールに基づく判定により，修飾語や転換表現が意図しない感情語に付与される場合がある．例えば，“extremely surprisingly good” の例では，booster が本来修飾すべき感情語ではなく，非感情語に誤って適用されている．

また，“hardly” や “scarcely” など NEGATE 辞書に含まれない否定表現，“moderately” など BOOSTER 辞書に含まれない程度修飾表現，および “although” や “however” といった “but” 以外の転換表現を含む文においては，原版 VADER のルールではこれらの表現が十分に考慮されず，感情スコアに反映されない要素が生じることが確認される．

一方，提案手法では，LLM による文脈的な関係判定に基づいて修飾語および転換表現の作用対象を限定することで，これらの誤付与を抑制し，より妥当な感情強度の推定が可能となる．

Algorithm 2: LLM-Assist VADER Scoring

Require: input text T, LLM output (targets/relations)
Ensure: compound score $s \in [-1, 1]$

$(w_1, \ldots, w_n) \leftarrow$ `tokenize`(T)
for $i = 1$ **to** n **do**
  if w_i in lexicon **then** $b_i \leftarrow$ `lexicon_valence`(w_i)
  else $b_i \leftarrow 0$
end for
`llm_tags` $\leftarrow$ `LLM`$(w_{1:n}, b_{1:n})$
`llm_index` $\leftarrow$ `build_index`(`llm_tags`)
$V \leftarrow \emptyset$
for each sentiment target position $i \in$ `llm_index.targets` **do**
  $v \leftarrow b_i$
  for $k = 1$ **to** 3 **do**
    $j \leftarrow i - k$
    if $j < 1$ **then break**
    if $j \in$ `llm_index.booster`$[i]$ **then**
      $v \leftarrow$ `apply_booster_llm`(v, j, i)
    if $j \in$ `llm_index.negation`$[i]$ **then**
      $v \leftarrow$ `apply_negation_llm`(v, j, i)
  end for
  $v \leftarrow$ `apply_caps_emphasis_vader`(v, w_i)
  $V \leftarrow V \cup \{v\}$
end for
$V \leftarrow$ `apply_contrast_llm`$(V,$ `llm_index.contrast_up/down`$)$
$V \leftarrow$ `apply_punctuation_emphasis_vader`$(V, w_{1:n})$
$s \leftarrow \dfrac{\sum V}{\sqrt{\sum V^2 + 15}}$

図 2 Procedure of the proposed LLM-assisted VADER sentiment scoring

4 実　　験

4.1 実験目的

本実験の目的は，提案手法である LLM-assisted VADER が，原版 VADER と比較して，レビュー文に付与されたユーザ評価（星評価）とより整合的な感情強度スコアを算出できるかを検証することである．

商品レビューにおいては，テキスト中に表現される感情的トーンと，最終的に付与される星評価が必ずしも一致するとは限らないことが，既存研究により指摘されている [5]．星評価は，個人の評価基準や文脈に依存する主観的指標であり，厳密な意味での感情ラベルの gold standard とは言えない．

しかしながら，多くの場合，星評価はレビュー執筆者の総合的な感情的判断を反映した結果として与えられており，テキストに表出した感情と一定の関連性を有する指標であると考えられる．そのため本研究では，星評価を感情強度の近似的な参照値として位置づけ，原版 VADER と提案手法によって算出された感情スコアが，どちらの方がユーザ評価により整合的であるかを比較する目的で用いる．

具体的には，両手法による感情スコアを同一の評価尺度に写像した上で，星評価との誤差（RMSE：Root Mean Squared Error，MAE：Mean Absolute Error）を算出し，誤差の小ささをもって感情スコアリング結果の相対的な妥当性を評価する．

4.2 データセット

本研究では，Hou ら [6] により構築・公開された Amazon Reviews 2023 データセットを用いて評価を行う．各レビューは，自然言語による本文テキストと，対応する星評価（1〜5）から構成されている．

本研究では，以下の条件を満たすレビューを抽出した：

表 1 Representative examples comparing original VADER and the proposed method

Text	Original VADER	Proposed Method	Explanation
This is extremely surprisingly good.	Over-amplifies “surprisingly”	Correctly amplifies “good”	Booster is incorrectly attached to a non-target in VADER due to window-based rules.
The movie was hardly interesting.	Positive	Negative	Because “hardly” is not included in the negation lexicon, its negating effect is not detected.
The performance is relatively stable.	Positive	Moderately positive	Because “relatively” is not included in the booster lexicon, its weakening effect is not captured.
Although the design is good, the performance is bad.	Neutral	Negative	Sentence-initial contrast (e.g., “although”) is not considered in VADER.
The battery is small; however, I absolutely love this phone	Positive	Strongly positive	Contrastive discourse markers other than “but” are ignored by VADER.

- 英語で記述されたレビュー文
- 星評価が明示的に付与されているもの

本データセットは，高評価（4〜5 星）のレビューが多数を占める評価分布の偏りを有していることが知られている．そのため本研究では，評価分布の影響を考慮するため，以下の二つの設定に基づいて評価実験を実施する：

- 自然分布設定（unbalanced setting）：元データの星評価分布を保持したまま，複数ドメインからレビューを無作為抽出した評価集合．
- 分布制御設定（balanced setting）：各星評価（1〜5）が同数となるようにレビューを抽出した評価集合．

これにより，提案手法による性能改善が，特定の評価分布に依存した見かけ上の効果ではなく，より一般的な傾向として成立しているかを検証する．

4.3 モデルの設定

提案手法では，VADER の数値スコアリング規則自体は変更せず，booster ／ negation ／ contrast の適用可否のみを判定するゲートとして大規模言語モデル（LLM）を用いる．

本研究における LLM 呼び出し条件は以下の通りであり，全ての実験（unbalanced / balanced の両設定）を通じて固定した：

- モデル：`gpt-5-mini`
- 推論設定：`reasoning_effort = low`
- 出力冗長度：`verbosity = low`

また，LLM は感情スコアを直接出力せず，トークン列上の関係（修飾・否定・転換）の有無を構造化形式で返し，VADER の既存規則が適用可能かどうかのみを決定する．これにより，提案手法は原版 VADER との可比性を維持する設計となっている．

4.4 評価指標

提案手法の有効性を評価するため，感情スコアと星評価との整合性を以下の指標により測定する．

a） 相関指標

- Pearson 相関係数：感情スコアと星評価との線形相関を評価する．
- Spearman 順位相関係数：順位関係の一致度を評価する．

b） 誤差指標

- MAE（Mean Absolute Error）
- RMSE（Root Mean Squared Error）

各手法により算出された compound score は，線形変換によって 1〜5 の評価尺度へ写像した上で，実際の星評価との誤差を算出する．

4.5 実験手順

評価実験は，原版 VADER と提案手法を同一のレビュー集合に対して適用するペア比較（paired comparison）の形式で実施する．実験手順は以下の通りである．

1. 各レビュー文に対し，原版 VADER を適用し compound score を算出する．
2. 同一のレビュー文に対し，提案手法（LLM-assisted VADER）を適用する．
3. 両手法で得られた感情スコアを 1〜5 の評価尺度へ変換する．
4. 実際の星評価との間で，相関指標および誤差指標を算出する．

さらに，両手法の差が統計的に有意なものかを検証するため，同一レビューに対する原版 VADER と提案手法の予測結果を対応付けた形で再標本化を行う Paired Bootstrap 法による信頼区間推定を行う．

ここで用いる Paired Bootstrap 法とは，各レビューについて両手法の予測結果の差を保持したままレビュー単位で再標本化を行う手法であり，同一データに対する二手法の性能差を安定的に評価するために用いられる．

5 実験結果

本章では，提案手法である LLM-assisted VADER の実験結果について報告する．まず，unbalanced データを用いて商品カテゴリ別の結果を示し，提案手法が特定のカテゴリに依存しない傾向を持つことを確認する．次に，unbalanced データおよび balanced データにおける全体的な評価結果を示す．最後

表 2 Unbalanced データにおける商品カテゴリ別誤差

	MAE			RMSE		
Category	Base	LLM	Δ	Base	LLM	Δ
Baby Products	0.693	0.673	-0.020	0.984	0.955	-0.030
Appliances	0.861	0.852	-0.009	1.138	1.126	-0.012
Fashion	0.637	0.625	-0.012	0.978	0.959	-0.019
Handmade	0.617	0.607	-0.010	0.882	0.865	-0.017
Home & Kitchen	0.698	0.684	-0.014	1.026	1.002	-0.024

表 3 Unbalanced データにおける商品カテゴリ別相関係数

	Pearson			Spearman		
Category	Base	LLM	Δ	Base	LLM	Δ
Baby Products	0.411	0.440	+0.030	0.213	0.230	+0.017
Appliances	0.500	0.520	+0.020	0.365	0.381	+0.015
Fashion	0.567	0.587	+0.020	0.409	0.431	+0.023
Handmade	0.539	0.566	+0.027	0.332	0.329	-0.003
Home & Kitchen	0.540	0.563	+0.022	0.393	0.417	+0.023

に，評価値別の誤差分析を行い，改善が生じる領域について考察するための基礎的な分析を提示する．

5.1 Unbalanced データにおける商品カテゴリ別結果

まず，unbalanced データを用いて，商品カテゴリごとの結果を示す．本分析の目的は，提案手法が特定のカテゴリにおいてのみ有効となるのではなく，異なる商品ドメインにおいて一貫した傾向を示すかを確認することである．

表 2 に，各商品カテゴリにおける誤差指標（MAE および RMSE）の結果を示す．いずれのカテゴリにおいても，LLM-assisted VADER は原版 VADER と比較して，予測誤差の低下という共通した傾向を示している．

また, 表 3 に, 対応する相関係数（Pearson および Spearman）の結果を示す．改善の程度にはカテゴリ間で差が見られるものの，多くのカテゴリにおいて，相関係数の増加が確認された．一方で，特定のカテゴリにおいて体系的な相関低下は観測されなかった．

これらの結果は，提案手法が特定のカテゴリに依存することなく，異なる商品ドメインに対して安定した挙動を示していることを示唆している．

5.2 Unbalanced データにおける全体結果

次に，ユーザ評価の自然分布を反映した unbalanced データにおける全体的な評価結果を示す．本実験は，本研究における主要な評価設定である．

本研究では，unbalanced データに含まれる全レビューを全体サンプル（ALL）と定義し，そのうち LLM がエラーなく応答を返したサンプルを評価対象として VALID とする．さらに，LLM の構造判定に基づき最終予測値が変更されたサンプルを，予測値変化サブセット（$\Delta \neq 0$）として区別する．なお，$\Delta = 0$ は LLM が適用されなかったことを意味するものではなく，LLM の判定結果が原版 VADER のスコアリング結果と一致し，最終予測値が変更されなかったケースを表す．なお, 予測値変化サブセットは VALID 全体のおよそ 40% を占めている．

表 4 全体および LLM 予測値変化サブセット（$\Delta \neq 0$）における誤差指標

	MAE			RMSE		
Subset	Base	LLM	Δ	Base	LLM	Δ
VALID (LLM response available)	0.701	0.688	-0.013	1.005	0.985	-0.020
CHANGED ($\Delta \neq 0$)	0.690	0.657	-0.033	1.050	1.000	-0.050

表 5 全体および LLM 予測値変化サブセット（$\Delta \neq 0$）における相関係数

	Pearson			Spearman		
Subset	Base	LLM	Δ	Base	LLM	Δ
VALID (LLM response available)	0.498	0.521	+0.022	0.303	0.318	+0.015
CHANGED ($\Delta \neq 0$)	0.486	0.540	+0.055	0.378	0.412	+0.034

表 4 および 表 5 に, MAE, RMSE, および相関係数を用いた評価結果を示す．提案手法である LLM-assisted VADER は，原版 VADER と比較して予測誤差を一貫して低減し，Pearson 相関および Spearman 相関の双方において改善を示した．これらの差分はいずれも paired bootstrap による検定の結果，95% 信頼区間が 0 を跨がないことが確認されており，統計的に安定した改善である．

特に，予測値変化サブセット（$\Delta \neq 0$）においては，MAE および RMSE の低減幅が VALID 全体と比較して大きく，加えて Pearson および Spearman の両相関係数がより顕著に改善されている．この結果は，LLM の構造判定が，すべてのサンプルに一様に影響するのではなく，最終予測に影響を与えるべきケースにおいてより強く機能していることを示唆している．

5.3 評価値別分析と LLM 介入挙動の特徴

本節では，5.2 節で示した unbalanced データにおける全体結果を踏まえ，ユーザ評価値（rating）に着目した分析を行う．商品レビューにおいては，評価値の分布が偏ることが多く，特定の評価帯における性能が全体指標へ与える影響は小さくない．そこで本分析では，評価値ごとの性能特性および LLM の介入挙動を明確にすることを目的とする．

5.3.1 評価値分布と分析の背景

まず，5.2 節で用いた unbalanced データにおけるユーザ評価値の分布を確認する．表 6 は，本データセットに含まれる各評価値のサンプル数およびその割合を示している．表から分かるように，本データセットでは高評価（5 ★）のレビューが全体の大部分を占めており，評価値の分布は一様ではなく，大きく偏っている．

このような評価値分布の偏りが存在する場合，特定の評価帯における性能特性が，MAE や相関係数といった全体指標に強く反映される可能性がある．そのため，unbalanced データにおける全体結果（5.2 節）を補足する目的で，次節以降では評価値ごとの性能特性および LLM の介入挙動をより詳細に分析

表 6 Unbalanced データにおける評価値の分布

Rating	Count	Ratio (%)
1	517	5.18
2	360	3.61
3	699	7.00
4	1,534	15.36
5	6,740	67.55

する．

5.3.2 評価値別サンプリング設定

前節で確認したように，unbalanced データにおいては評価値の分布が大きく偏っており，特定の評価帯（特に高評価）のサンプルが全体結果に強く影響している可能性がある．そこで本節では，評価値ごとの性能特性を公平に比較するため，評価値を均等化したサンプリング設定を用いた分析を行う．

具体的には，5 つの商品カテゴリ（Baby Products，Appliances，Fashion，Handmade，Home & Kitchen）それぞれについて，評価値 1〜5 の各評価帯から同数のレビューを抽出した．各カテゴリ・各評価値につき 200 件のレビューをランダムに選択し，合計で 5 categories $\times$ 5 ratings $\times$ 200 = 5,000 件のデータセットを構成した．

このような評価値別に均等化されたサンプリングにより，評価帯に依存しない形で，LLM-assisted VADER の性能変化および LLM の介入挙動を分析することが可能となる．

5.3.3 評価値別の性能比較結果

本節では，前節で構成した rating-balanced データセットを用いて，評価値ごとの性能比較結果を示す．以降では，平均絶対誤差（MAE）を指標として，評価値別に LLM-assisted VADER と原版 VADER の性能差を分析する．

表 7 に，unbalanced データおよび rating-balanced データの双方における，評価値別の MAE を示す．unbalanced データにおいては，評価値 1 ★，2 ★，3 ★，および 5 ★ において LLM-assisted VADER が原版 VADER と比較して一貫した MAE の低減を達成していることが確認できる．特に，評価値 1 ★ および 3 ★ においては，誤差低減の幅が相対的に大きく，提案手法の効果が顕著に現れている．

一方で，評価値分布を均等化した rating-balanced データにおいても，評価値 1 ★，2 ★，3 ★，および 5 ★ において MAE の低減が引き続き観察されており，性能向上が特定の評価値の過剰なサンプル数に依存したものではないことが確認された．ただし，rating-balanced 条件下では unbalanced データと比較して改善幅が全体的に小さくなる傾向が見られ，評価値分布の違いが全体指標に与える影響が大きいと考えられる．

また，評価値 4 ★ においては，unbalanced および rating-balanced の双方の条件下で MAE の低減が限定的，もしくはわずかに悪化する傾向が一貫して観察された．この結果は，LLM-assisted VADER の効果がすべての評価帯において一様に現れるわけではなく，評価値に依存する可能性がある．

以上の結果から，LLM-assisted VADER は極端な評価値（1 ★および 5 ★）を含む複数の評価帯において安定した誤差低減を示す一方で，中間的な評価帯では改善の程度が相対的に小さいという，評価値依存的な挙動を示すことが明らかとなった．

表 7 Unbalanced および Rating-balanced データにおける評価値別 MAE 比較

	Unbalanced			Rating-balanced		
Rating	Base	LLM	Δ	Base	LLM	Δ
1	1.856	1.779	-0.077	1.759	1.743	-0.016
2	1.435	1.415	-0.020	1.321	1.300	-0.020
3	1.172	1.139	-0.033	1.003	0.989	-0.014
4	0.732	0.733	+0.001	0.658	0.666	+0.007
5	0.517	0.509	-0.009	0.679	0.672	-0.007

5.4 結果のまとめ

LLM-assisted VADER は，商品カテゴリや評価値分布の違いに依存することなく，安定した性能改善を示すことが確認された．

まず，unbalanced データを用いた分析により，提案手法は複数の商品カテゴリにおいて一貫した誤差低減および相関向上を達成しており，特定のドメインだけに効果があるのではないことが示された．また，LLM により実際に予測値が変更されるサブセットでは改善幅がより大きく，このサブセットにおける誤差低減が，全体の性能向上に大きく寄与していることが示された．

さらに，評価値分布を均等化した rating-balanced データを用いた分析から，これらの改善が高頻度な評価帯（特に 5 ★）に依存した結果ではないことが確認された．一方で，評価値別の比較により，改善の程度は評価帯によって異なり，極端な評価値を含む領域において提案手法の効果が相対的に大きいことが明らかとなった．

以上のことから，LLM-assisted VADER が既存のルールベース感情分析の枠組みを維持しつつ，従来の VADER では捉えられなかった構造的に曖昧な文章を識別できるようになったことが，全体性能の向上につながっている可能性を示している．

6 おわりに

6.1 本研究の総括

本研究では，ルールベース感情分析手法である VADER の可解釈性を維持したまま，文脈依存の修飾関係判定に起因する構造的誤判定（修飾語の誤付着，否定範囲の曖昧さ，転換表現の作用範囲の不整合）を低減することを目的とした．そのために，大規模言語モデル（LLM）を感情スコア算出器として用いるのではなく，booster ／ negation ／ contrast の適用可否を決定する補助的判断器として利用する LLM-assisted VADER を提案した．提案手法は，VADER の数値的スコアリング規則および処理順序を変更せず，LLM により推定された関係構造に基づいてルール適用対象のみを制御する点に特徴がある．

6.2 本研究の貢献

本研究の主な貢献を以下にまとめる．

- VADER の数値スコアリング規則を保持したまま，文脈に基づく関係判定を導入する LLM-assisted 拡張フレームワークを提案した．
- LLM 出力を targets/relations の構造化形式として定義し，booster ／ negation ／ contrast の適用対象を明示的に制御可能とした．
- 複数カテゴリ（5 ドメイン）および評価分布の異なる設定（unbalanced / balanced）において，誤差指標（MAE, RMSE）と相関指標（Pearson, Spearman）により有効性を検証した．

6.3 実験結果のまとめ

実験の結果，提案手法は複数の商品カテゴリにおいて一貫した誤差低減を示し，相関係数についても多くの条件で改善が確認された．このことから，提案手法の効果が特定ドメインに依存した見かけ上の改善ではなく，より一般的に成立する可能性が示唆された．

また，全体結果の分析では，正常な LLM 応答が得られたサンプルにおいて，提案手法が原版 VADER に対して誤差指標および相関指標の双方で改善を示した．さらに，最終予測値が実際に変化したサブセット（$\Delta \neq 0$）では改善幅がより大きく，LLM による構造判定が「修正すべきケース」において選択的に機能している可能性が示された．

加えて，評価値別分析では，改善の程度が評価帯によって異なる傾向が観察され，極端な評価値を含む領域で効果が相対的に大きい一方，中間的な評価帯では改善が限定的となるケースも確認された．

6.4 限界と今後の課題

6.4.1 限　　界

本研究には以下の限界がある．

1. 本研究では星評価をテキスト感情強度の近似的参照値として用いているが，星評価は主観や文脈に依存するため，感情強度の厳密な gold standard ではない．そのため，評価指標には一定のノイズが含まれる可能性がある．
2. LLM 導入効果を明確に検証するため，本研究では VADER の数値規則および処理構造を原実装に忠実に再現した．この設計により比較可能性は確保されている一方で，固定的なルール制約を緩和することで，さらなる性能向上の余地が残されている．
3. VADER と同様に参照 window を 3 に固定しているため，感情語から離れた修飾語や否定表現を十分に捉えられない可能性がある．

6.4.2 今後の課題

本研究の結果を踏まえ，以下の課題が今後の研究として挙げられる．

1. LLM-assisted VADER において導入した booster，negation，contrast の 3 種類の判定ゲートについて，それぞれの寄与を明確にするアブレーション実験が必要である．具体的には，各ゲートを個別に有効化・無効化した条件を比較することで，どの構造判定が性能改善に最も寄与しているかを定量的に評価することが考えられる．
2. レビュー文の長さに着目した追加分析が挙げられる．商品レビューは文長のばらつきが大きく，文が長くなるほど修飾関係や否定構造が複雑化する傾向がある．また，原版 VADER は短文テキストを主な対象として設計されており，長文に対しては十分に対応できない可能性がある．今後は token 数や文長に基づいてデータを層別化し，短文・長文それぞれにおける LLM-assisted 判定の有効性を分析することで，提案手法の適用特性をより詳細に把握することが可能となる．
3. 参照 window 構造そのものの再設計が考えられる．一つの方向性としては，window size を拡張することで感情語から離れた修飾語や否定表現を取り込む設計が挙げられる．さらに，LLM が文全体の文脈を考慮して感情語と修飾語の対応関係を直接推定できる特性を活用し，固定長の参照 window に依存せず，関係があると判定された語のみをルール適用対象とする枠組みへ発展させることも考えられる．このような設計により，当時のルールベース技術の制約から固定的な参照 window 構造を採用せざるを得なかった VADER の考え方を，LLM による文脈的関係推定を通じて，より柔軟な形で実装し直すことが可能となる．

文　　献

[1] C. J. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text,” Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media (ICWSM), pp. 216–225, 2014.

[2] N. Alturaief, H. Aljamaan, and M. Baslyman, “AWARE: Aspect-Based Sentiment Analysis Dataset of Apps Reviews for Requirements Elicitation,” in *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pp. 211–218, 2021.

[3] A. Aramanda, S. M. Abdul, and R. Vedala, “Emotions in recommender systems for discrepant-users,” *Knowledge and Information Systems*, Vol. 67, pp. 953–976, 2025.

[4] N. Zuhir, A. M. Salim, P. Premkumar, and M. Farazi, “Beyond Stars: Bridging the Gap Between Ratings and Review Sentiment with LLM,” arXiv preprint arXiv:2509.20953, 2025.

[5] A. Almansour, R. Alotaibi, and H. Alharbi, “Text-rating review discrepancy (TRRD): an integrative review and implications for research,” *Future Business Journal*, Vol. 8, No. 1, Article 3, 2022.

[6] Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and J. McAuley, “Bridging Language and Items for Retrieval and Recommendation,” *arXiv preprint arXiv:2403.03952*, 2024.

LLM に基づく説明可能なソーシャルネットワークの将来予測

方 俊翔 † 伊藤 寛祥 †† 徐 哲林 †† 森嶋 厚行 ††

† 筑波大学情報学群知識情報・図書館学類 〒305 – 8550 茨城県つくば市春日 1-2

†† 筑波大学図書館情報メディア系 〒305 – 8550 茨城県つくば市春日 1-2

E-mail: †{s2213022}@u.tsukuba.ac.jp, ††{ito, morishima}@slis.tsukuba.ac.jp {zhelin}@ce.slis.tsukuba.ac.jp

あらまし ソーシャルネットワークにおける将来予測は、ネットワーク進化の理解および将来関係の推定において重要な課題である。既存研究では、ヒューリスティック指標やグラフニューラルネットワーク（GNN）に基づく手法が主に用いられてきたが、多くの場合、これらは構造情報への依存が強く、ノードが持つ意味的コンテキストの活用や、予測結果の説明可能性に課題が残されている。一方、大規模言語モデル（LLM）は高度な文脈理解と推論能力を有するものの、ソーシャルネットワークの将来予測への有効性は十分に検証されていない。本研究では、LLM がソーシャルネットワークにおける構造的情報と意味的情報を統合し、将来予測にどの程度できるかを検討する。NIPS 共著ネットワークを用いた実験により、(i) LLM による予測精度の評価、(ii) 予測に寄与する情報要素の分析、(iii) 判断理由の生成の可能性の検討を行った。その結果、提案手法は既存手法と比較して高い精度を示すとともに、各要素が予測結果に与える影響と、予測根拠を自然言語で提示できることを確認した。

キーワード ソーシャルネットワーク予測, LLM, リンク予測, 説明可能性

1 はじめに

1.1 研究の背景

近年，インターネットの発展に伴い，人々のコミュニケーション形態は大きく変化している．Facebook，X（旧 Twitter），Instagram などの代表的なソーシャルネットワーキングサービス（SNS）は，世界中で広く利用されており，物理的距離を超えた情報共有や人間関係の構築が日常的に行われている．また，学術分野における研究者の共著関係ネットワークや，E コマースにおけるユーザと商品の購買関係など，多くの社会活動は人や対象をノード（頂点），関係性をエッジ（辺）として表現するグラフ構造データとして捉えることができる．

これらのソーシャルネットワークは，時間の経過とともに新たなノードやエッジが生成・消滅する動的な性質を有している．また，将来のヒット予測ランキングの結果からグッズの生産やイベント企画やソーシャルメディアでのトレンド予測など実世界で予測に基づいた計画が幅広く行われている．その結果，このような動的なネットワークに対して，観測済みのデータから将来の状態を推定するソーシャルネットワークの将来予測は，ネットワーク科学およびデータマイニング分野における重要な研究課題の一つである．

ソーシャルネットワークの将来予測は，実社会においても幅広い応用可能性を持つ．例えば，SNS における友人推薦やフォロー推薦，学術ネットワークにおける将来的な共同研究関係の予測，さらには感染症拡散モデルにおける接触関係の推定など，予測結果に基づく意思決定支援が多くの分野で行われている．そのため，精度の高い将来予測手法が強く求められている．

一方で，ソーシャルネットワークにおけるリンク形成は，単純な構造的要因のみによって決定されるものではない．個人の興味関心や行動履歴，社会的役割，さらには時代的・社会的文脈といった多様な要因が複合的に影響し合うことで成立している．このような複雑な意思決定過程を適切にモデル化することは，依然として困難な課題である．本研究におけるソーシャルネットワークの将来予測とは，時間 t までに観測されたネットワーク $G(t)$ に基づき，将来時刻 $t + \Delta t$ においてノード対間にエッジが形成されるか否かを推定する問題である．以降では，このリンク予測問題を中心に，既存研究および本研究の位置付けについて述べる．

1.2 既存研究と課題

ソーシャルネットワークの将来予測に関しては，これまでに多くの手法が提案されてきた．初期の研究では，共通近傍数（Common Neighbors）[1] や Jaccard 係数 [2] といったグラフ構造に基づくヒューリスティックな指標が広く用いられてきた．これらの手法は計算コストが低い一方

で，複雑な非線形関係や高次の依存関係を十分に捉えることは難しい．

従来のリンク予測では，グラフニューラルネットワーク（GNN）[3] を用いたリンク予測手法が主流となっている．Graph Convolutional Networks（GCN）[5] や Graph Attention Networks（GAT）[19] などの手法は，ネットワーク構造を低次元のベクトル表現に埋め込み，高精度な予測を実現している．さらに，Temporal Graph Networks（TGN）[8] に代表される動的グラフ学習手法も提案され，時系列情報を考慮した予測が可能となっている．

GPT-4 や LLaMA などの大規模言語モデル（LLM）の飛躍的な進化により，テキストデータに対する高度な推論能力と常識的知識の利用が可能となった．最新の研究では，LLM をグラフ学習に統合する試みが急速に進展している．例えば，TAPE [10] は LLM を用いてテキスト属性から説明可能な特徴を抽出し GNN を強化する手法を提案しており，LinkGPT [11] は LLM を直接的なリンク予測器として利用するエンドツーエンドのフレームワークを提示している．また，ReaL-TG [12] のように，強化学習を用いて LLM に時系列グラフの推論能力を獲得させる試みも登場している．

しかし，これらの既存手法にはいくつかの課題が残されている．第一に，ノードが持つテキスト情報や意味的背景を十分に活用できていない点である．多くの手法では，テキスト属性を単純な特徴量として扱っており，語義や文脈といった高次の意味情報を適切に反映することが困難である．第二に，予測結果に対する説明可能性が低い点である．GNN を含む多くの深層学習モデルはブラックボックス的性質を持ち，なぜ特定のリンクが予測されたのかを人間に理解可能な形で説明することが難しい．このことは，実社会での応用を考える上で大きな制約となる．

1.3 本研究の目的とアプローチ

本研究の目的は，LLM の高度な言語理解能力および推論能力に着目し，ソーシャルネットワークの将来予測，において，どの程度の予測精度が達成可能であるかを明らかにするとともに，その予測過程を人間が理解可能な形で説明できるかを検討することである．

本研究では，ソーシャルネットワークの将来予測を単なる構造的欠損補完問題としてではなく，ネットワーク上のノードが有する文脈情報に基づく意思決定結果として捉える．具体的には，各ノードに付随する過去の接続履歴，周辺構造，および属性・テキスト情報を将来予測における入力コンテキストとして整理し，これらの情報を LLM に与えることで，リンク形成の有無を推論させる枠組みを採用する．

本研究の第一の関心は，LLM がグラフ構造およびその時間的文脈をどの程度捉え，将来のリンク形成を予測可能であるかという点にある．さらに，構造的情報，意味的情報，および時系列情報のうち，どの要素が予測性能に大きく寄与しているのかを分析することで，LLM による予測がどのような情報に依存して行われているのかを明らかにする．

また，本研究では，LLM による予測結果に対して，「なぜそのノード対が将来つながると判断されたのか」という判断根拠を，自然言語によって説明させることを試みる．これにより，予測精度のみならず，予測理由の一貫性および人間にとっての理解可能性の観点から，LLM を用いたソーシャルネットワーク将来予測手法の有効性と限界を検討する．

1.4 論文の構成

本論文は関連研究（第 3 章），問題設定（第 3 章），前提知識（第 4 章）提案手法（第 5 章），実験（第 6 章），まとめ（第 7 章）の順に構成される．

2 関連研究

ソーシャルネットワークのリンク予測は，未観測または将来のリンクを推定する古典的課題である．初期のヒューリスティック手法（共通近傍数，Jaccard 係数等 [1, 2]）は解釈性が高いものの，大域構造やノード属性の活用に限界があった．その後，DeepWalk [3] や Node2Vec [4] を代表とするネットワーク表現学習，GCN [5] 等の GNN，さらに EvolveGCN [7] や TGN [8] といった動的 GNN が提案され，時間的依存関係のモデル化により精度が向上したが，依然として構造情報への依存度が高く，非構造データの意味的活用が不十分である．近年の LLM は，Transformer アーキテクチャによる広範な世界知識と，CoT プロンプティング [9] による多段階推論能力を備え，推論エンジンとしての機能が実証されている．この LLM とグラフ学習の融合研究は大きく「Enhancer」と「Predictor」に分類される：Enhancer アプローチ（例：TAPE [10]）は LLM で抽出した特徴で GNN を強化するが説明可能性に課題があり，Predictor アプローチ（例：LinkGPT [11]）は LLM に直接予測を行わせるものの，大規模グラフでのコンテキスト制約や計算コストが問題となる．本研究では，LLM を単なる特徴抽出器や予測器ではなく，構造的・意味的情報を統合して推論する主体として位置付け，特徴抽出・文章化（モジュール 1）と推論・説明生成（モジュール 2）を分離した構成を提案する．これにより，LLM による情報利用の様式と推論・説明能力を検討することを目的とする．

3 問題設定

本章では，本研究で扱うソーシャルネットワーク将来予測問題を明確に定式化するとともに，本研究が検討する三つの研究課題（Research Questions）を示す．本研究は，LLM を用いた将来リンク予測に関して，その予測性能，情報依存性，および説明可能性を体系的に分析することを目的とする．

3.1 RQ1：LLM によるソーシャルネットワーク将来予測の精度評価

本研究では，リンク予測問題を，単なるグラフ構造の欠損補完としてではなく，ノードが周囲の環境に関する多様な情報を踏まえて関係形成を行った結果として捉える．すなわち，ソーシャルネットワークにおけるリンク形成は，確率的に自動生成されるものではなく，過去の相互作用，周辺構造，およびノードに付随する意味的情報など，複数の要因が統合された判断の結果として生じると考えられる．

例えば，研究者が将来の共著相手を選択する場合，相手の研究分野やこれまでの実績といった言語的に表現可能な情報に加え，過去の交流頻度や所属コミュニティの活発さといった数値的・構造的な情報も同時に考慮される．このように，ソーシャルネットワークにおけるリンク形成は，意味的情報と構造的情報が相互に関係しながら意思決定に影響を与える過程として理解することができる．

本研究では，このようなリンク形成の背景にある情報統合の過程を，分析上の観点として「感知（Perception）」と「認知（Cognition）」という二つの側面に分けて捉える．ここでいう 感知 とは，ネットワーク構造やノード属性など，予測に関連する情報が与えられる状況を指し，認知 とは，それらの情報に基づいて将来のリンク形成について熟慮し判断が行われる過程を指す．本研究は，これらの情報統合の視点に基づき，LLM を用いた将来リンク予測がどの程度の予測精度を達成可能であるかを検証する．

3.2 RQ2：LLM による予測において重要となる情報の分析

ソーシャルネットワークに関する情報は，大きく分けて，ノード間の接続関係や次数，共通近傍数などの構造的情報，ノード属性や投稿内容といった意味的情報，および時間的な変化を表す時系列情報に分類できる．本研究では，これらの情報を段階的に制御・削減しながら LLM に入力することで，各情報要素が予測結果に与える影響を分析する．

このような分析により，LLM が推論を行う際にどの情報に強く依存しているのか，また，従来の構造中心の手法とは異なる情報利用の傾向が存在するかを検討する．

3.3 RQ3：LLM による将来予測結果の説明可能性の検討

近年，予測精度の向上に加えて，予測結果の解釈性や説明可能性の重要性が指摘されている．特にソーシャルネットワークに関する予測は，実社会における意思決定支援に用いられる場面が多く，「なぜその予測が行われたのか」を説明できることが求められる．

本研究では，LLM に予測結果とともに判断理由の生成を行わせ，その説明が入力情報と整合的であるか，また，人間にとって理解可能な内容となっているかを定性的に分析する．これにより，LLM を用いた将来予測における説明可能性の有効性と限界を明らかにする．

3.4 記号・定義

本研究で扱う時系列グラフ（Temporal Graph）および関連する概念を以下のように定義する．

- **時系列グラフ** $\mathcal{G}$: 時系列グラフは，タイムスタンプ付きの相互作用（エッジ）の集合として定義される．

  $$\mathcal{G} = \{(u_i, v_i, t_i) \mid u_i, v_i \in \mathcal{V}, t_i \in \mathcal{T}, 0 \le t_1 \le \cdots \le t_{|\mathcal{G}|}\}$$

  ここで，$\mathcal{V}$ はノード集合，$\mathcal{T}$ は離散的な時間ステップを表す．
- **グラフスナップショット** $G^{(t)}$: 特定の時間ステップ t におけるグラフの状態を表す．

  $$G^{(t)} = (V, E^{(t)}, X^{(t)})$$

  ここで，$E^{(t)}$ は時刻 t に存在するエッジ集合，$X^{(t)}$ は各ノードに付随する属性情報（テキストを含む）を表す．
- **動的ペルソナ** $\mathbf{P}_u^{(t)}$: ノード u の時刻 t における発展方針を動的ペルソナと定義する．これは以下の要素を含む構造体である．

  $$\mathbf{P}_u^{(t)} = \{\text{EvolutionState}^{(t)}, \text{Memory}^{(t)}\}$$

  ここで，$\text{EvolutionState}^{(t)}$ は活動の拡大・縮小傾向，$\text{Memory}^{(t)}$ は過去の接続履歴を示す．

本研究のタスクは，時刻 t までに観測された履歴情報 $\{\mathcal{G}_\tau \mid \tau \le t\}$ に基づき，将来時刻 $t+1$ において，ノード対 (u, v) 間にエッジが形成される確率 $y_{uv}^{(t+1)} \in [0, 1]$ を推定することである．

4 前提知識

本章では，提案手法を構成する要素技術について解説する．特に，コード実装で用いられているグラフ解析手法と，LLM の推論技術について詳述する．

4.1 時系列グラフにおける構造的特徴量の抽出手法

LLM はテキスト処理には長けているが，グラフのトポロジー構造を数値行列として直接理解することは苦手とする．そのため，ノードがネットワーク内での自身の立ち位置を把握するための補助として，構造的特徴量を事前に計算し，言語化して与える必要がある．

4.2 スペクトル埋め込み (Spectral Embedding)

ネットワークの大域的な構造類似性を捉えるために，スペクトル埋め込みを用いる．これはグラフのラプラシアン行列の固有値分解に基づく手法である．隣接行列を A，次数行列を D とするとき，正規化ラプラシアン行列 $L_{sym} = I - D^{-1/2}AD^{-1/2}$ を構成する．この行列の固有ベクトルを用いることで，グラフ上で構造的に近いノード同士（例えば同じコミュニティに属するノード）は，ベクトル空間上でも近い距離に配置される．本研究では，この埋め込みベクトルのコサイン類似度を「構造的類似性 (Structural Similarity)」として LLM に提示する．

4.3 PageRank と中心性

ノードの重要度や影響力を測るために，PageRank アルゴリズムを使用する．これは，多くの重要なノードからリンクされているノードは重要であるという再帰的な定義に基づく．

$$PR(u) = \frac{1-d}{N} + d \sum_{v \in \mathcal{M}(u)} \frac{PR(v)}{deg(v)} \tag{1}$$

4.4 集積係数 (Clustering Coefficient)

集積係数は，あるノードの隣接ノード同士が互いに接続している度合い，すなわち「友人の友人は友人である」確率を表す指標である．本研究では，ノードが所属するコミュニティの密度を LLM に伝達するために使用する．ノード u の次数を k_u，その隣接ノード間に実際に存在するエッジ数を L_u とするとき，集積係数 C_u は次式で定義される [13].

$$C_u = \frac{2L_u}{k_u(k_u - 1)} \tag{2}$$

C_u が高い場合，そのノードは強固で閉じたコミュニティに属していることを示唆し，低い場合は異なるグループをつなぐ位置にいる可能性がある．

4.5 共通近傍数 (Common Neighbors)

Common Neighbors (CN) は，2 つのノード u と v が共有する隣接ノードの数である．

$$CN(u, v) = |\mathcal{N}(u) \cap \mathcal{N}(v)| \tag{3}$$

社会ネットワークにおいては，共通の友人が多いほど，その 2 人も友人になる可能性が高いという「三者閉鎖（Triadic Closure)」の原理が働くため，この指標はリンク予測において強力な特徴量となる．

4.6 ジャカード係数 (Jaccard Coefficient)

共通近傍数を，両ノードの次数（友人数）の和集合で正規化したものである．

$$Jaccard(u, v) = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v)|} \tag{4}$$

次数が非常に高いハブノード（有名人など）とのリンク確率が過大評価されるのを防ぐ効果がある．

4.7 重み付き・時間的指標

時系列グラフにおいては，単なる接続の有無だけでなく，「いつ」「何度」接続したかが重要となる．本研究では，最近の相互作用を重視するために，時間減衰（Time Decay）を考慮した重み付き指標を用いる．これにより，ノードにとって「最近よく交流しているコミュニティ」を適切に反映させることができる．

4.8 ベクトル類似度尺度

本研究では，スペクトル埋め込みによって得られた構造ベクトルや，キーワード集合から生成された特徴ベクトルの類似性を評価するために，コサイン類似度（Cosine Similarity）を用いる．二つのベクトル $\mathbf{a}, \mathbf{b}$ のなす角を θ とするとき，コサイン類似度は以下のように定義される．

$$Sim_{cos}(\mathbf{a}, \mathbf{b}) = \cos(\theta) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{5}$$

5 提案手法

本章では，ソーシャルネットワークにおけるリンク予測を，LLM による多層的な感知（Perception）と適応的な熟慮（Cognition）の統合過程として定式化する．提案手法である **Neuro-Symbolic Process)** は，時間的ネットワーク履歴とノード属性（テキスト）を入力として受け取り，対象ノード対 (u, v) に対する将来リンクの成立確率と，その判断根拠となる自然言語説明を同時に生成する枠組みである．本章では，提案手法の全体像（図 1）と，モジュール 1 / モジュール 2 によって実装される各モジュールの役割・入出力・設計意図を順に述べる．

5.1 システムアーキテクチャ

図 1 本研究で提案する NSP の全体フレームワークを示す．本システムは，以下の二つの処理系を明確に分離・統合したハイブリッドアーキテクチャを採用している．具体的には，(i) 大規模ネットワークから解釈可能な構造・意味シグナルを抽出する **Module 1（Perception）**と，(ii) それらのシグナルと過去事例を根拠として熟慮的推論を行

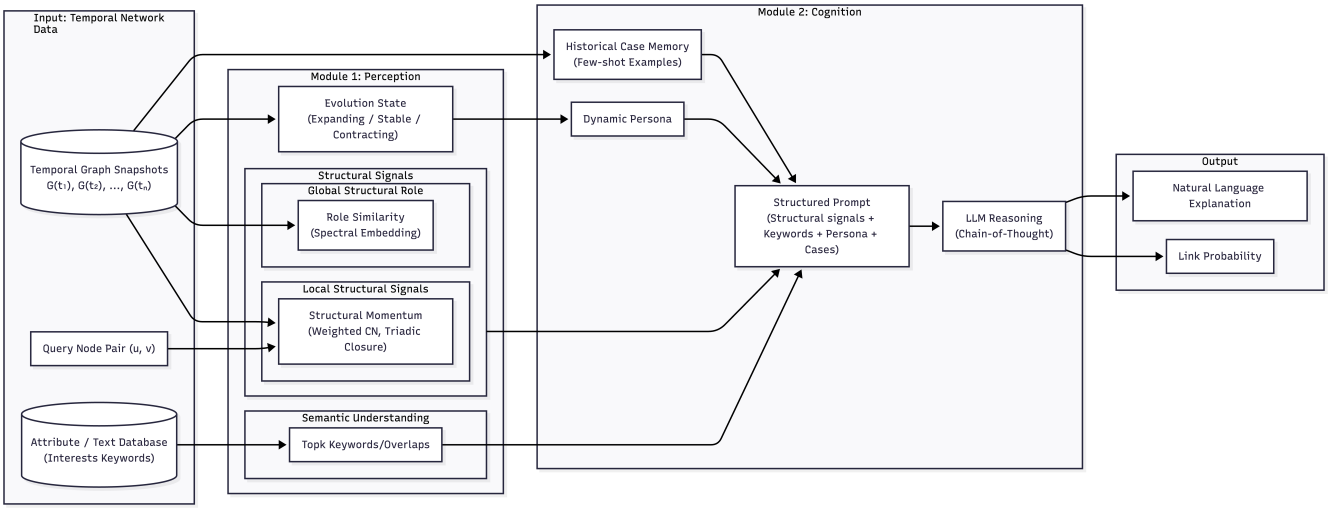

図 1: (NSP)Neuro-Symbolic Process. Input 層（GraphDB, TextDB, QueryPair）を参照し，モジュール 1 が構造・意味・状態シグナルを抽出，モジュール 2 が動的ペルソナ・過去事例・CoT 推論を統合してリンク確率と説明を生成する．

う **Module 2（Cognition）**によって構成される．

5.1.1 Input 層：Temporal Graph Database と Attribute Database

図 1 の Input 層は時間ステップ $t_1, \ldots, t_n$ におけるグラフスナップショット $G(t)$ を蓄積した **Temporal Graph Snapshots** と，各ノードの関心・研究キーワード等の属性を格納した **Attribute / Text Database** を用意する．推論時には対象ノード対 (u, v) を与え，Graph Snapshots / Text Database から必要な情報を取得して Module 1 / 2 に供給する．

5.1.2 モジュール 1/2：LLM を用いたパイプライン

- **Module 1（Perception / Signal Processing）**: LLM の「知覚」に相当する．GraphDB / TextDB から取得した情報に対し，(i) 構造シグナル（大域役割・局所動態），(ii) 意味シグナル（Top-k キーワード重複等），(iii) ネットワーク進化状態（拡大/安定/縮小）を計算し，LLM が解釈可能な形に整理する．すなわち情報のプリプロセッサである．
- **Module 2 (Cognition / Symbolic Reasoning)**: LLM の「熟慮」に相当する．Module 1 のシグナルを根拠として動的ペルソナ（行動戦略）を形成し，過去の類似事例（Memory）を参照しつつ，Chain-of-Thought に沿ってリンク形成を判断する．さらに，確率出力に加えて自然言語の説明を生成する．

この構成により，ネットワーク科学の厳密性（構造指標）と LLM の柔軟な統合推論能力（説明生成・意味統合）を両立する．

5.2 感知モジュール (Perception)

Module 1 は，生のグラフデータおよびテキスト属性を入力とし，図 1 に示すように，(a) 構造シグナル，(b) 意味シグナル，(c) 進化状態の 3 観点から特徴抽出を行う．ここで重要なのは，Module 1 の出力はブラックボックスな埋め込み表現ではなく，**LLM が言語化・比較・統合できる「解釈可能な証拠（evidence)」**として構成される点である．

5.3 認知モジュール (Cognition)

Module 2 では，Module 1 が抽出した構造シグナル（GlobalSig/LocalSig），意味シグナル（SemanticSig），進化状態（StateSig）を統合し，対象ノード対に対するプロンプトで プロフィルを構築する．

具体的には，Evolution State Detector の出力に基づき，探索（Exploration）と活用（Exploitation）のペルソナ方針を調整する．さらに，意味シグナル（キーワード）を用いて，対象ノードの関心をプロフィルに反映し，推論時に「どのような相手をパートナーとみなすか」という判断基準を明確化する．

Module 2 の中核をなすのが，LLM による推論プロセスである．図 1 に示すように，本手法は Module 1 の各シグナル，動的ペルソナ，さらに過去の類似事例メモリ（Memory）を **Structured Prompt** として統合し，LLM に入力する．

Structured Prompt（図 1 の Prompt）は，(i) Structural Signals（Global/Local），(ii) SemanticSig，(iii) Persona，(iv) Memory を一つの推論コンテキストに整理して含む．これにより，LLM は「どの証拠に基づき，どの順序で判断するか」を段階的に実行できる．

1. **Step 1: Temporal Structure Analysis (時間的構造解析)**
   ここでは Module 1 が算出した重み付き共通近傍数（Weighted CN）や，トライアド閉鎖の発生イベント数を確認する．
2. **Step 2: Global Structural Role (大域的役割の評価)**
   次に，直接的な繋がりがない場合でも，ネットワーク内での役割が似ているかを確認する．
3. **Step 3: Research Fit (研究関心の適合性)**
   ここではキーワードの関係性を確認する．
4. **Step 4: Learn from History (過去の事例参照)**
   Memory モジュールから検索された，類似した過去の成功/失敗事例を参照する．
5. **Step 5: Synthesis (総合判断)**
   最後に，すべての要素を統合して最終決定を下す．

最後，本手法では，将来リンクの成立確率を推定するだけでなく，その判断根拠を人間に理解可能で説得的な自然言語として提示することを目的とする．そのため，推論結果として確率値と併せて簡潔な理由文を生成する設計とした．

この段階的な推論を経ることで，数値モデルでは検出できない微妙なニュアンスや学際的交差を汲み取り，かつ人間にとって納得感のある「説明」を生成することが可能となる．

6 実験

6.1 データセット（NIPS 共著ネットワーク）

提案手法の有効性を検証するために，機械学習分野のトップカンファレンスである NIPS（Neural Information Processing Systems）の共著ネットワークデータセットを用いた．このデータセットは，著者（ノード）と共著関係（エッジ）から構成され，各著者はその年に発表した論文のタイトルに含まれる単語を属性として持ち，ノード数は 32，特徴数は 2411 である．データは 2008 年から 2017 年までの 10 年分を含んでおり，これを 1 年ごとのスナップショットに分割して時系列データとした．前半を学習用，後半をテスト用として使用した．

なお，評価フェーズにおいては，LLM の推論に伴う膨大な計算コスト（時間およびトークン消費量）を現実的な範囲に抑制するため，テストデータに対してサンプリングを適用した．

6.2 実験設定

提案手法における推論エンジンとして **Llama3:70B** を採用した．なお，モデルの実行環境には **Ollama** フレームワークを使用し，ローカル環境にて推論を行っている．

比較対象として，リンク予測において代表的な以下の手法を採用した．

- **Heuristic-based methods**: Jaccard 係数 [1]．いずれも局所的なネットワーク構造のみに基づく単純な指標であり，時間的情報やノード属性は考慮しない．
- **Static Embedding methods**: Node2Vec [4]．静的なグラフ構造からノード埋め込みを学習し，構造的類似性に基づいてリンクを予測する代表的手法である．
- **Temporal Sequence models**: LSTM [17]. 各ノード対の時系列特徴を入力とし，時間的依存関係をモデル化する手法である．
- **Graph Neural Networks**: GCN [5]．グラフ構造とノード近傍情報を用いて表現学習を行うが，本実験では各スナップショットを独立に扱う静的設定で評価した．
- **Reinforcement Learning based method**: RL [18]．リンク形成を逐次的な意思決定問題として定式化し，報酬に基づいて方策を学習する手法である．

リンク予測では以下の指標を用いて評価を行った．

- **AUC-ROC**: 受信者操作特性曲線の下側面積．全体的なランキング性能を評価する．

6.3 実験結果（時間的リンク予測性能）

図 2 に，各時間ステップ（時間ステップ 5–9）における AUC-ROC の推移を示す．

提案手法は，全ての時間ステップにわたって一貫して高い性能を示し，平均 AUC-ROC は **0.9545** に達した．

特に，ネットワーク構造が変化する時間区間においても，予測性能が大きく低下することなく安定して維持されている点が確認できる．この結果は，本手法の設計がソーシャルネットワークの将来予測において有効に機能していることを示唆している．

また，各スナップショットにおける性能のばらつきが小さいことから，本手法は単一時点に依存せず，ネットワークの進化過程全体を通じて安定した予測が可能であると考えられる．

6.4 アブレーション実験による各構成要素の寄与分析

提案手法における各構成要素の寄与を検証するため，主要コンポーネントを一つずつ除去したアブレーション実験

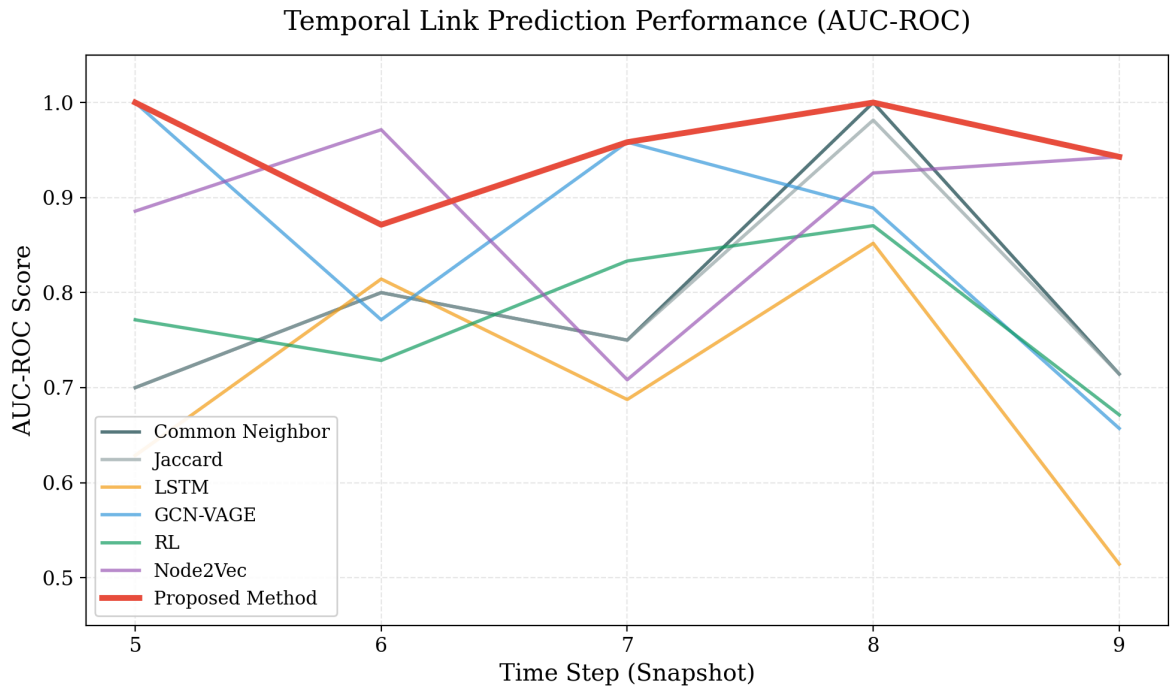

図 2: 各スナップショットにおける AUC-ROC の推移

を行った．その結果を表 1 に示す．

以上アブレーション実験の結果より，提案手法の高い性能は単一の要素によるものではなく，各シグナルを統合する設計に支えられていることが示された．

6.5 ケーススタディ：生成された判断理由

提案手法が生成する判断理由の具体例を示すため，リンク予測結果の一部についてケーススタディを行う．表 2 に代表的な 3 例を示す．

6.6 考察

本節では，第 6 章で得られた実験結果を踏まえ，提案手法 NSP が高い予測性能を示した要因，および各構成要素の役割について考察する．特に，本研究で設定した RQ1（予測精度），RQ2（重要情報の分析），RQ3（説明可能性）の三点との対応関係に着目する．

6.6.1 時間ステップを通じた安定性のある高い精度

図 2 に示したように，提案手法は 時間ステップ 5–9 の全ての時間ステップにおいて高い AUC-ROC を維持し，平均で 0.9545 を達成した．時間変化に頑健な予測精度を維持できることが確認され，RQ1「LLM による将来リンク予測の精度評価」に対する肯定的な結果を示している．

6.6.2 アブレーション結果から見た主要因：局所構造動態と意味情報

アブレーション実験の結果から，LLM によるリンク予測の核心要因と補助的要素が明らかとなった．局所構造情報（Local Structural Signals）と意味情報（Semantic Signal）は予測性能に決定的な影響を与え，これらを除去すると性能が大幅に低下したことから，複数の異種情報を統合した推論が有効であることが実証された（RQ2 への回答）．一方，大域的役割情報（Global Structural Role）と動的ペルソナ は補助的な役割を担う：大域的役割情報は直接的な近傍構造では説明しにくいケース（コミュニティ跨ぎの接続等）で潜在的な互換性を示し，動的ペルソナは探索 / 活用の推論方針を安定化させる機能を発揮した．また，Historical Memory は平均 AUC-ROC への直接的な貢献は限定的であったものの，類似事例に基づく説明文の説得力向上や，疎・大規模データ環境での推論の頑健性を高める価値を持つと考えられる．

6.6.3 生成された判断理由の解釈性に関する考察

ケーススタディ（表 2）では，共通近傍数（CN）や意味的重複率といった数値的証拠に基づき，予測確率と自然言語による判断理由が整合的に生成されていることが確認できる．例えば，CN や Semantic Overlap が低い場合には「moderate chance」「reasonable likelihood」といった表現が用いられ，数値的条件の弱さが言語的にも反映されている．

この結果は，LLM が与えられた構造・意味シグナルを単に読み上げるのではなく，相対的な強弱を踏まえて解釈していることを示しており，RQ3「予測結果を説明可能か」に対して，定性的に肯定的な示唆を与えるものである．

表 1: Ablation Results across Temporal Snapshots (AUC-ROC)

Model Variant	t5	t6	t7	t8	t9	Average
Full Model	1.0000	0.8714	0.9583	1.0000	0.9429	**0.9545**
No Global Structural	1.0000	0.7857	0.9167	0.9907	0.9143	0.9215
No Local Structural	0.9286	0.8000	0.8125	0.8796	0.7857	0.8413
No Semantic Signal	0.8714	0.9143	0.9167	0.7407	0.7429	0.8372
No Dynamic Persona	1.0000	1.0000	0.7708	0.9630	0.9643	0.9396
No Historical Memory	0.9714	0.9000	1.0000	0.9259	0.9571	0.9509
Raw Graph Baseline	0.8000	0.8000	0.7500	0.6852	0.7357	0.7542

表 2: 生成された判断理由のケーススタディ

例	Reasoning（生成文）	Probability
CN = 2.55 Semantic Overlap=50%	Researcher A and B have a moderate chance of collaborating due to their shared research focus on artificial intelligence and selection, as well as their compatible network positions and stable evolution trends, indicating a sense of stability and reliability that can foster trust and cooperation.	0.75
CN = 2.75 Semantic Overlap=25%	Both researchers have stable network positions and similar research focuses, indicating a potential for mutual benefit and complementary expertise. While strong historical connections are limited, the overall context suggests a reasonable likelihood of academic collaboration.	0.65
CN = 0.61 Semantic Overlap=12%	Although they have not collaborated before, their similar research orientations and stable trends suggest a potential collaboration motivated by exploring new research directions and diversifying academic connections.	0.40

7 まとめ

本研究では，ソーシャルネットワークにおける将来リンク予測を，単なる構造的な欠損補完としてではなく，各ノードが環境から得られる証拠を統合して意思決定を行う**認知プロセス**として捉え直した．その上で，LLM を「予測器」ではなく「推論エンジン」として位置付け，構造・意味・状態の解釈可能なシグナルに基づく Neuro-Symbolic な推論枠組み **NSP（Neuro-Symbolic Process）**を提案した．

本手法の特徴は，処理を二つの段階（Module 1 / Module 2）に分け，それぞれの役割を明確に整理した点にある．Module 1 では，動的ネットワーク履歴とテキスト属性から解釈可能な構造・意味・状態シグナルを抽出し，Module 2 では，それらを根拠として LLM による統合的推論を行う．

実験では，NIPS 共著ネットワークを用いた評価により，提案手法が全期間にわたり高い予測精度を維持できることを示した（RQ1）．また，アブレーション実験を通じて，局所的構造動態および意味的情報が予測において重要な役割を果たしていることを確認した（RQ2）．さらに，ケーススタディにより，予測結果と整合的な自然言語説明が生成されることを示し，LLM を用いた説明可能な将来予測の可能性を示唆した（RQ3）．

本研究の成果は，ネットワーク科学と LLM の融合領域における新たな可能性を示したが，同時にいくつかの課題も残されている．

まず，LLM の推論コストは高く，数百万ノード規模

の巨大ネットワークへの直接適用は困難である．今後は，Module 1 のフィルタリング精度を向上させ，LLM の呼び出し回数を最小化する蒸留技術などを検討する必要がある．

今後はリンク（構造）の将来予測に加え，ノード属性（興味関心・研究キーワード等）そのものも時間とともに変化するという前提に立ち，属性の変化を同時にモデル化することで，より現実的な将来予測へ拡張することが課題である．

以上を通じて，本研究は，リンク予測を高精度に行うだけでなく，その判断根拠を人間に理解可能な形で提示するという観点から，LLM を用いた将来予測の新たな活用可能性を示した．また，より大規模かつ多様なデータセットへの適用や，説明の定量評価手法の確立を通じて，実運用に耐える予測枠組みへと発展させることを目指す．

謝辞

本研究の一部は JSPS 科研費 (22H00508，22K17944) と，JST CREST(Grant Number JPMJCR22M2) AIP チャレンジの支援を受けたものである．ここに謝意を示す．

参考文献

[1] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7), 1019–1031.

[2] Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social networks*, 25(3), 211–230.

[3] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710.

[4] Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

[5] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*.

[6] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[7] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., ... & Kaler, T. (2020). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 5363–5370.

[8] Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv preprint arXiv:2006.10637*.

[9] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.

[10] He, X., Bresson, X., Laurent, T., & Hooi, B. (2023). Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. *International Conference on Learning Representations (ICLR)*.

[11] He, Z., Liu, Z., & Zhao, P. (2024). LinkGPT: Teaching Large Language Models To Predict Missing Links. *arXiv preprint arXiv:2406.04640*.

[12] Ding, Z., Huang, S., Cao, Z., Kondrup, E., Yang, Z., Huang, X., Sui, Y., Yuan, Z., Zhu, Y., Hu, X., He, Y., Poursafaei, F., Bronstein, M., & Vlachos, A. (2025). Self-Exploring Language Models for Explainable Link Forecasting on Temporal Graphs via Reinforcement Learning. *arXiv preprint arXiv:2509.00975*.

[13] Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 440–442.

[14] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

[15] Lu, X., et al. (2024). Improving Temporal Link Prediction via Temporal Walk Matrix with Time Decay. *Advances in Neural Information Processing Systems (NeurIPS)*.

[16] Chen, Z., Mao, H., Li, H., Jin, W., Wen, H., Wei,

X., ... & Tang, J. (2023). Exploring the potential of large language models (LLMs) in learning on graphs. *ACM SIGKDD Explorations Newsletter*, 25(2), 42–61.

[17] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

[18] Miyake, K., Ito, H., Faloutsos, C., Matsumoto, H., & Morishima, A. (2024). NETEVOLVE: Social Network Forecasting using Multi-Agent Reinforcement Learning with Interpretable Features. *Proceedings of the ACM Web Conference 2024 (WWW '24)*, 2542–2551.

[19] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks. *International Conference on Learning Representations (ICLR)*.

[20] Trivedi, R., Farajtabar, M., Biswal, P., & Zha, H. (2019). DyRep: Learning Representations over Dynamic Graphs. *International Conference on Learning Representations (ICLR)*.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[22] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.

[23] Fatemi, B., Halcrow, J., & Perozzi, B. (2023). Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*.

[24] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*.

[25] Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 1373–1396.

[26] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5), 513–523.

[27] Evans, J. St. B. T., & Stanovich, K. E. (2013). Dual-Process Theories of Higher Cognition: Advancing the Debate. *Perspectives on Psychological Science*, 8(3), 223–241.