本邦金融分野における大規模言語モデルに関するサーベイと展望

中川 慧,* 平野 正徳† , 高野 海斗‡

May 21, 2025

Abstract

本論文は、日本国内における金融分野での大規模言語モデル（LLMs）の研究と応用動向を包括的に調査・整理したサーベイである。2022 年 11 月の ChatGPT 公開以降、2025 年 3 月までに発表された日本語の関連論文 69 本を収集し、タスク軸（分類・生成・予測・構築・評価・その他）および目的軸（意思決定の高度化・業務の効率化・モデル性能の評価）に基づいて分類・可視化を行った。最多の研究は「分類」タスクに集中しており、特にセンチメント分析や ESG 評価が中心的テーマとして浮上している。次いで「生成」タスクが多く、レポートや要約文の自動生成による業務の効率化を目指す研究が目立った。一方で、予測や評価タスクは相対的に少なく、今後の研究課題とされる。さらに、実務への応用に際しての判断基準と検討課題を提示した。

Keywords: LLMs, 大規模言語モデル, 金融応用

*大阪公立大学 経営学研究科 E-mail: kei.nak.0315@gmail.com (Corresponding Author)
†株式会社 Preferred Networks E-mail: research@mhirano.jp
‡野村アセットマネジメント株式会社 E-mail: takaito0423@gmail.com

1　はじめに

金融分野は、日々変動する市場価格、規制や制度の変更、そして取引データや財務報告、統合報告といった複雑かつ膨大な情報に囲まれた領域である。株式市場の変動、金利や為替レートの変化、マクロ経済環境の動向、グローバルな経済政策の影響や、さらには気候環境変動など、さまざまな要因が企業の財務状況や投資家の判断に深く関わっている。このような複雑でダイナミックな環境の中で、金融の専門家は、正確で迅速な意思決定を求められ、これを達成するために日々進化する技術や手法を駆使することが不可欠である。

このような背景から、金融分野では、理論と実務が密接に結びついている。新たな理論・技術が開発されると、それは即座に実務に応用される。例えば、新たな投資理論や (予測) 技術が資産運用に適用され、その効果が実際の市場で測定される [1, 2, 3]。また、企業が直面する複雑な経済活動に対応する会計処理や、市場の急変に対応する中で生じた運用手法やリスク管理手法が次なる研究テーマとなり、新たな理論やモデルの開発を促進してきた [4]。こうして、理論と実務は相互に影響を与え合いながら、金融および会計分野は絶えず発展を続けている。

近年、特に急速に発展している大規模言語モデル (LLMs) は、まさに新たな技術の象徴と言える存在である。LLMs は、多くの分野で多岐にわたる活用が進んでいる [5]。例えば教育分野において、LLM の応用範囲は広く、インタラクティブな教育コンテンツの生成や、学生のエンゲージメント向上のためのツールとして重要な役割を果たしている。具体的には、問題の自動生成や学習内容に基づいたカスタマイズされたフィードバックの提供が可能となり、学習者の理解が深まることが期待されている。また、LLMs は学生の質問に対して迅速かつ的確な解答を提供する能力を持ち、教師の負担を軽減し、より効果的な教育を実現するための重要な手段となりえる。

金融や会計分野でも、LLMs はその卓越した言語処理能力を駆使することで、従来の技術で達成できたタスクをさらに高度化し、精度と効率性を大幅に向上させるだけでなく、これまでの技術では実現できなかった新たなタスクを実行できることが期待されている。そして海外では既に、LLMs を活用した金融および会計分野の研究が進んでおり、実際に多くの応用事例が報告されている [6]。これらの研究は、LLMs が他分野での事例と同様にさまざま

なタスクにおいて有効であることを示している。

もっとも、日本市場は言語構造・データ環境・規制枠組みが英語圏と大きく異なり、英語コーパスを主とするLLMsがそのまま有効性を発揮するとは限らない。企業開示様式、日本独自の会計基準、金融庁ガイドラインといった制度面の差異、さらには漢字混じり文書特有の形態素粒度などが複合的に影響するためである[7]。従って、主として英語を基盤とするLLMsが、日本における金融分野で有効であるかどうかは自明ではない。さらに、日本における金融分野のLLMsの応用に関する包括的なサーベイは著者らの知る限り存在していない。一方で、ChatGPT公開（2022年11月[1]）以降、日本における金融分野の研究が爆発的に増加し、本稿が対象とした2022年11月から2025年3月のわずか30か月足らずで69本の関連論文が確認できた。

そこで本サーベイでは、日本における金融分野のLLMsの研究を包括的にレビューし、その応用可能性と課題を明らかにすることを目的としている。日本の金融分野のLLMsの研究事例を網羅的に収集し、現状を体系的に理解するための次の視点から研究を分類し、整理を行う。具体的には、対象論文を「分類」、「生成」、「予測」、「構築」、「評価」、「その他」という六つのタスク軸で区分し、さらに用途を「意思決定の高度化」と「業務の効率化」、「LLMsの性能評価」という目的軸に投影することで、現状の研究が領域を可視化した。分類軸と目的軸を組み合わせた俯瞰は、学術的位置づけを明らかにすると同時に、実務導入を評価できる点にある。タスク軸はテーマ別文献レビューの基本枠組みである[6]。一方、実務面では、金融機関が生成AIを採用する利点として、業務コスト削減と効率化（高度化）のどちらに寄与するかを挙げている[2]。したがって、タスクをこれら2目的に投影することで、技術的成果を経営指標に変換する余地ができ、LLMs導入を定量的に比較できる。さらに第三の軸として「LLMsの性能評価」は、安全な社会実装に不可欠な検証手続きが研究上も実務上も未整備である点に起因する。特に金融分野は様々な規制[8]でも、評価専用タスクを独立して設計しない限りモデル間比較が困難になると指摘している。特に金融実務はその重要性から様々な規制やリスク管理体制の下で運用されているため、モデルの出力に対する説明可能性や頑健性、さらには倫理的・法的な妥当性の観点といった、性能評価の体系的な枠組みが求

[1] https://help.openai.com/en/articles/6825453-chatgpt-release-notes
[2] https://deloitte.wsj.com/cfo/what-does-generative-ai-ready-look-like-for-finance-9ceb27c9

められている。金融機関が安心して導入するためには、タスクに応じた精緻な評価指標とベンチマークの整備が不可欠である。以上を踏まえて、本サーベイでは、タスク分類と目的軸によって日本における金融分野の LLMs 研究の体系化を試みる。またその活用可能性を評価し、今後の研究および実務への応用に向けた指針を示す。

2　範囲・方法

本研究では、日本における金融分野の大規模言語モデルの応用に関する研究事例を包括的にレビューすることを目的とし、ChatGPT がリリースされた 2022 年 11 月から 2025 年 3 月末までに発表された日本語の金融分野の論文を対象とした。

文献レビューの対象範囲は、以下の学会およびその論文誌に基づく。具体的には、人工知能学会（JSAI[3]）、人工知能学会金融情報学研究会（SigFin[4]）、言語処理学会（NLP[5]）、電子情報通信学会（IEICE[6]）、情報処理学会（IPSJ[7]）、経営情報学会（JASMIN[8]）であり、本邦において金融と情報技術の交差領域で活発な研究活動が行われている [9]。

サーベイ対象とする論文の抽出にあたっては、次のような検索条件を設定した。まず、論文タイトル、要旨（アブストラクト）、およびキーワードに、「大規模言語モデル」、「LLM(s)」、「ChatGPT」または「GPT」のいずれかを含むものを検索対象とした[9]。次に、それぞれの研究が金融・会計分野に関連しているかをタイトルおよび要旨、本文に基づいて著者らが判断し、レビューの対象とした。これらの基準により、最終的に 69 本の研究がサーベイ対象として特定された。

表 1に示す通り、該当論文のうち SigFin（29 件）と言語処理学会（NLP）（18 件）で全体の 68%を占めており、LLM の金融・会計分野への応用に関する研究がこれらの分野を中心に活発に行われていることがわかる。これは、従来の自然言語処理分野（NLP）と金融情報

[3] https://www.ai-gakkai.or.jp/
[4] https://sigfin.org/
[5] https://www.anlp.jp/
[6] https://www.ieice.org/jpn_r/
[7] https://www.ipsj.or.jp/
[8] https://www.jasmin.jp/
[9] ただし、第 34 回の SigFin において有価証券報告書の情報抽出に関するコンペティションが実施されているが、当該コンペティションに関する論文はコンペティションに特化された内容のため除外した。

Table 1: 学会論文誌別の研究数

学会・論文誌	該当論文数
人工知能学会金融情報学研究会（SigFin）	29
言語処理学会（NLP）	18
人工知能学会（JSAI）、論文誌	15
電子情報通信学会（IEICE／NLC）	6
経営情報学会（JASMIN）	1
情報処理学会（IPSJ）	0
合計	69

学という応用領域（SigFin）の双方で注目を集めていることを反映している。また、LLMsが自然言語処理における技術革新であるだけでなく、金融という応用現場で直面する複雑な情報処理課題に対しても有効な解決策となり得ることの期待を示唆している。

検索の結果特定されたこれらの研究を、以下の手順に従って分類および分析を行った。まず、それぞれの研究が対象とするタスクを、「分類」、「生成」、「予測」、「構築」、「評価」、「その他」の6つのカテゴリに分類した。それぞれのカテゴリは、金融分野のLLMs活用に際して直面する多様な課題を反映し、LLMsが提供できる解決策を検討するうえで重要な視点となると考える。そして、各研究の目的を「意思決定の高度化」と「業務の効率化」という二つの大きな軸に基づいて分類する。これは、金融機関等の導入意思決定において、LLMs導入がもたらす経済的インパクトの評価基準を、前者は主に収益面の向上に置き、後者は主にコストの削減に置くことを反映したものであり、実務的便益を考慮する点で有用である。

次に、タスクの解決に使用されたモデルと、各モデルに適用された手法を記録した。最後に、これらの研究を通じて明らかになった金融分野特有の課題やニーズを抽出する。

3　サーベイ結果

3.1　タスクの分類と考察

ここでは今回のサーベイで扱われた論文のタスクを、以下の6つのカテゴリに分類する。

分類: 分類カテゴリはテキストを所定のクラスに割り当てるタスクである。これらの分類タスクは、LLMs以前から金融分野のテキストマイニングの重要なタスクであり、投資判

断やリスク管理といった金融機関の業務に直結する。そのため、LLMs を用いて従来の手法を上回る精度や新たなタスクの金融実務への実用的な応用が期待される。

生成: 生成カテゴリはテキストや文書を自動生成するタスクである。決算短信の要約やマーケットコメントの生成，質問応答システムの構築等が具体的な事例として該当する。これにより、金融分野の専門家の業務効率が大幅に向上することが期待される。

構築: 構築カテゴリは、金融特化データセットや金融特化 LLMs モデルの構築に関するタスクである。LLMs やその他のテキストマイニングのアルゴリズムが動作する、あるいは評価の基盤を提供する。特に、「評価」に分類されるタスクと合わせて金融特有の課題に対応するためのデータセットやモデルの構築は、今後の LLMs の金融分野への応用研究のインフラとなるため重要である。

評価: 評価カテゴリは複数のモデルやデータセットの性能・公平性を金融タスクで検証するタスクである。LLMs の日本語や金融といった特定のドメインにおける LLMs や既存モデルの適応性、精度比較が含まれる。

予測: 予測カテゴリは時系列データや株価変動などの予測に関するタスクである。分類タスク同様に、投資判断やリスク管理といった金融機関の業務に直結し、実用的な応用が期待される。ただし、金融時系列は一般に予測が困難である [10]。

その他: その他のカテゴリには、上記に当てはまらない特定の金融分野の LLMs 活用に関連するタスクが含まれる。

Table 2: 6 つのカテゴリ別の研究数

カテゴリ	件数
分類	38 件
生成	11 件
構築	8 件
評価	5 件
予測	3 件
その他	4 件
合計	69 件

表 2は、上記の 6 つのカテゴリに基づく研究数の集計結果である。全体の中で、最も多いのが「分類」カテゴリに関するタスクであり、38 件が該当している。これは、金融分野においてテキストデータのセンチメント分析やタカハト分類など様々な分類タスクが存在するため、重要なタスクであることを反映している。分類タスクは、投資判断やリスク管理といった金融機関の業務に直接影響を与え、高い精度が求められる分野であるため、多くの研究が行われている。

次に多いのが、「生成」カテゴリに関するタスクで、11 件が該当している。生成タスクには、報告書やコメントの自動生成など、実務での業務効率化を目指したものが多い。以上から、**「分類」および「生成」タスクが LLMs の強みを生かした金融応用のカテゴリーとして注目されていることが分かる。**

「構築」カテゴリに該当する研究は 8 件あり、データセットやモデルの構築に関する研究が一定数行われていることが示されている。特に、金融特化型のデータセットや LLMs モデルの開発が進行中である。

一方で、「評価」分類される研究は 5 件、そして「予測」のカテゴリは、3 件にとどまっている。機械学習を用いたタスクにおいて、予測や推論時に意図しないバイアスが生じる問題が知られている [11]。LLMs においても同様の問題が発生し得るため、評価タスクでは、LLMs モデルのバイアスや性能を分析し、信頼性や解釈性の向上を目指した研究が行われている。対して、「予測」タスクは、金融市場の予測が非常に難しいタスクであり、慎重なアプローチが求められるため、他のタスクに比べて研究の数が限られている可能性がある。

3.2 具体的なタスク一覧

表 3は具体的なタスクの一覧をまとめたものである。分類カテゴリのタスクとしては、例えば、金融ニュースのタグ付けや ESG 評価、SDGs 関連文の分類など、金融分野では多様な分類タスクが存在する。特に ESG 関連の分類は、近年のサステナビリティに対する関心の高まりを背景に重要性を増している。また、FRB 金融政策のタカハト分析[10]、ファクト（事

[10] タカハト分析とは、中央銀行の政策や発言において、タカ派（hawkish）とハト派（dovish）に分類する分析手法である。タカ派はインフレ抑制のために金融引き締めを支持し、ハト派は経済成長を重視して金融緩和を支持する。

実)、オピニオン（意見）分類[11] や株価変動用語の分類など、従来のテキスト分析手法では困難な分類タスクに対しても LLMs の適用が進んでいる。

次に、生成カテゴリのタスクとして、決算短信の要約生成や市況コメントの生成、さらには企業の環境活動の改善案の生成など、アナリストやファンドマネージャーなどの金融専門家の作業負担を軽減し、効率的な情報提供を意図するものが多い。生成タスクでは、LLMs がユーザーの入力に対して適応的かつ柔軟に応答する能力があるため、より人間が作成した文章に近い自然な言語生成が期待される。

金融市場における独自の文脈や用語を理解するために、汎用的な LLMs に加えて、金融分野に特化したモデルの構築が必要である。このような専門特化型の事前学習モデルの開発は、金融業界における LLMs の普及と、その実用性と精度を向上させるための重要である。なぜなら、A.1章で言及した通り、汎用的な LLMs では、金融市場特有の高度な専門用語や規制文書、あるいは市場データの複雑な関係性を十分に捉えることが難しい場合が多い。金融分野に限らず専門用語は、一般的な文脈ではほとんど使われないため、汎用モデルの事前学習データには十分な情報が含まれていない。このギャップを埋めるためには、金融分野に特化したデータセットを活用し、専門用語や独自の市場動向を深く理解できるモデルを構築する必要がある。さらに、特化型モデルの構築においては、具体的なタスクに基づく問題意識が重要であり、これに基づき多様なベンチマークデータセットの作成も必要である。構築カテゴリのタスクとしては、このような問題意識から様々なベンチマークやモデル構築が求められる。

評価タスクにおいては、特に LLMs のバイアス評価が重要なテーマとして挙げられる。金融分野の機械学習のバイアスを評価した例として、米国の住宅ローンにおけるデフォルト率をランダムフォレストなどの機械学習で予測した研究 [12] がある。[12] では、ローン申込者の人種など、公平性の観点で使用に配慮が必要な属性を訓練データとして用いていなかったにもかかわらず、デフォルト率の予測値がローン申込者の上記属性によってバイアスを持つ結果が報告されている。LLMs も同様にその出力に様々なバイアスが生じることが実証され

[11] 金融テキストにおいて、定量情報や事実の説明である「ファクト」と、その解釈や見解、見通しなどの「オピニオン」を分類すること。「今期の利益は○○億円である。」といった定量情報は財務諸表等のその他の情報から取得できるが、「今期の利益は期待以上のものである。」といった定性情報はそのテキストに固有の情報であるため、重要性が高い。

ており、具体的には、人種や性別に関するバイアスがあり、これらは公平な意思決定を妨げる可能性がある [13]。LLMs を含む機械学習を金融分野で応用しサービス等を提供する際に、予測・推論の偏りに十分注意する必要がある。例えば、人種や性別といった配慮が必要な属性に関わる場合、それをそのままサービスに反映すると、利用者への意図せざる差別につながる可能性がある。他にも、センチメント分析におけるバイアスが誤った投資判断を導くリスクがある [14]。また、金融教育やファイナンシャルプランニングにおいても特定のバイアスが存在する場合には、適切な教育、助言が妨げられる恐れがある。よって、LLMs のバイアスを適切に評価し、分析し、その改善を図ることは、金融分野における LLMs 活用の信頼性の確保につながり応用上不可欠な作業である。ただし、これらの属性が明らかにタスクの精度向上の観点から有益である場合、公平性と効率性のトレードオフ関係が生じる。

LLMs は大量のデータを学習することで、パターン認識やトレンドの把握に優れた能力を発揮し、より精度の高い予測が可能となる可能性がある。実際に LLMs を用いた時系列予測の応用も盛んに研究されており、時系列予測においても高い性能が報告されている [15, 16]。そこで他の分野同様に金融時系列の予測タスクにおいても、LLMs は従来の予測モデルよりも高い精度の予測が可能となることが期待される。一方で、金融分野においては LLMs の時系列分析事例が少なく今後の発展が期待される領域である。

以上から、金融分野における LLMs の活用に関するタスクの多様性が表れており、様々なニーズに対応する需要があることが示唆される。

Table 3: 論文ごとのカテゴリおよびタスク一覧

論文	カテゴリ	タスク
[17]	構築	日本語金融ベンチマークの構築
[18]	分類	金融ニュースのタグ付け（分類）
[19]	分類	ESG 評価（分類）
[20]	生成	決算短信の要約生成
[21]	分類	ESG トピック分類

次のページへ続く。

論文	カテゴリ	タスク
[22]	生成	サプライチェーン関係の生成
[23]	分類	SDGs 関連文の分類
[24]	生成	市況コメントの生成
[25]	分類	決算説明会の主観的表現の分類
[26]	生成	企業の環境活動の改善案の生成
[27]	分類	公認会計士試験短答式監査論の回答（分類）
[28]	分類	FRB 金融政策のタカハト分類（センチメント分析）
[29]	生成	企業の環境活動の改善案の生成
[30]	生成	特許マップの生成
[31]	分類	TCFD 推奨開示項目の分類
[32]	分類	事業セグメント言及文抽出（分類）
[33]	分類	推論ベースのセンチメント分析
[34]	分類	FRB 金融政策のタカハト分類（センチメント分析）
[35]	分類	株価変動用語の分類
[36]	構築	日本語金融ベンチマークの構築
[37]	分類	ファイナンシャル・プランニング技能検定の回答（分類）
[38]	生成	市況コメントの生成
[39]	分類	MD&A の定性的表現の分類
[40]	分類	センチメント分析
[41]	評価	LLM のバイアス評価
[42]	評価	金融分野特有のタスクのインストラクション・チューニング
[43]	予測	金融時系列予測
[44]	生成	キーワード生成
[45]	分類	推論ベースのセンチメント分析（分類）
[46]	分類	株価変動用語の分類

次のページへ続く。

論文	カテゴリ	タスク
[47]	分類	ESG のセンチメント分析（分類）
[48]	分類	業績文の区切り位置推定とセンチメント分析（分類）
[49]	分類	技術的要因文の分類
[50]	その他	事業リスクの記述の次元削減（分散表現）
[51]	構築	金融事前学習言語モデルの構築
[52]	予測	不動産価値推定（予測）
[53]	分類	カーボンプライシング関連論文の分類
[54]	分類	カーボンプライシング関連論文の分類
[55]	分類	人的資本情報の分類
[56]	分類	公認会計士試験短答式企業法の回答（分類）
[57]	生成	テキストの二値分類の定義文生成
[58]	評価	金融テキストのセンチメント分析のバイアス評価
[59]	構築	日本語金融インストラクションデータセットの構築
[60]	その他	有価証券報告書の質問応答
[61]	分類	ESG 情報の抽出 (分類)
[62]	分類	会計基準に関する質疑応答
[63]	予測	利益変化の分類
[64]	分類	ESG 情報の抽出 (分類)
[65]	構築	金融特化 LLM の構築
[66]	構築	金融特化 LLM の構築
[67]	分類	金融テキストの二値分類
[68]	分類	見通し文と結果文のアラインメント
[69]	分類	ESG 情報の抽出
[70]	分類	表質問応答
[71]	構築	LLMs の性能評価

次のページへ続く。

論文	カテゴリ	タスク
[72]	分類	表構造認識
[73]	生成	株価変動記事生成
[74]	分類	会計分野の質問応答
[75]	分類	スキルマトリックス分類
[76]	評価	センチメント分析
[77]	その他	人工市場シミュレーション
[78]	その他	経済フェルミ推定
[79]	その他	センチメント分析
[80]	評価	物価センチメント分析
[81]	構築	自動レポート生成
[82]	生成	ESG 情報の抽出（分類）
[83]	分類	大量保有報告書の契約情報の抽出・分類
[84]	分類	業種区分の分類
[85]	分類	ESG 情報の抽出 (分類)

終了

3.3 タスクの目的

次に、今回のサーベイで取り上げた論文のタスクの目的を整理する。LLMs 活用の目的は、金融分野における **1. 意思決定の高度化**と **2. 業務の効率化・自動化**という 2 つの大きな軸で行われている。前者は、意思決定プロセスの精度向上やリスク管理の強化によって、直接収益に貢献する一方で、後者は膨大な作業を迅速かつ効率的に処理することで業務負担を軽減し、コスト削減に貢献する。また、金融市場における独自の文脈やドメイン知識を評価し、金融タスクにおける LLMs の性能や限界を明らかにするための **3. LLMs の性能評価**も重要な軸である。今後のモデル改善や最適化につなげる示唆を得ることで、前述の意思決定の高度化や業務の効率化がより進展する。表 4は、論文のタスクの目的分類を集計した結果で

ある。

Table 4: LLMs のタスクの目的分類の集計

目的分類	件数
意思決定の高度化	21
業務の効率化	29
LLMs の性能評価	19

1. 意思決定の高度化

金融分野における意思決定は、データに基づく定量的な分析だけでなく、経験や知識に裏打ちされた定性的な判断も重要な役割を果たしている。LLMs の導入は、この定性的な意思決定プロセスを高度化する可能性を秘めている。すなわち、LLMs はその高い言語能力により従来の定量的分析を補完し、これまで人間の判断に依存していた定性的な要素をデータに基づいてサポートすることができる。具体的な例は次の通り。

センチメント分析: 中央銀行の政策トーンや要人発言に対するタカハト分析は、中央銀行の将来の金融政策予測や、様々な金融資産の投資判断を下す上で重要である [86]。中央銀行が公表する文書の政策トーンを評価し、将来の金融政策見通しを捉える研究や中央銀行の要人発言に対するタカハトのセンチメントを自動で付与することで、金融政策の予測精度を高め、投資判断をサポートしている [34]。また、[48] では、業績文中の意味的な区切りを推定し、その各セグメントに対してセンチメントを付与する手法が提案され、高精度な文の切り出しと分類を実現している。さらに LLMs の推論能力を活用したセンチメント分析手法も提案されている。[45] では、企業の事業概要を LLMs を用いて自動生成し、それに基づいてイベント（自然災害、経済イベント等）が与える業績影響を推論し、センチメント評価を行う統合フレームワークが提案された。自動生成された事業要約は、事業構造の把握が困難な企業についても、影響評価を可能とする点で有用である。

業績および市場動向の予測: 金融時系列データの予測精度を向上させるため、LLMs の推論能力を活用した予測手法を構築 [43] することや、MD&A に含まれる定性的表現が経営者の業績予想の精度に与える影響を分析 [39] する、あるいは企業利益そのものを LLMs

によって予測 [63] することで投資判断の質を向上させている。また、株式市場以外では、[52] が、不動産価格推定における LLMs の適用可能性を検討し、LLMs を用いて地域特性やスペル誤りといった非構造的ノイズを処理しながら価格を推定する手法を提案している。

ESG 関連の意思決定: 事業文書から ESG 指標への影響を自動的に推論 [47, 19] する方法や、企業のサステナビリティ報告書から ESG 評価を実施する方法 [19] が提案されている。また、[26, 29] は、企業の統合報告書に記載された環境活動の一貫性を評価し、記述が不足している階層（特に PDCA の Plan や Do）に対して自動的に改善案を生成する手法を提案している。特に環境活動が PDCA サイクルに基づいて網羅的に記述されている企業では、収益性が高い傾向が示されており、ESG 活動の「質的充実」が財務的成果に結びつく可能性が示唆された。また、[75] では、企業が開示する株主招集通知の取締役推薦文から、取締役のスキルマトリックスを自動分類する方法が提案されている。取締役会の構成と ESG 戦略との整合性の評価が可能となる。

リスク・機会分析: 企業や投資家が将来の不確実性に対応するためには、リスクや機会に関する情報を的確に把握・解釈する能力が求められる。[49] は、マクロ環境における技術的・制度的変化をニュース記事から抽出・分類し、サステナビリティ・トランスフォーメーションにおけるマテリアリティを特定するフレームワークを提示した。脱炭素技術の市場実装進展や水素経済に対する社会的関心の高まりを定量的に可視化している。また、[50] では、パンデミック前後における企業の事業リスク認識の変化を分析し、感染症リスクに関する言及の頻度・内容の変遷を可視化した。企業がパンデミック期にどのような新たなリスク要因を認識したかを明らかにしており、経済全体に対するリスク認識の構造的変化が示唆された。

最後に、分類精度の向上だけでなく、分類根拠の明確化や解釈性の担保を目的とした「定義文自動生成」手法が提案されている。[57] では、テキストの (二値) 分類タスクにおいて、事前に定義文（分類の判断基準となるルール）を生成し、さらに誤分類事例を用いてこの定義文を動的に更新する枠組みが構築された。この枠組みのもとでの zero-shot ／ few-shot 学

習により、安定した精度と高い汎化性能を示すとともに、分類理由の可視化による解釈性の向上が確認された。続く [67] では、同様の枠組みを金融文書に適用し、分類精度と説明性を両立するチューニング手法を提案した。具体的には、BERT モデルによるベースライン手法に対し、LLMs を組み合わせて定義文の生成・改良を行うことで、より高い解釈性を実現した。これにより、金融業界における分類結果の説明責任やガバナンス要請にも応える実装ができる可能性がある。これらの研究は、LLM の強みである言語的柔軟性と説明能力を最大限に活かし、分類モデルの「ブラックボックス性」問題を軽減する有望なアプローチである。

2. 業務の効率化・自動化

金融分野においては、報告書や資料の作成、データ分析など多くの業務が依然として手作業で行われていることが少なくない。このような状況を改善するために、LLMs を活用し、業務の効率化を図ることが注目されている。業務の効率化の中心は、データ抽出、分類・要約といった反復的な作業の自動化であり、LLMs を活用することで、反復的なタスクを自動化することができる。また、このような自動化により、人的リソースを削減できるだけでなく、作業スピードの向上と精度の確保も期待できる。LLMs の導入による業務の効率化および自動化は、金融分野におけるデジタルトランスフォーメーションの一環としても位置付けられ、重要な目的である。具体的な応用例は次の通り。

文書作成の自動化: 株価変動を表す記事に使用される株式市場の専門用語を自動的に選択し、記事の自動生成を支援 [46, 35]、運用報告書における市況コメントと見通しを自動生成する [24, 38] ことで、文書作成業務を効率化することで、金融機関や投資家の業務負担を軽減することを目的としている。

データ抽出と分類: 有価証券報告書から事業セグメント関連情報を高精度に抽出 [32]、特許文書から技術課題と解決手段を抽出し、視認性の高い特許マップを自動生成 [30] するなど、金融文書からの情報抽出・分類を LLM が正確に行うことで、情報整理の迅速化することを目的としている。

ESG 関連項目抽出: ニュース記事から ESG に関連するトピックを自動的に抽出 [21]、統合報告書などから ESG に関連する情報を効率的に抽出 [61]、有価証券報告書におけるサ

ステナビリティ情報の開示状況を自動で判定し、開示コストを削減する [31] など、ESG に関連する情報抽出を自動化し、評価プロセスを迅速化することを目的としている。[?] では、バリューモデルを用いた ESG 評価の実装に向けて、企業報告書からの情報抽出手法が提案されている。企業ごとのレイアウトや表現の違いが抽出精度に与える影響が分析されており、報告書設計の標準化と自然言語処理技術のさらなる発展の重要性が示されている。

3. LLMs の性能評価

金融タスクにおける LLMs の性能を適切に評価すること重要な研究目的である。LLMs の性能や限界を明らかにし、今後のモデル改善や最適化につなげる示唆を得ることで、前述の意思決定の高度化や業務の効率化がより進展することが期待される。

金融知識評価: 海外での事例と同様に、LLMs が資格試験に合格できるかどうかを検証することで、該当分野の知識を評価する目的である。日本の公認会計士試験（短答式試験）の監査論および企業法に合格できるかを検証 [27, 56] したり、3 級 FP 技能検定 [37] に合格できるかを検証している。

LLMs の金融タスクでの有効性検証: 金融ニュースのタグ付けタスクにおける LLMs の有効性を検証し、従来の手法に比べた LLMs の性能や適用可能性を判断 [18]。また、金融分野特有のタスクでの性能を最大化するためのインストラクション・チューニングの効果を評価し、特定タスクに合わせた LLMs の最適化方法を見出すこと [59] などを目的としている。さらに、日本語の金融分野に特化したベンチマークを構築 [36] し、主要な言語モデルの性能を評価する [17] など、金融市場や日本語に焦点をあてた性能を検証する取り組みが行われている。

バイアス評価: LLMs の金融投資意思決定におけるバイアスを評価し、その影響を明らかにする [41]、LLMs の企業固有のバイアスが金融テキストのセンチメント分析に及ぼす影響を評価 [58] することで、LLMs が持ちうるバイアスを正確に評価し、その影響を把握する取り組みが行われている。

3.4 目的および分類のクロス集計

Table 5: 目的および分類のクロス集計

目的分類	その他	予測	分類	構築	生成	評価	合計
LLMs の性能評価	2	0	4	8	0	5	19
意思決定の高度化	1	3	14	0	3	0	21
業務の効率化	1	0	20	0	8	0	29
合計	4	3	38	8	11	5	69

表 5に示すように、目的分類とタスクカテゴリとの間に明確な偏りが認められる。第一に、LLMs の性能評価を目的とする 19 件の研究は、「構築」（8 件）と「分類」（4 件）が中心であり、データセット整備やベンチマーク設計といったインフラ構築型の試みが主導している。これは、汎用英語ベンチマークでは測定しきれない日本語金融タスクの特性に対応する必要性が高まっていることを裏付ける。

第二に、意思決定の高度化を目指す 21 件では、「分類」（14 件）が最多で、「予測」（3 件）や「生成」（3 件）が続く。とりわけ政策文書のタカ派・ハト派判定や ESG 情報の抽出など、判断材料の質的な向上を狙った応用が多く、これまでよりも微妙なニュアンスが要求される投資判断プロセスに付加価値を与える可能性がある。一方で、資産価格予測やリスクシナリオ生成といった、定量的予測タスクは少数にとどまる。第三に、業務の効率化を掲げる 29 件は、「分類」（20 件）と「生成」（8 件）が大半を占め、定型的業務の自動化が研究の焦点になっている。

最後に、「分類」は三目的すべてに貢献しうる汎用タスクである一方、構築と評価は性能評価目的に強く結びつき、予測は意思決定高度化に偏在する傾向が読み取れる。

3.5 手法およびモデル

金融における LLMs 活用の手法は、以下の 4 つの主要なアプローチに分類される。

ファインチューニング: LLMs を特定の金融タスクに最適化し、精度を向上させる手法。特に BERT や T5 などのモデルが用いられている。決算短信と業績発表記事の対応関係

を学習データとして T5 をファインチューニングし、決算短信から重要文を抽出して要約を生成。また、GPT-2 をファインチューニングして製品関係の文を生成し、部品関係のペアを抽出。BERT をファインチューニングして、サステナビリティレポートや統合報告書から環境活動に関連する文を分類。

プロンプト・エンジニアリング: プロンプトの工夫により、LLMs の応答の質を高め、タスクの遂行を効率化させる。少量の学習データから適切なプロンプトを LLMs に生成させる [57, 67]。プロンプトのデザインが結果に大きく影響を与える。プロンプトエンジニアリングを用いてニュース記事から異なる抽象度のトピックを抽出。

Zero-shot/Few-shot 学習: ゼロまたは少量のデータでモデルが新しいタスクに対応できる柔軟性を持たせる手法。これにより、データが不足している場合でも高精度の結果が得られる。

他のモデルとの組み合わせ: RAG、Word2Vec、クラスタリング手法などを LLMs と組み合わせることで、データ処理の効率化や精度向上を図る。

Table 6: 研究で使用された LLMs の集計

Model	**件数**
ChatGPT (GPT-3.5)	22
GPT-4	21
BERT	15
Llama	11
GPT-2	3
Gemini	4
Claude	3

表 6は研究で使用された LLMs の集計結果である。表から、金融関連のタスクにおいても ChatGPT が最も多く使用されていることがわかる。また、本稿執筆時点での GPT の最新のバージョンである GPT-4 も、ChatGPT に次いで多く使用されている。一方で、BERT モデルも多くのタスクで特にベンチマークあるいはベースラインとして使用されており、特に分類タスクでその効果が発揮されている。BERT をベースにしたファインチューニングの利用も見られる。GPT-2 は古いモデルであり、使用例は減っているものの、単純なテキスト生

成や文書の要約といったタスクで使われる。Llama や Gemini などは、比較的新しいモデルであり、GPT-4 や ChatGPT と組み合わせて、あるいは比較して使用されている。さらに、FinGPT の使用例も増えており、特に金融特化のタスクにおいて評価されている。

3.6 研究結果の考察

GPT-4 の性能

多くの研究で GPT-4 が最も高い性能を示しており、特にセンチメント分析や分類タスクにおいて、その精度は他のモデルを大きく上回っている。GPT-4 は、金融タスクにおいてもその汎用性が実証され、例えば、企業の環境活動に関する改善案の自動生成 [26, 29] や、株価変動記事に最適な株式用語を選択する [46, 35] 際に高い精度を発揮している。また、公認会計士試験の回答 [27, 56] などの分類タスクでも GPT-4 は合格点を大幅に超える結果を出しており、その能力の高さが評価されている。金融テキストマイニングで重要となる分類基準である、定性的表現 [25] あるいは主観的表現の抽出 [39] といった、従来の分類モデルでは分類が困難な問題に対しても、高い精度の分類が可能となっていることが実証されている。また、株式銘柄の選定におけるセンチメントスコアの分析 [40] では、高いセンチメントスコアを持つ銘柄が高リターンを示す傾向がある。

プロンプト設計の重要性

プロンプト設計の質が結果に与える影響が大きく、GPT-3.5 や GPT-4 を用いたタスクでは、プロンプト・エンジニアリングがタスクの精度に大きく影響を与える。例えば、ニュース記事の ESG トピック分析において、プロンプトの改良により抽象化が部分的に成功したものの、さらなる改良が必要とされている [21]。また、定性的表現 [25] あるいは主観的表現の抽出 [39] の場合には、プロンプトを英語で与えること、また、適切な例を与えること (Few-shot 学習) で精度が向上することが示されている。

生成タスクにおける精度

生成タスクにおいても、GPT-4 や GPT-3.5 が良好な結果を得ている。特に、市況コメントや投信ディスクロージャー資料の自動生成タスク [24, 38] では、生成されたテキストが人手で作成された文書と高い類似性を示すことを実証している。したがって、LLMs を有効に活

用することで金融レポート作成の効率化に寄与することを示唆している。また、ChatGPT や GPT-4 以外のモデルでは、特許マップの生成タスク [30] において、T5 と Word2Vec の組み合わせが、高精度な技術課題と解決手段の推定を可能にし、人手で作成された特許マップに匹敵する成果を達成している。

バイアスとリスク評価

金融投資における LLMs のバイアスについても研究が進められており、GPT-4 や PaLM2 を用いた金融投資意思決定バイアスの評価が行われている [41]。PaLM2 では特定の質問で年齢に基づく偏りが確認され、LLMs が金融投資意思決定に影響を与える可能性が示唆されている。さらに、企業固有のバイアスがセンチメント分析に及ぼす影響も検証されており、企業名を含むプロンプトではバイアスが定量化され、性能が高いモデルほどバイアスが少ない傾向が見られた [58]。

4 金融実務で LLMs を適用する際の考慮事項

本章では、サーベイ結果を踏まえて、金融機関や関連事業者が LLMs を金融実務に導入・運用する際に直面する主要な判断軸、課題をまとめる。

4.1 LLMs の必要性の判断

LLMs の導入を検討する際には、まず対象タスクの性質を踏まえて、その必要性を慎重に評価することが重要である。[87] によって述べられたように、LLMs の利点は以下の通り。

事前学習知識の活用: LLMs は事前学習データから得られた知識を活用できる。タスクに十分なトレーニングデータやアノテーションデータが不足しているが、常識的な知識が必要な場合、LLMs を使う利点がある。

推論能力: LLMs は推論や新たな能力を伴うタスクに優れている。これは、タスクの指示や期待される回答が明確でない場合や、分布外データを扱う場合に LLMs が適していることを意味する。

モデル間の調整能力: LLMs は異なるモデルやツール間を調整する役割を果たすことができる。複数のモデルの協調が必要なタスクにおいて、LLMs はこれらのツールを統合し、活用することができる。

LLMs はサードパーティの API を利用する場合や、オープンソースの LLMs を finetuning する場合には、相応のコストがかかる。そのため、タスクが明確に定義されており（例：回帰、分類、ランキング）、十分なアノテーション付きトレーニングデータが存在し、かつ上述の LLMs の利点である常識的な知識や推論といった能力に依存しない場合には、LLMs は必ずしも必要ではない。

本サーベイでは、分類タスクが 38 件、生成タスクが 11 件、構築タスクが 8 件、評価タスクが 5 件、予測タスクが 3 件、その他が 4 件の計 6 カテゴリに整理されており、中でも分類と生成の合計 49 件が全体の多くを占めている。この結果は、文書からの情報抽出や要約、自動生成といった LLMs の強みを活かせる領域で高い成果が報告されていることを示しており、同様のタスクにおいては LLMs を適用候補とする価値が大きい。

さらに、サーベイ結果を目的別に分類すると、業務効率化が 29 件、意思決定高度化が 21 件、性能評価が 19 件となっている。業務効率化の文脈では、反復的なデータ抽出やレポート作成の自動化といった生成・分類タスクで顕著な効果が挙げられており、定型データが十分に整備されている場合や処理速度、コスト削減が重視される状況では、LLMs 導入によるメリットが大きい。一方、意思決定高度化ではセンチメント分析や ESG 評価など、定性的情報の解釈や洞察獲得を要するケースが多く、LLMs の推論能力を活用して複雑な定性判断の質を向上させることが可能であると考えられる。

4.2 LLMs 導入の検討事項

まず、LLMs を導入するにあたっては、

1. 社内に GPU を搭載したサーバーを構築
2. AWS などのクラウドサービスを利用

のいずれかを選択することになる。前者のサーバー構築は、構築するだけでなく、その後の保守・運用も必須となるため、人的コストも見積もる必要がある。また、少なくとも開発環境と本番環境を分けるべきであり、本番環境のサーバーがダウンすることも考えられるため、バックアップとして複数のサーバーを構築する必要もある。

対して、後者のクラウドサービスを利用は、環境の保守・運用は不要である。ただし、クラウドサービスの利用には様々なリスクや課題が存在する。データセキュリティ（データ漏洩、不正アクセス）、データプライバシー、ネットワーク遅延、プロバイダーの障害によるサービス停止、ガバナンスとコンプライアンスなどがあげられる。それ以外にも、サービスの利用料が高額であることや、近年の GPU 需要により、契約や使用方法次第では、インスタンスがすぐに立ち上がらない、spot では短時間の利用でも途中にインスタンスが落ちるなど、何かと不便な点も多い。

加えて、法的・ライセンス上の検討事項が欠かせない。多くの LLM は研究利用や非商用利用は無償だが、商用利用には別途契約やライセンス料が発生することが多い。例えば、Meta が公開している LLMs である Llama3 は、月間アクティブユーザーが 7 億人以下の場合は無償で商用利用が可能である。さらに、継続事前学習しているモデルに関しては追加の注意が必要である。もともとの事前学習モデルの利用制約に加えて、継続事前学習に使用したデータの利用制約に加えて、独自に公開した組織が制約を付け加える可能性もある。

また、公開されている LLMs がどのような出力を返すのかに関しても、入念な確認作業が必要である。継続学習前のモデルでは不適切な語彙制御やバイアス低減のための安全フィルターが実装されていることが多いが、継続事前学習されたモデルでは必ずしも同様の制御が残っているとは限らない。それ以外にも、前章で紹介した通り、LLMs にはバイアスがあることが先行研究でも指摘されている。金融業務において、露骨に特定の銘柄を推薦する LLMs は、許容されないだろう。禁止ワードや不自然なセンチメントが検出された場合は再生成や人手によるチェックに回す仕組みを用意しておくことが肝要である。

4.3 LLMs の今後の展望

ここでは、LLMs の金融における今後について議論する。本サーベイの結果からは、現状で金融における LLMs の応用は進んでいるものの、その特性や性能限界は依然として十分に解明されていない。LLMs は将来的に汎用人工知能（AGI）への発展可能性を指摘する意見はあるものの、仮に AGI が実現した場合に金融実務がどのように変貌するのかは未だ具体的な議論を欠く。AGI の実現可否自体が不確実である以上、まずは現行の LLMs が延長線上で示す機能向上の範囲と、その実務適用における性能限界を体系的に把握する必要がある。特に日本市場特有の言語的・慣習的要件を満たすためには、英語圏向けモデルを単に翻訳するのではなく、日本語の金融文書や報告様式に最適化したローカライズが不可欠である。

また、LLMs の社会実装という観点では、すでに、「情報抽出機」としての役割は達成され始めている。3.1章でも確認した通り、現状では分類タスクのような問題が主流となっており、現在の社会実装の主流は情報抽出である。また、分類タスクの次点で多い生成タスクも、要約や情報整理の観点としての生成が主流であり、情報抽出機能としての役割が多い。より具体的には、ESG 評価のための文書分類やセンチメント分析、大量の契約書・報告書からのキーワード抽出・要約生成など、非構造化データを構造化情報へ変換するユースケースは報告されており、「情報抽出機」としての役割はすでに多く社会実装が始まり、情報抽出の文脈においては、LLMs が十分に活用されていると言える。しかし、この段階では依然として「何を取り出すか」に特化した適用が主流であり、取り出した情報をもとに LLMs 自身が最終判断や高度な意思決定を行う「情報出力機」としての役割は限定的に留まっている。

今後より重要になってくるのが、「情報出力機」としての LLMs の機能強化が鍵となるだろう。たとえば、アナリストレポートの自動生成や、決算短信・有価証券報告書・統合報告書などの自動生成、あるいはチャットボットによる投資家等の支援といった分野である。具体的には、アナリストレポートや決算短信の自動生成、決算短信・有価証券報告書・統合報告書などの自動作成、投資家向けチャットボットによる対話型意思決定支援など、生成したコンテンツの責任所在や品質保証を担保しつつ実務に組み込む技術的・運用的仕組みの確立が求められる。

このような分野は、生成した出力の責任性などの問題が実装を阻んでいるという問題が

あることに加え、LLMs のハルシネーションや出力の制御技術などの難点から金融分野での社会実装はまだまだ進んでいない。ハルシネーション防止や出力制御技術、生成物のトレーサビリティ確保、さらには金融ベンチマークや専門家フィードバックを組み込んだ RLHF の応用など、生成精度と信頼性を両立する枠組みの開発が急務である。そのため、今後の金融分野における LLMs の活用という観点では、「情報出力機」としての役割を促進するような技術の開発が進むことで、LLMs はこれまで以上に幅広い金融業務の自動化・高度化を実現し得ると期待される。

5　まとめ

本サーベイでは、日本の金融分野における LLMs の適用状況を調査し、2022 年 11 月から 2025 年 3 月末までに発表された 69 本の論文を分析した。これらの研究を、「分類」、「生成」、「予測」、「構築」、「評価」、「その他」の 6 つのタスクに分類し、それぞれの応用事例と課題を明らかにした。また、使用されたモデルや手法を記録し、金融分野特有のニーズと課題を抽出した。調査の結果、分類タスクが最も多く、投資判断やリスク管理に直結するセンチメント分析や ESG 評価などが注目されていることが分かった。次いで生成タスクでは、決算短信の要約や市況コメントの自動生成が多くの研究で扱われ、業務の効率化に寄与している。また、構築タスクにおいては、日本語金融特化型のデータセットやモデルが開発され、今後の研究基盤として期待されている。評価タスクでは、モデルのバイアスや性能が分析され、金融特有のタスクへの適応性が検証された。金融分野における LLMs の活用は、**「意思決定の高度化」**と**「業務の効率化・自動化」**という 2 つの大きな軸で進展しており、センチメント分析や市場動向予測、ESG 情報の抽出、自動文書生成など多岐にわたる応用が見られる。一方で、データ不足やモデルのバイアス、特定タスクへの最適化における課題が指摘されており、これらを克服するための研究が進行中である。本研究の結果は、金融分野における LLMs の適用可能性を包括的に理解するための重要な知見を提供するとともに、今後の研究や実務への示唆を与えるものである。

References

[1] Richard C Grinold and Ronald N Kahn. Active portfolio management. 2000.

[2] Richard O Michaud and Tongshu Ma. Efficient asset management: a practical guide to stock portfolio optimization and asset allocation., 2001.

[3] Anthony Neuberger. The black–scholes paper: a personal perspective. *Decisions in Economics and Finance*, 46(2):713–730, 2023.

[4] Rosalind Z Wiggins, Thomas Piontek, and Andrew Metrick. The lehman brothers bankruptcy a: overview. *Journal of financial crises*, 1(1):39–62, 2019.

[5] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

[6] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. Large language models in finance: A survey. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 374–382, 2023.

[7] 中川慧 and 伊藤友貴. 機械が読む英文開示. **企業会計**= *Accounting*, 75(3):339–348, 2023.

[8] Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, et al. Finben: A holistic financial benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:95716–95743, 2024.

[9] 坂地泰紀 and 中川慧. 金融・経済ドメインにおける言語処理の進展. **自然言語処理**, 31(2):763–768, 2024.

[10] Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance*, 1(2):223, 2001.

[11] Lu Cheng, Kush R Varshney, and Huan Liu. Socially responsible ai algorithms: Issues, purposes, and challenges. *Journal of Artificial Intelligence Research*, 71:1137–1181, 2021.

[12] Andreas Fuster, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther. Predictably unequal? the effects of machine learning on credit markets. *The Journal of Finance*, 77(1):5–47, 2022.

[13] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179, 2024.

[14] Kei Nakagawa, Masanori Hirano, and Yugo Fujimoto. Evaluating company-specific biases in financial sentiment analysis using large language models. In *2024 IEEE International Conference on Big Data (BigData)*, pages 6614–6623. IEEE, 2024.

[15] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*, 2024.

[17] 平野正徳. 言語モデル性能評価のための日本語金融ベンチマーク構築と各モデルのパフォーマンス動向. **人工知能学会第二種研究会資料**, 2023(FIN-032):28–35, 2024.

[18] 山口流星, 田代雄介, 鈴木彰人, and 辻晶弘. 金融ニュースのタグ付けにおける大規模言語モデルの有効性検証. **人工知能学会第二種研究会資料**, 2023(FIN-032):36–40, 2024.

[19] 濱田祐馬, 石野亜耶, and 中尾悠利子. 大規模言語モデルを活用した esg 評価. **人工知能学会第二種研究会資料**, 2023(FIN-032):45–52, 2024.

[20] 仲泰成, 酒井浩之, and 永並健吾. T5 を用いた決算短信の生成型要約. **人工知能学会第二種研究会資料**, 2023(FIN-031):32–35, 2023.

[21] 小杉樹来, 小澤誠一, 廣瀬勇秀, 池田佳弘, 中川憲保, 飯塚正昭, and 西田大輔. Chatgpt を用いたニュース記事の esg トピック分析. **人工知能学会第二種研究会資料**, 2023(FIN-031):36–41, 2023.

[22] 永並健吾 and 酒井浩之. 大規模言語モデルを用いたサプライチェーンマップの自動生成. **人工知能学会第二種研究会資料**, 2023(FIN-031):50–54, 2023.

[23] 指田昌樹, 和泉潔, and 坂地泰紀. Bert および chatgpt を用いたサステイナビリティレポートからの sdgs 関連文抽出. **人工知能学会第二種研究会資料**, 2023(FIN-031):55–60, 2023.

[24] 高野海斗, 中川慧, and 藤本悠吾. Chatgpt を活用した運用報告書の市況コメントの自動生成. **人工知能学会第二種研究会資料**, 2023(FIN-031):61–67, 2023.

[25] 黒木裕鷹 and 中川慧. 決算説明会テキストデータに含まれる主観的表現の抽出とその使用傾向の分析. **人工知能学会第二種研究会資料**, 2023(FIN-031):68–74, 2023.

[26] 児玉実優, 酒井浩之, 永並健吾, 高野海斗, and 中川慧. 企業における環境活動の改善案の自動生成. **人工知能学会第二種研究会資料**, 2023(FIN-031):75–80, 2023.

[27] 増田樹, 中川慧, and 星野崇宏. Chatgpt は公認会計士試験を突破できるか?: 短答式試験監査論への挑戦. **人工知能学会第二種研究会資料**, 2023(FIN-031):81–88, 2023.

[28] 澤木智史 and 仲山泰弘. ゼロショットテキスト分類を活用した含意判定モデルによる frb 金融政策コミュニケーションの読解. **人工知能学会第二種研究会資料**, 2023(FIN-030):72–77, 2023.

[29] 児玉 実優, 酒井浩之, 永並 健吾, 高野 海斗, and 中川慧. 企業の環境活動における収益性の関係解析と改善案の自動生成. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[30] 小堀 佑樹, 酒井 浩之, and 永並 健吾. T5 を用いた技術課題・解決手段推定による特許マップ自動生成. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[31] 土井 惟成, 小田 悠介, 中久保 菜穂, and 杉本淳. ゼロショットテキスト分類による tcfd 推奨開示項目の自動判定. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[32] 伊藤 友貴 and 平松 賢士. 有価証券報告書の活用による事業セグメント関連語の拡張. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[33] 高野 海斗 and 中川 慧. 大規模言語モデルを用いた金融テキストに対する推論ベースの極性付与. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[34] 川原一修. Llm を用いたタカハトセンチメント付与タスクの検証. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[35] 西田 隼輔 and 宇津呂 武仁. 株価変動に対する大規模言語モデルを用いた株式用語選択. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[36] 平野 正徳. 金融分野における言語モデル性能評価のための日本語金融ベンチマーク構築. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[37] 西脇 一尊, 大沼 俊輔, 工藤 剛, and 門脇一 真. ファイナンシャル・プランニングの自動化に向けた gpt-4 及び rag の性能評価. In **言語処理学会第** *30* **回年次大会** *(NLP2024)*. 一般社団法人 言語処理学会, 2024.

[38] 高野海斗, 中川慧, and 藤本悠吾. 大規模言語モデルによる投信ディスクロージャー資料の市況および見通しコメントの自動生成. **人工知能学会論文誌**, 39(4):FIN23–B_1, 2024.

[39] 屋嘉比潔, 黒木裕鷹, and 中川慧. Md&a における定性的表現と経営者予想の精度. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 2G6GS601–2G6GS601. 一般社団法人 人工知能学会, 2024.

[40] 鈴木雅弘. 会社四季報のセンチメントを用いた株式銘柄選定の試み. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 2I6GS1001–2I6GS1001. 一般社団法人 人工知能学会, 2024.

[41] 立花竜一, 中川慧, 伊藤友貴, and 高野海斗. 大規模言語モデルの金融投資意思決定バイアスに関する評価指標の構築. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 3Xin236–3Xin236. 一般社団法人 人工知能学会, 2024.

[42] 山田正嗣 and 井本稔也. 金融ドメイン特化のための大規模言語モデルのインストラクションチューニング評価. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 3Xin253–3Xin253. 一般社団法人 人工知能学会, 2024.

[43] 伊藤克哉 and 中川慧. Llm-traders: 大規模言語モデルを用いた金融時系列予測. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 3Xin283–3Xin283. 一般社団法人 人工知能学会, 2024.

[44] 徳武悠 and 齋藤悠司. 家計簿データに基づく任意の生活スタイルに対応したユーザ抽出手法の検討. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 3Xin299–3Xin299. 一般社団法人 人工知能学会, 2024.

[45] 高野海斗 and 中川慧. 大規模言語モデルによる事業概要を考慮した金融テキストの推論ベース極性分析. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 4Xin211–4Xin211. 一般社団法人 人工知能学会, 2024.

[46] 西田 隼輔 and 宇津呂 武仁. 株価変動画像に対する大規模マルチモーダルモデルを用いた株式用語選択. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 3M5OS12b02–3M5OS12b02. 一般社団法人 人工知能学会, 2024.

[47] 大坪 悠介. 大規模言語モデルを用いた事業文書から esg 指標への影響推論及び統計的因果推論との関係の検証. In **人工知能学会全国大会論文集 第** *37* **回** *(2023)*, pages 3Xin429–3Xin429. 一般社団法人 人工知能学会, 2023.

[48] 高野海斗, 中川 慧, and 酒井浩之. 業績文の分析を目的とした文中の区切り位置推定. In **第** *20* **回テキストアナリティクス・シンポジウム**, pages 69–74. 一般社団法人 電子情報通信学会, 2023.

[49] 山本昂, 篠崎玲菜, 篠原滉, and 毛利研. サステナビリティ・トランスフォーメーションに向けたマクロ環境モニタリングに基づくマテリアリティ分析. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 1O3GS1105–1O3GS1105. 一般社団法人 人工知能学会, 2024.

[50] 梅原武志 and 武田英明. 大規模言語モデルを利用したパンデミック期の事業等のリスクの記述分析. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 2I6GS1004–2I6GS1004. 一般社団法人 人工知能学会, 2024.

[51] 鈴木雅弘, 坂地泰紀, 平野正徳, and 和泉潔. Findebertav2: 単語分割フリーな金融事前学習言語モデル. **人工知能学会論文誌**, 39(4):FIN23–G_1, 2024.

[52] 鈴木彰人, 田代雄介, 辻晶弘, and 山口流星. 不動産価値推定における大規模言語モデルの活用可能性に関する検証. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 2I5GS1002–2I5GS1002. 一般社団法人 人工知能学会, 2024.

[53] 伊藤央峻. Llama2 を用いたカーボンプライシング関連論文の自動分類. In **人工知能学会全国大会論文集 第** *38* **回** *(2024)*, pages 3Xin286–3Xin286. 一般社団法人 人工知能学会, 2024.

[54] 伊藤央峻. Bert を用いたカーボンプライシング関連論文の分析. **人工知能学会第二種研究会資料**, 2023(FIN-031):89–93, 2023.

[55] 稲葉達也. 自然言語処理技術を用いた人的資本情報の抽出. **経営情報学会 全国研究発表大会要旨集**, 202311:253–256, 2024.

[56] 増田 樹, 中川 慧, 高野 海斗, and 星野 崇宏. Chatgpt による公認会計士短答式試験（企業法）のパフォーマンス分析. In **第 *21* 回テキストアナリティクス・シンポジウム**, pages 27–32. 一般社団法人 電子情報通信学会, 2024.

[57] 高野 海斗, 中川 慧, and 藤本 悠吾. 大規模言語モデルを用いたテキストの二値分類における定義文自動生成. In **第 *21* 回テキストアナリティクス・シンポジウム**, pages 75–80. 一般社団法人 電子情報通信学会, 2024.

[58] 中川 慧, 平野 正徳, and 藤本 悠吾. 大規模言語モデルを活用した金融センチメント分析における企業固有バイアスの評価. In **第 *21* 回テキストアナリティクス・シンポジウム**, pages 81–86. 一般社団法人 電子情報通信学会, 2024.

[59] 田邉耕太, 鈴木雅弘, 坂地泰紀, and 野田五十樹. Jafin：日本語金融インストラクションデータセット. In **第 *21* 回テキストアナリティクス・シンポジウム**, pages 87–92. 一般社団法人 電子情報通信学会, 2024.

[60] 佐藤栄作 and 木村泰知. 有価証券報告書を対象とした質問応答タスクのデータセット構築と llm を用いた手法の評価. In **第 *21* 回テキストアナリティクス・シンポジウム**, pages 93–98. 一般社団法人 電子情報通信学会, 2024.

[61] 鹿子木 亨紀 and 森山 治紀. Llm を用いた統合報告書からの esg 情報抽出. **人工知能学会第二種研究会資料**, 2024(FIN-033):48–52, 2024.

[62] 増田 樹, 中川 慧, and 星野 崇宏. 会計基準グラフを用いた質問応答モデルの構築 収益認識基準を用いた実験. **人工知能学会第二種研究会資料**, 2024(FIN-033):53–60, 2024.

[63] 屋嘉比 潔, 黒木 裕鷹, and 中川 慧. 日本企業データを用いた機械学習による利益変化の予測. **人工知能学会第二種研究会資料**, 2024(FIN-033):68–75, 2024.

[64] 酒井 浩之, 永並 健吾, 木村 賢二郎, 寺口 舞紘, 大江 いづみ, and 中島 泰暉. 企業 web サイトからの esg の方針・取り組みに関する情報の抽出. **人工知能学会第二種研究会資料**, 2024(FIN-033):41–47, 2024.

[65] 平野 正徳, 今城 健太郎, 齋藤 俊太, 岡田 真太郎, 的矢 知樹, 谷口 徹, and 太田 佳敬. 金融特化大規模言語モデルの構築と検証. **人工知能学会第二種研究会資料**, 2024(FIN-033):142–149, 2024.

[66] 山口 流星, 田代 雄介, 鈴木 彰人, 辻 晶弘, 亀田 希夕, and 宮澤 朋也. 進化的モデルマージを用いた日本語金融 llm モデルの構築. **人工知能学会第二種研究会資料**, 2024(FIN-033):150–154, 2024.

[67] 高野 海斗, 中川 慧, and 藤本 悠吾. 大規模言語モデルを用いた金融テキスト二値分類タスクの定義文生成とチューニング手法の提案. **人工知能学会第二種研究会資料**, 2024(FIN-033):155–162, 2024.

[68] 悠太 平松, 泰弘 小川, and 勝彦 外山. 決算短信における見通し文と結果文のアラインメント. In **言語処理学会第 *31* 回年次大会**, 2025.

[69] 悠利子 中尾, 亜耶 石野, 克彦 國部, and フィリップ 須貝. バリューモデルを活用したサステナビリティ情報抽出. In **言語処理学会第 *31* 回年次大会**, 2025.

[70] 張引 司龍, 小天 王, and 武仁 宇津呂. 大規模言語モデルを用いた有価証券報告書の表質問応答. In **言語処理学会第 *31* 回年次大会**, 2025.

[71] 正徳 平野 and 健太郎 今城. 金融分野に特化した複数ターン日本語生成ベンチマークの構築. In **言語処理学会第 *31* 回年次大会**, 2025.

[72] 惟成 土井 and 麻由梨 田中. 大規模言語モデルを用いた few-shot プロンプティングによる jreit の投資物件に関する表構造認識. In **言語処理学会第 *31* 回年次大会**, 2025.

[73] 隼輔 西田 and 武仁 宇津呂. 株価変動要因情報を手掛かりとする株価変動記事生成への llm の適用. In **言語処理学会第 *31* 回年次大会**, 2025.

[74] 頌平 飯田, 槙山 古俣, 聖 三田寺, 遼 長谷川, 宇津呂, 武仁 林, 超 友, and 里絵 宍戸. 会計ドメインにおける質問応答のための llm を用いた解説ページの順位付け. In **言語処理学会第 *31* 回年次大会**, 2025.

[75] 大 山脇, 賢也 野中, 光太郎 田村, 海斗 高野, and 慧 中川. 取締役推薦理由文を用いた取締役のスキルマトリックス分類モデルの開発. In **言語処理学会第 *31* 回年次大会**, 2025.

[76] 海斗 高野. 金融テキストにおけるセンチメント分析の課題整理. In **言語処理学会第 *31* 回年次大会**, 2025.

[77] 平野 正徳. Llm ベースのエージェントによる人工市場シミュレーションの構築. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[78] 安田 卓矢, 村山 友理, and 和泉 潔. 経済フェルミ推定問題: 因果構造推論による推定精度の改善. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[79] 高野 海斗. 金融テキストを対象とした強弱を捉えることができるセンチメントモデル開発のためのデータセット構築方法の検討および分析. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[80] 鈴木 雅弘 and 坂地 泰紀. 景気ウォッチャー調査のデータセット構築と物価センチメント分析. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[81] 竹下 蒼空 and 酒井 浩之. 決算短信を用いた業種ごとの年間レポートの自動生成. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[82] 島津 雛子, 濱崎 良介, 大塚 浩, and 飯島 泰裕. Llm を用いた情報系企業の esg 評価についての一考察. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[83] 高橋 明久 and 戸辺 義人. Llm を用いた大量保有報告書における担保契約等重要な契約に関する情報の構造化. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[84] 矢野 一樹, 平野 正徳, and 今城 健太郎. 多様なテンプレートと合成データを用いた大規模言語モデルの業種区分予測における知識抽出. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[85] 大堀 遼介. 有価証券報告書と統合報告書を活用した esg 投資のためのオントロジー構築と生成 ai による情報抽出. **人工知能学会第二種研究会資料**, 2025(FIN-034), 2025.

[86] Agam Shah, Suvan Paturi, and Sudheer Chava. Trillion dollar words: A new financial dataset, task & market analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6664–6679, 2023.

[87] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[88] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2, 2019.

[89] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[90] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[91] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

[92] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[93] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[94] V Sanh. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[95] Z Lan. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[96] 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)*, pages 3982–3992, 2019.

[97] Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. PromptBERT: Improving BERT sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, 2022.

[98] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[99] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[100] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[101] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

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

[103] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[104] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[105] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[106] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*, 2023.

[107] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[108] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[109] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a Japanese tokenizer for business. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi,

Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[110] 松田 寛. Ginza - universal dependencies による実用的日本語解析. **自然言語処理**, 27(3):695–701, 2020.

[111] 鈴木 雅弘, 坂地 泰紀, 平野 正徳, and 和泉 潔. Findebertav2: 単語分割フリーな金融事前学習言語モデル. **人工知能学会論文誌**, 39(4):FIN23–G_1–14, 2024.

[112] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[113] Mitchell Wortsman, Peter J Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, et al. Small-scale proxies for large-scale transformer training instabilities. *arXiv preprint arXiv:2309.14322*, 2023.

[114] Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models. *arXiv preprint arXiv:2312.16903*, 2023.

[115] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[116] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: solution for edge computing applications*, pages 87–104, 2021.

[117] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`, 2022.

[118] Penghui Qi, Xinyi Wan, Guangxing Huang, and Min Lin. Zero bubble pipeline parallelism. *arXiv preprint arXiv:2401.10241*, 2023.

[119] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.

[120] Masahiro Suzuki and Hiroki Sakaji. Language Model Construction and Domain Adaptation using Multiple Nodes. In *Intelligent Computing Systems (ICS)*, volume 213, pages 1–6, 2024.

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

[122] Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, et al. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. *arXiv preprint arXiv:2407.03963*, 2024.

[123] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[124] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[125] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[126] 聡 関根, まや 安藤, 美知子 後藤, 久美 鈴木, 大輔 河原, 直也 井之上, and 健太郎 乾. ichikara-instruction: Llm のための日本語インストラクションデータの構築. In **言語処理学会第** *30* **回年次大会**, 2024.

[127] Masanori HIRANO, Masahiro SUZUKI, and Hiroki SAKAJI. llm-japanese-dataset v0: Construction of Japanese Chat Dataset for Large Language Models and its Methodology. In *The 26th International Conference on Network-Based Information Systems*, pages 442–454, 2023.

[128] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[129] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[130] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[131] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[132] 尚人 水野, 利彦 柳瀬, and 正太郎 佐野. 自動プロンプト最適化のソフトウェア設計. In **言語処理学会第 *30* 回年次大会**, pages 2537–2542, 2024.

[133] XTX Investments. Ai mathematical olympiad - progress prize 1, 2024.

A　LLMsの技術的背景

A.1　トークン化

LLMsが文章を処理するにあたっては、文章を語彙に当てはめてID化する必要性がある。一般にLLMsでは、あらかじめ語彙を作成し、その語彙を文章の処理に使用する。この操作を「トークン化 (トークナイゼーション)」と呼ぶ。また、この処理を実施するモジュールを「トークナイザー」と呼ぶ。

トークン化に必要な語彙には、高頻度に出現する単語を含める必要がある。しかし、低頻度の語彙をどの程度収録するかについては慎重な検討が求められる。低頻度の語彙を多く収録すると、学習時にそれらの語彙が十分に学習されない可能性がある。一方で、適度に低頻度の語彙を収録することで、文章をトークン化した際のトークン数が減り、文章が短く扱われるため、その後の計算効率が向上するという利点もある。このようなトレードオフの関係の中、どの程度語彙を収録するかについては、語彙数をハイパーパラメータとして設定する必要がある。

この語彙分割の方法の主要な手法として、バイト符号化 (BPE; Byte-pair encoding) やWordpieceなどが存在する。後述するBERTモデル [88] においては、Wordpieceが主流であったが、LLMsにおいてはBPEが頻繁に使用される。BPEは、頻繁に出現する文字や文字列のペアを反復的に結合し、効率的な語彙セットを構築する。ただし、通常のBPEでは、あらかじめ定義した語彙に含まれていない文字や文字列が入力に登場した場合、未知語として扱われる可能性がある。この点、LLMsのBPEにおいては、byte fallbackという機能を使用することが一般である。語彙の中に、UTF-8で表現可能なすべての1バイト (\x00〜\xff) を個別のトークンとして含める。これにより、任意の文字列をバイト単位で分解してトークン化できる[12]。この処理により、UTF-8表記可能なすべての文字は、ほかの語彙に含まれていないとしても未知語とはならないため、すべての文章を未知語なしにエンコード・デコード可能となる。

一方で、BPEのbyte fallbackを利用しているがゆえに、語彙の扱いが乱雑になってしまっている場合がある。例えば、後述のGPT-3.5 (ChatGPT[5]) の場合、トークナイザーは日本語に完全には対応していない。GPT-3.5のトークナイザーで、

> デリバティブには様々な種類があり、先物取引、先物オプション取引、スワップ取引などがあります。

という文章をトークン化した場合、

> デ / リ / バ / テ / ィ / ブ / に / は / \xe6 / \xa7 / \x98 / \xe3\x80 / \x85 / な / \xe7\xa8 / \xae / \xe9\xa1 / \x9e / が / あり / 、 / 先 / 物 / 取 / 引 / 、 / 先 / 物 / オ / プ / シ / ョ / ン / 取 / 引 / 、 / ス / \xe3\x83 / \xaf / ッ / プ / 取 / 引 / な / ど / が / あり / ます / 。 (49 tokens)

というように分割される。一方で、GPT-4oのトークナイザーでトークン化した場合、

[12]たとえば、語彙に含まれていない特殊文字「漢」が入力されたとする。UTF-8エンコードで「漢」は3バイト \xE6\xBC\xA2 に分解される。Byte fallback機能により、これらのバイト列はそれぞれ語彙内の既存トークン (\xE6,\xBC,\xA2) としてマッピングされる。その結果、「漢」は未知語とされることなくエンコード可能である。また、デコード時には、このバイト列を再結合することで、元の文字列「漢」を復元できる。

デ / リ / バ / ティ / ブ / には / 様 /々 / な / 種類 / が / あり / 、 / 先 / 物 / 取 / 引 / 、 / 先 / 物 / オ / プ / ション / 取 / 引 / 、 / ス / ワ / ップ / 取 / 引 / など / があります / 。 (34 tokens)

とかなり効率が上がっている。これは、GPT-3.5 のトークナイザーの語彙数が約 10 万であるのに対し、GPT-4o のトークナイザーの語彙数が約 20 万に拡張されており、英語以外の言語に対する語彙のカバー率が大幅に高まったことによるものであると推測される。一方で、日本語からスクラッチで作成された PLaMo-100B のトークナイザーでトークン化した場合、

デリ / バ / ティブ / に / は / 様々 / な / 種類 / が / あり / 、 / 先物 / 取引 / 、 / 先物 / オプション / 取引 / 、 / スワ / ップ / 取引 / など / が / あり / ます / 。 (26 tokens)

のように分割され、語彙数が約 5 万しかないにも関わらず、日本語のトークン効率が圧倒的に良いことがわかる。

最終的には、モデルの生成性能の良さでしか語彙の良さは評価できないものの、LLMs のモデルパラメータ規模が同程度であるとされる PLaMo-100B と GPT-3.5 で比較した場合には、PLaMo-100B のほうが金融ベンチマーク [17] 性能が良いため、PLaMo-100B のほうがトークン効率もよく、性能もよいという結論になる。使用されている技術やコーパスも異なるため、厳密な直接比較はできないものの、日本語に特化してトークナイザーから構築することは重要であると言える。

A.2 言語モデル

言語モデル (Language Model) とは、テキスト内の単語や文の出現確率を学習し、新しいテキストの生成や既存のテキストの評価を行うためのモデルである。具体的には、与えられた単語列に基づいて次に来る単語や文の確率を予測することを目的とする。これにより、言語モデルは統計的な手法を用いてテキスト生成、解析、分類などのタスクを実現する。

言語モデルの基本的な形式は、ある文中での単語の出現確率を表現するものである。例えば、単語列 $W = w_1, w_2, \ldots, w_T$ から構成される文全体の確率は次のように定義される：

$$P(W) = P(w_1, w_2, \ldots, w_T) \tag{1}$$

この確率をチェーンルールを用いて展開すると、

$$\begin{aligned} P(W) = P(w_1) \times P(w_2|w_1) \times P(w_3|w_1, w_2) \\ \cdots \times P(w_T|w_1, w_2, \ldots, w_{T-1}) \end{aligned} \tag{2}$$

と書ける。この式は、文全体の確率を、各単語がその前までに出現した単語に依存した確率の積で表せることを示している。

一般に、言語モデルでは、次の単語 w_t がそれまでの $t-1$ 個の単語 $w_1, w_2, \ldots, w_{t-1}$ に依存する確率

$$P(w_t|w_1, w_2, \ldots, w_{t-1}) \tag{3}$$

を学習することを目的としている。この確率を学習することで、文脈に基づいて適切な単語を予測する能力を獲得する。

A.3　言語モデルの普遍性

言語モデルは、文脈を基に意味を理解し適切な出力を生成する能力を持つ。この特性により、テキスト生成や要約、質問応答、感情分析といった幅広い自然言語処理タスクに応用できる。言語モデルが十分に学習されている場合、次に続く内容を予測する能力が高まり、さまざまなタスクを統一的に解決することができる。この汎用的な特性は、言語モデルの「普遍性」ともいえる。以下では、具体的なタスクとしてテキスト生成、要約、質問応答、感情分析の4つを例に挙げ、言語モデルの適用可能性を説明する。

A.3.1　テキスト生成

テキスト生成タスクでは、言語モデルは既存の文脈に基づいて次に続く単語 (文) を生成できる。言語モデルが完全に学習されていると、最初の単語を与えることで、次に来る単語 w_t の確率 $P(w_t|w_1, w_2, \ldots, w_{t-1})$ に基づいて次の単語を生成することができる。このプロセスを繰り返すことで、文全体が構成される。

例えば、テキスト「今日は天気が」と与えられたとき、モデルは「良い」という単語が高い確率で続くと予測し、次に続く単語や文を生成する。この繰り返しによって新しいテキストが生成される。

$$P(\text{良い} \mid \text{今日は}, \text{天気が}) > P(\text{悪い} \mid \text{今日は}, \text{天気が}) \tag{4}$$

A.3.2　テキスト要約

テキスト要約では、言語モデルは長い文書の要約を作成するために、重要な情報を抽出し、簡潔な形で表現する。具体的には、入力文 $W = \{w_1, w_2, \ldots, w_T\}$ のうち、要約に必要な単語や文を選び出し、その確率が高い部分を要約文 S として出力する。

例えば、元の文が「この映画は素晴らしい映像と感動的なストーリーで評価されている。」であれば、言語モデルは各単語の重要度を評価し、要約にふさわしいキーフレーズ「素晴らしい映像」や「感動的なストーリー」の確率を高く見積もる。

これを言語モデルの形式で表すと、各キーフレーズの重要度を次のように確率で表すことができる。

$$P(\text{素晴らしい映像} \mid \text{この映画は}) > P(\text{他のキーフレーズ} \mid \text{この映画は}) \tag{5}$$
$$P(\text{感動的なストーリー} \mid \text{この映画は}) > P(\text{他のキーフレーズ} \mid \text{この映画は}) \tag{6}$$

完全な言語モデルがあれば、このように高い確率を持つキーフレーズを選び出し、それらを組み合わせて要約を生成することができる。

A.3.3　質問応答

質問応答タスクでは、言語モデルは与えられた質問に対する適切な回答を見つけ出す。言語モデルは、テキスト内のどの部分が質問に関連するかを特定し、その確率が高い部分を回答として生成する。

例えば、テキスト「太陽は東から昇る」とし、質問が「太陽はどこから昇りますか？」であれば、言語モデルは質問に対して最も関連性の高い単語を選択する。この場合、モデルは

「東」という単語の確率が最も高いと判断し、それを回答として出力する。

$$\begin{aligned} &P(\text{東} \mid \text{太陽はどこから昇りますか？}) \\ &> P(\text{他の選択肢} \mid \text{太陽はどこから昇りますか？}) \end{aligned} \tag{7}$$

言語モデルは、このように与えられた文脈と質問に基づいて、最も適切な回答を確率的に選び出し、それを出力する。

A.3.4 センチメント分析

センチメント分析では、言語モデルはテキストの感情ラベル（ポジティブ、ネガティブ、中立など）を分類する。例えば、レビュー「この製品は非常に良いです」に対して、モデルは「非常に良い」というフレーズがポジティブな感情を表していると判断し、テキスト全体をポジティブとして分類する。

$$P(\text{ポジティブ} \mid \text{この}, \text{製品}, \text{は}, \text{非常に良い}) > P(\text{ネガティブ} \mid \text{この}, \text{製品}, \text{は}, \text{非常に良い}) \tag{8}$$

A.4 言語モデルの発展

言語モデルは、古典的手法から始まり、現在では深層学習を基盤とした高度なモデルへと大きく進化してきた。初期の言語モデルである n-gram モデルは、テキストの短い文脈を捉えることに特化していたが、長期的な文脈や複雑な依存関係を表現する能力には限界があった。これを克服するために、Recurrent Neural Network（RNN[89]）や Long Short-Term Memory（LSTM[90]）が開発され、長期的な依存関係を学習する手法が提案された。さらに、Transformer モデル [91] の登場により、並列処理の効率化と長距離依存関係の学習能力が飛躍的に向上し、LLMs の発展を支える基盤が確立された。

A.4.1 n-gram モデル

n-gram モデルは、ある単語がそれ以前の $n-1$ 個の単語にのみ依存するというマルコフ性を仮定する。この仮定に基づき、(3) 式は次のように簡略化される。

$$P(w_t|w_1, w_2, \ldots, w_{t-1}) = P(w_t|w_{t-n+1}, \ldots, w_{t-1}) \tag{9}$$

例えば、$n = 2$ の場合、bi-gram モデルとなり次式で表される。

$$P(W) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_2) \cdots P(w_T|w_{T-1}) \tag{10}$$

このモデルは計算が効率的である一方で、文脈の範囲が限られており、長期的な依存関係を捉えることができないという欠点がある。また、もう一つの課題として、データのスパースネスの問題が挙げられる。データのスパースネスとは、特定の n-gram がトレーニングデータ中でほとんど出現しない、または全く出現しないため、モデルが正確な予測を行うことが難しくなる現象である。例えば、金融分野の文書に「このモデルを適用した後、リターンが悪化した」というフレーズが非常に少ない場合、n-gram モデルはこのフレーズの出現頻度が低いために学習できず、「このモデルを適用した後、」の言葉の予測の精度が大幅に低下す

る。さらに、文脈を考慮するために n の値を大きくすると、スパースネスの問題はさらに深刻化する。これは、可能な単語の組み合わせが指数関数的に増加するためである。例えば、単語の語彙サイズが V である場合、n-gram の可能な組み合わせは V^n となる。これにより、トレーニングデータに存在しない組み合わせが多くなり、モデルはこれらの未学習の n-gram に対して推定を行うことができず、予測が著しく不安定になる。これらの欠点を克服するために、以下の深層学習に基づくモデルが導入された。

A.4.2 RNN モデル、LSTM モデル

RNN は系列データを時刻方向に再帰的に処理し、隠れ状態 h_t を更新しながら次単語の条件付き確率 $P(w_t|w_{<t})$ を推定する。

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t). \tag{11}$$

RNN は n-gram の文脈長制限を克服できる一方、長系列では勾配が消失して遠過去の情報が学習しにくいという勾配消失問題が生じる。特に、RNN でテキストデータを扱う場合、文章を文字列や単語の系列データとして扱うため、文章が長くなるほどこの問題は顕著になる。

LSTM は勾配消失を防ぐため，セル状態 C_t と 3 つのゲート（忘却 f_t，入力 i_t，出力 o_t）を導入した拡張 RNN である。

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t]),\ i_t = \sigma(W_i[h_{t-1}, x_t]), \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_C[h_{t-1}, x_t]), \\ h_t &= o_t \odot \tanh(C_t). \end{aligned} \tag{12}$$

セル状態が長期依存を保持し、RNN より長文脈に強い。ただし逐次計算ゆえ大規模データでは計算負荷が高く、極端に長い依存には限界が残る。これを解消する構造として Transformer が登場した。

A.4.3 Transformer

Transformer モデルは、RNN や LSTM とは異なり、逐次処理の制約を排除し、自己注意機構（Self-Attention Mechanism）を用いて入力データを並列に処理する [91]。この特性により、Transformer は長距離依存関係を効率的に捉えることができると同時に、計算効率を大幅に向上させることに成功している。

Transformer の核心となる自己注意機構（Self-Attention）は、入力シーケンス内の単語間の関係性を動的に学習し、各単語が他の単語にどれだけ依存するかを表現する。自己注意機構により、文脈の長距離依存を捉えることが可能となり、複雑な文法構造や意味的なつながりを効果的に学習する。

自己注意 $\mathrm{Attention}(Q, K, V)$ は次のように定義される。

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{13}$$

ここで、Q、K、V はそれぞれクエリ、キー、バリューの行列を表し、d_k はキーの次元数である。$QK^\top$ を計算することで各単語間の関連度が求められ、softmax 関数で正規化することにより、特定の単語が文脈内でどれほど重要かを定量化する。さらに、Transformer では

マルチヘッド注意（Multi-Head Attention）が導入されており、複数の異なる視点から文脈を捉えることができる。これにより、文法的構造と意味的な関係の両方を同時に考慮することができる。

Transformer の全体構造はエンコーダとデコーダの 2 つの部分から成り立っている。エンコーダは入力シーケンスを処理しその特徴量を学習し、デコーダはエンコーダの出力を元に新しいシーケンスを生成する。各層には自己注意機構とフィードフォワード・ニューラルネットワークが含まれており、正規化と残差接続が適用されている。

Transformer の利点として、長距離依存関係を効率的に学習できる点、並列計算が可能な点、そして多様な自然言語処理タスクに応用可能な点が挙げられる。例えば、自己注意機構により、シーケンス内のすべての単語間の依存関係を同時に評価できるため、RNN や LSTM のように逐次的に情報を処理する必要がなくなる。また、並列計算の実現により、従来モデルと比較して学習速度が大幅に向上している。

Transformer の最も重要な点は、モデルのスケーリング則（Scaling Laws）に基づく設計と応用である。このスケーリング則は、モデルサイズ、データ量、計算量の増加がモデルの性能向上にどのように寄与するかを定量的に示すものである [92]。すなわち、Transformer の性能は以下の式に従い、パラメータ数 N、トレーニングデータ量 D、および計算量 C が増加するにつれて予測誤差が減少することが示されている。特に、パラメータ数の増加に伴う性能向上がデータ量の増加や計算リソースの増加と調和して進み、その性能の上限は未だに発見されていないことが、この則の意義である。

$$L \propto N^{-\alpha_N} + D^{-\alpha_D} + C^{-\alpha_C} \tag{14}$$

ここで、L はモデルの損失、$\alpha_N, \alpha_D, \alpha_C$ はそれぞれパラメータ、データ、計算量に対するスケール指数である。この式は、モデル性能の改善が主に 3 つの要因（モデルサイズ、データ、計算量）のバランスによることを示している。このスケーリング則の知見に基づき、近年の大規模言語モデル（LLMs）は数十億から数兆のパラメータを持つように設計されている。

さらに、このスケーリング則は単なる学習効率の最適化に留まらず、創発的能力の出現（Emergent Abilities）とも関連する。たとえば、巨大なモデルサイズと豊富なデータで訓練されたモデルでは、ゼロショット学習や少数ショット学習といった特性が自然に発現することが確認されている [87]。これにより、Transformer を基盤とした LLMs は、従来の特化型モデルが対応できなかった汎用的で柔軟なタスク処理が可能になっている。

A.5 LLMs の種類

大規模言語モデル（LLMs）は、その設計理念や学習目的に応じて大きく BERT 型と GPT 型の 2 つに分類される。これらのモデルは、それぞれ独自の特性を持ち、特定のタスクにおいて効果的に応用されている。

BERT 型モデルは双方向の文脈理解を特徴とし、分類等の識別タスクに優れた性能を発揮する。一方、GPT 型モデルは自己回帰型の生成アプローチを採用しており、テキスト生成や対話応答といった生成タスクにおいて高い汎用性を示す。これらの違いは事前学習目的、モデルの設計、そしてタスクへの適用方法に反映されている。

以下の表 7に、BERT 型と GPT 型の主な特徴を比較して示す。

表 7から、BERT 型モデルは双方向的な文脈を取り入れるため、文や単語間の関係を深く理解することに特化している。具体的には、BERT はテキストの一部をマスクし、その部分を予測するタスク（Masked Language Modeling, MLM）を通じて学習する。この方法によ

種類	代表的なモデル	事前学習目的	主な利用方法
BERT 型	BERT, RoBERTa, DistilBERT	マスクされた単語の予測	ファインチューニング
GPT 型	GPT-2, GPT-3, GPT-4	次単語の予測	プロンプト

Table 7: BERT 型と GPT 型の比較

り、テキスト全体の構造を効果的に把握することが可能となる。一方で、BERT 型モデルは生成能力が制限されているため、主に識別タスクに利用される。そして、主にファインチューニングによって様々な言語処理タスクに活用される。

GPT 型モデルは、異なるアプローチを採用している。これらのモデルは自己回帰型アーキテクチャを基盤とし、テキストの文脈を順次処理しながら次単語を予測する（Autoregressive Language Modeling）。この設計は、自由なテキスト生成や対話タスクにおいて優れた性能を発揮する。たとえば、GPT-3 や GPT-4 は少量の入力例（Few-shot Learning）やプロンプト設計により、多様なタスクに適応可能な汎用性を備えている。

BERT 型と GPT 型は、いずれも様々な自然言語処理タスクの基盤技術として広く活用されている。

A.5.1 BERT 型

BERT (Bidirectional Encoder Representations from Transformers) は、2018 年に Google AI によって発表され、自然言語処理の様々なタスクを効率的に解くモデルである [88]。BERT の最大の特徴は、その双方向性にある。従来のモデルが文脈の片側（通常は左側）からの情報に依存していたのに対し、BERT は文の両側の文脈情報を同時に学習することで、単語間の関係をより深く理解することを可能にした。この特性により、BERT は分類や質問応答といった自然言語理解タスクにおいて、従来の手法を大幅に上回る性能を発揮している。BERT の登場以前、自然言語処理タスクではタスクごとに最適なモデルが異なり、それぞれに特化したモデルが用いられていた。しかし、BERT の登場により、事前学習済みの BERT モデルを各タスクに応じてファインチューニングすることで、多様なタスクにおいて高精度な処理が可能となった。

BERT の事前学習は、以下の 2 つの主要タスクを通じて行われる。1 つ目は「Masked Language Model (MLM)」であり、文中の一部の単語を隠してそれを予測するタスクである。これにより、文の前後関係に基づく深い意味理解が促進される。2 つ目は「Next Sentence Prediction (NSP)」であり、与えられた 2 つの文が論理的に連続しているかを予測するタスクである。このタスクは文章間のつながりを理解するために重要であり、特に文書全体の構造を把握する際に有用である。

BERT は、その構造に基づいて Base と Large という 2 つのモデルバリエーションを提供している。BERT Base は 12 層の Transformer エンコーダを持ち、約 110M のパラメータで構成される。一方、BERT Large は 24 層のエンコーダを持ち、約 340M のパラメータを持つ。これにより、タスクの要件やリソースに応じて適切なモデルを選択することが可能である。

BERT を改良したモデルも数多く登場しており、それぞれが異なる用途や制約に対応する形で設計されている。以下に代表的なモデルを詳述する。

RoBERTa (Robustly Optimized BERT Pretraining Approach) は、BERT を基盤としながらも事前学習プロセスを最適化したモデルである [93]。RoBERTa は、BERT で使用されていた「Next Sentence Prediction (NSP)」タスクを削除し、文脈理解に集中する形で訓練された。また、訓練データ量やバッチサイズ、学習率の最適化により、自然言語推論やテキ

スト分類タスクで BERT を上回る性能を示している。

DistilBERT は、BERT の軽量化モデルであり、パフォーマンスを 80%以上維持しながらも計算資源の消費を大幅に削減することに成功した [94]。DistilBERT は「知識蒸留 (Knowledge Distillation)」という技術を使用し、大規模モデルから重要な特徴を抽出することでモデルサイズを削減している。このモデルは、リソースが限られた環境で特に有用であり、応答時間が重要なアプリケーションに適している。

ALBERT (A Lite BERT) はさらに効率化を進めたモデルであり、パラメータの共有とファクター化埋め込みという 2 つの技術を特徴としている [95]。パラメータ共有により、Transformer の各層で重みを共有することでパラメータ数を大幅に削減し、メモリ効率を向上させた。また、ファクター化埋め込みは単語埋め込みと出力層のサイズを分離することで効率化を図っている。ALBERT は特に大規模データセットを用いた文書分類や推論タスクで高い性能を発揮している。

Sentence-BERT は、文の類似性計算に特化したモデルであり、BERT をベースに文ベクトルの計算を最適化したものである [96]。[96] や [97] でもあげられている課題として、従来の BERT モデルは、文章や文のベクトルの質が GloVe[98] と呼ばれる 2014 年に提案された手法に劣ることがある。Sentence-BERT はこの課題を解決するために、類似した文章間の距離は近くなるようにしつつ、意味の異なる文章間の距離は遠くなるようにモデルを学習させる対照学習を取り入れている。その結果、検索エンジンや質問応答システムなどの意味的な文の類似性が重要なアプリケーションで Sentence-BERT は重宝されている。特に、B.3.2で取り上げる LLMs の生成を手助けする RAG において、大量のテキストデータから必要な情報を検索する必要がある。したがって、軽量で高品質な文ベクトルの計算に優れたモデルの開発は今後も重要である。

A.5.2 GPT 型

GPT (Generative Pre-trained Transformer) 型は、言語モデルの中でも生成タスクに特化したモデルとして位置づけられている。このモデルは、2018 年に初めて OpenAI によって提案された [99]。GPT 型モデルの主な特徴は、次単語の予測を基本タスクとして事前学習を行い、その後、適切なプロンプトを用いることで幅広い自然言語処理タスクに対応可能である点にある。

GPT の第一世代は Transformer のデコーダ部分のみを活用したアーキテクチャを採用しており、その構造のシンプルさと柔軟性が特長である。このモデルは単一方向の文脈情報を利用し、次に来る単語を予測する Next Token Prediction タスクを学習することでテキストを生成する。このアプローチは、双方向的に文脈を理解する BERT とは対照的であるが、生成タスクにおいては大きな利点となる。次単語の予測を繰り返すことで、GPT は自然で一貫性のある文章を生成する能力を持つようになる。

その後、GPT-2[100] が登場し、パラメータ数が 1.5B に増加したことで、より大規模なデータセットから学習が可能となり、多様なタスクでの性能向上が実現した。このモデルは、特定のタスク専用にファインチューニングがされなくても、多様なタスクに対して、Zero-shot または少数の例示を与える Few-shot で優れた性能を発揮する。これにより、事前学習済みのモデルを汎用的に利用するという新しいアプローチが広まった。

GPT-3[101] ではさらにその規模が飛躍的に拡大し、175B という圧倒的なパラメータ数を持つモデルが実現された。このモデルは、膨大な計算リソースとデータセットを活用することで、幅広いタスクにおいて高精度な応答を生成可能となった。特に、GPT-3 は高度なプロンプト設計により、ほとんどの自然言語処理タスクを適切に解く能力を示しており、プロ

ンプトを調整することでタスクに応じた応答を生成する汎用性を持つ。

GPT-4[102] ではさらなる改良が施され、マルチモーダル入力への対応が可能となった。これにより、単なるテキストデータだけでなく、画像データを入力として処理することが可能となり、より幅広い応用範囲を持つようになった。また、GPT-4 はモデルのスケールアップによる性能向上が続く中、効率性にも重点が置かれ、より高度な推論能力を実現している。

GPT 型モデルは OpenAI に限らず、他の組織や研究者によっても開発されている。例えば、Meta による LLaMA(Large Language Model Meta AI[103]) や、Google の PaLM (Pathways Language Model[104]) などが挙げられる。これらのモデルは、GPT 型の設計思想を取り入れつつ、それぞれの目的や環境に合わせた改良が加えられている。さらに、BloombergGPT[105] のような金融特化型モデルも開発されており、特定のドメインにおける高度なタスク遂行が可能となっている。

B LLMs の学習と活用

B.1 LLMs の事前学習

LLMs の学習は、大きく分けると、事前学習と事後学習に分けることができる。事前学習とは、いわば、言語体系や知識を学ぶフェーズであり、事後学習はその出力や表現を学ぶフェーズである。

LLMs の事前学習においては、様々な大規模コーパスを学習させる。日本語の場合、日本語版 Wikipedia[13] や、Common Crawl[14] や The Pile や RefinedWeb の日本語 split、Common Crawl から抽出を行った mC4 や CC100 や OSCAR などが使われることが多い。これらはすべて Web 由来のコーパスである。

さらに、これらのコーパスの前処理を行う必要がある。Web 由来のコーパスであるため、ノイズ除去が必要であり、それぞれのモデルで様々な工夫が行われている。例えば、HojiChar[15] というライブラリは、前処理に必要な様々なフィルターを提供しており、句読点が少なすぎる文章や平均分長が短いドキュメントを削除するなどできる。さらに、重複除去という、似たような文章を削除することも重要であり、MinHash を用いて重複除去を行う手法 [106] が比較的ポピュラーである。

これらの前処理後のコーパスに基づいて、トークナイザーの学習を行う。トークナイザーの構築にあたっては、Sentencepiece [16] を用いるのが一般的である。Sentencepiece では、言語に依存しないトークナイザーを実現し、単語の境界やスペースに依存せずにサブワード単位の分割を行う。Sentencepiece には、BPE と unigram という二つのモードが存在する。

BPE とは、すべての単語を文字 (バイト) レベルに分割し、それを 1 つの単位として語彙に格納する。そのうえで、それらの語彙を組み合わせてできる語彙の中で頻度の高いものから順に語彙に追加するという手法である。

一方で、unigram は Kudo[107] によって提案された手法である。このアルゴリズムでは、初期の最も大きい語彙を構築したのちに、言語モデルを用いて、尤度が低下する具合が最も低い語彙を削除していくことで目的の語彙サイズまで圧縮していく手法である。

[13] https://ja.wikipedia.org
[14] https://commoncrawl.org/
[15] https://github.com/HojiChar/HojiChar
[16] https://github.com/google/sentencepiece

これらの unigram と BPE の実装のどちらも Sentencepiece で利用可能となっているが、これ以外の観点として、辞書等を用いた形態素解析を併用する場合もある。形態素解析機としては、MeCab[108][17] や Sudachi[109]、Ginza[110] などが知られている。これらの形態素解析機を用いて、形態素分解を行った後に、その形態素境界を維持するという前提で BPE や unigram といった手法を適用するという方法もある。

さらに、形態素解析機を併用する場合においても、語彙構築時のみに形態素解析機を適用する場合と、実際の推論時においても形態素解析機を適用するという 2 つのパターンがあり、これらの性能を網羅的に検証した研究 [111] も存在する。

これらをまとめると、トークナイザーの構築にあたっては、

- BPE と unigram のどちらを用いるか？
- 形態素解析機を適用するか？
  - 形態素解析機を適用しない
  - 形態素解析機を構築時のみ使用する
  - 形態素解析機を常に適用する

の 6 パターン存在している。ただし、形態素解析機は環境への導入の不便さ[18] や処理の遅さなどの観点から、少なくとも推論時にも適用することはあまり多くない。また、形態素解析機を導入しても、最終的なパフォーマンスには大きな差がないことからも、近年は利用されない場合も多い。そのため、一般的には、BPE または unigram のみでトークナイザーを構築するのが一般である。

Sentencepiece で語彙を構築後、必要に応じて、特殊トークンを語彙に追加する。例えば、Begining of Sentence (BOS) トークンや、End of Sentence (EOS) トークンがある。また、後述のインストラクション・チューニングモデルに対応するトークンとして、インストラクションメッセージ開始/終了トークン (im_start/im_end) などを追加しておくこともある。

トークナイザーの構築後、学習に使用するテキストをトークナイズすることにより、学習に使用するデータの構築ができる。

そのうえで、A.5章で示した、各モデルに対応するタスクを学習させていくこととなる。BERT 型の場合、MLM および NSP を、GPT 型の場合には Next Token Prediction を学習させることとなる。

学習を行うにあたって考えなければならないこととして主たることは以下である。

- モデルアーキテクチャ
- 学習に使うライブラリ
- 計算機環境
- ハイパーパラメータ

[17] `https://taku910.github.io/mecab/`

[18] Mecab に対応した fugashi の登場などにより、基本的に pip install で導入可能になったため、ハードルとしてはそこまで高いものではなくなっている

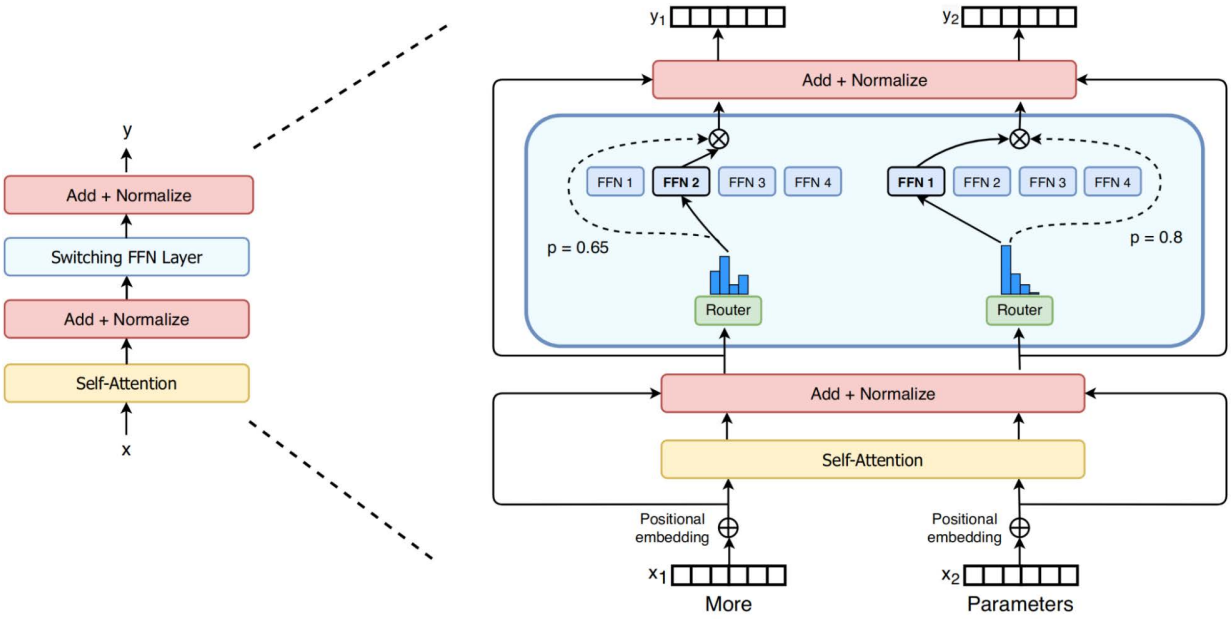

Figure 1: MoE 型ネットワークの構造の模式図 ([112] より)

これらは、複雑に相互作用しているため、総合的に判断を行う必要がある。

まず、モデルアーキテクチャについて、BERT 型の場合、概ね Transformer Encoder を積み重ねただけのモデルが中心であるが、GPT 型の場合、様々なアーキテクチャが存在する。最も大きな違いは Dense 型か、Mixture of experts (MoE) 型かという点である。Dense 型とは、単に Transformer Decoder を積み重ねただけの構造である。一方で、MoE 型というのは、ニューラルネットワークの途中層に Router が存在し、その Router が計算に使用するネットワークを選択し、一部のネットワークしか使わないようにするという構造であり[19]、模式図を図 1に掲載する。

MoE 型にはメリット・デメリットが存在する。メリットとしては、推論時の高速化・低コスト化があげられる。MoE の場合、一部分しか推論時に有効化されないため、計算コストが低くなる。例えば、mistralai/Mixtral-8x22B-v0.1[20] や mistralai/Mixtral-8x7B-Instruct-v0.1[21] の場合、8 つの experts のうち、常に 2 つしか有効化されないため、大規模なパラメータを持つにもかかわらず、1/4 程度のパラメータ数のモデルと同等の計算コストとなる。一方で、学習の難しさがデメリットである。どの Expert に割り振るかということを決める Router と選ばれた Expert のネットワークの両方を同時に学習するため、このバランスをうまくとりながら学習させることがかなり難しいとされている[22]。そのため、現状では、限られたモデルでしか MoE 構造は採用されていない。

また、Dense 型の GPT であっても、その細部において、モデルごとに異なるアーキテクチャを用いている場合がある。例えば、QK-Normalization[113] という、Attention における安定性を高める技術や、z-loss[113] という出力部分に対する損失関数の cross entropy の

[19] よくある間違いとして、複数の LLMs を持ってきて合議制をとると考えられている場合があるが、一般に MoE は一つの LLMs 内で完結する話である。

[20] https://huggingface.co/mistralai/Mixtral-8x22B-v0.1

[21] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

[22] https://tech.preferred.jp/ja/blog/pretraining-1t/

数値を安定させる損失関数、学習の安定性を高める Embeddings Normalization[114] などがある。さらに、Transformer 事態をやめて、RNN のような状態空間モデルをベースとした Mamba[115] なども存在する。これらの膨大な選択肢の中から、モデルアーキテクチャを決める必要がある。

さらに、学習にあたっては、高速化のための様々なライブラリが存在する。LLMs の学習においては、python の深層学習ライブラリである PyTorch[116] が使用されることが多いが、それだけでは高速化が十分ではない。高速化のさらなる方向性としては、大きく分けてモデルパラレルとデータパラレルの 2 つである。モデルパラレルとは、モデルを複数個に分割し、それぞれ異なる GPU に置くというものである。データパラレルは、モデルのレプリカを複数個作成し、それらを異なる GPU においたうえで、データを分割して並列で計算させる方法である。このモデルパラレルとデータパラレルは併用可能である (パイプラインパラレルともいう)。

モデルパラレルやデータパラレルに対応するため、様々なライブラリが存在する。マルチ GPUs などに対応させたり、メモリを最適化するためのライブラリとして、Accelerate[117] が存在する。また、Zero Bubble[118] という、パイプラインパラレルの場合における計算のスケジュールを最適化することで、モデルを分割している場合における GPU の処理の順序性を維持しつつ全体の学習の効率化を行う手法があり、それを実現する DeepSpeed[119] が存在する。また、モデルパラレル・データパラレル・パイプラインパラレルをまとめて 3D parallelism と言い、それを実現する Megatron-LM[23] も存在する。

大規模な LLM の構築においては、上述のパラレル化ということは不可欠であり、これは非常に多くの GPU を使用することを意味する。そのため、計算機の設計も考える必要がある。

パラレル処理を行い場合には、GPU 間での通信が必要になる。例えば、モデルを複数に分割した際に、ひとつ前のブロックから数値を受け取ったり、一つ後ろのブロックから勾配を受け取ったりする必要がある。また、データパラレルやパイプラインパラレルを行う際には、モデルの重みの更新の際に、モデルのレプリカが持つ勾配情報をすべてまとめたうえで、モデルの重みの更新をすべてのレプリカに配信する必要がある。NVIDIA A100 や H100 といった GPU においては、GPU 上のメモリが 80GB 存在しており、そういった規模の通信を GPU 間で行わなければならない。また、GPU が複数のノード (計算機) に分かれていたとした場合には、そのノード間通信も必要となる。たとえば、A100 の GPU を 8 枚積んだ計算機が 2 台あり、その 2 台の間ですべての GPU 上のメモリを同期しようとする場合、合計 640GB の通信が必要になり、100Gbps の光ファイバーで接続したとしても、約 1 分 (640GB ≈ 5Tb) 近くかかることがわかる。そのため、通信の高速化も不可欠である。

通信の高速化を実現する方法はいくつか存在する。まず、伝統的な方法としては、InfiniBand を用いるケースである。そもそも GPU を計算機の基盤に接続している PCIe は、バージョン 4.0 で x16 スロットの場合、32GB/s 出るため、1 台の GPU に 8 枚の GPU が刺さっている場合、2Tbps の速度が出る[24]。そのため、GPU と計算機基盤間の速度はあまり問題ではない。そのため、計算機同士を InfiniBand で接続した場合には、InfiniBand NDR 12 レーンで 1.2Tbps 出るため、概ね学習における通信速度問題は解決できる。もう一つの方法としては、NVIDIA の NV Link[25] を用いるパターンである。こちらも、A100 に対応する NV Link の場合、約 5Tbps の速度がノードを超えた GPU 間でも出るため、通信速度は大幅に改善さ

[23] https://github.com/NVIDIA/Megatron-LM

[24] 実際には基盤側の処理速度などのボトルネックは存在する場合もある

[25] https://www.nvidia.com/ja-jp/data-center/nvlink/

れる。こうしたLLMsの学習に向いた通信速度の改善は、グローバルなクラウド計算機環境では実現が難しい部分もあり、前出のZero Bubbleのような計算スケジューリングの最適化と通信速度の改善の両輪をうまく使って計算効率を高めていく必要がある。なお、100Gbps未満の限られたTCP通信のみを用いた場合のLLMs向けのインフラ構築の取り組みに関しては、[120]も参照されたい。

最後に、LLMsの学習に必要なハイパーパラメーターの調整についてだが、基本的には先行研究などを参照しながらモデルにあったパラメータを適切に選ぶ必要がある。例えば、Adam optimizerのϵ項のパラメータの設定値をLlama 2[121]に従って高めに設定してしまった結果、175Bモデルの学習に失敗したllm-jp[122]のケース[26]がある。このように、モデルごとに適切なハイパーパラメータが異なるため、この決定はとても難しい。

これらの一連のプロセスを通じて、LLMsの学習は可能になる。

B.2 LLMsの事後学習

LLMsの事前学習が完了後、LLMsの応答の有用性、正しさ、無害性を確保する観点から、出力の調整を行うようなチューニングを実施する必要性がある。この学習を事後学習や、Supervised Fine Tuning (SFT)などという。以下では、その主要な手法について説明する。

B.2.1 Instruction-Tuning

Instruction-Tuning（指示チューニング）は、2021年にGoogleのFLAN[123]で提案された。

FLANで提示されているインストラクションデータの例

次の文章を読んで、仮説が前提から推測できるかどうかを判断しなさい。
前提: ロシアのヴァレリー・ポリアコフ宇宙飛行士は、1994年から1995年にかけて、438日間という驚異的な宇宙滞在最長記録を樹立した。
仮説: ロシア人は宇宙滞在最長記録を保持している。
選択肢:
- yes
- no

このようなインストラクションデータをたくさん用意して追加のLLMsの学習を行うことで、好ましい出力をできるようにすることを目指す手法である。

その後、Self-Instruction[124]やAlpaca dataset [125]といったインストラクションチューニングデータが登場し、そのテンプレートも洗練され、現在主流の形式となった。

Self-Instructionで提示されているインストラクションデータの例

###指示: 下記のトピックに関する著名人の言葉を教えてください。
###入力: トピック: 正直であることの重要性
###応答: 正直は知恵という本の第一章である。（トーマス・ジェファーソン）

これらのインストラクションデータセットの多くはOpenAIのモデルを用いた生成を行っており、LLMsの学習に使用することは禁じられており、学術的研究目的のみで使用可能である。また、インストラクションデータの構築は、原則として人手で行うしかないため、比

[26] https://drive.google.com/file/d/1ceCQwN4ZS7_IwmR37l3WAJQH_maZb3qH/viewを参照

較的高価である。例えば、日本語のデータで商用利用可能な ichikara-instruction [126] は、1万件のインストラクションデータが 400 – 800 万円程度である[27]。

高価である点とライセンス上の問題を解決するためによく使われるデータセット構築手法が、ルールベースで既存のコーパスを加工する手法である。例えば、llm-japanese-dataset v0 [127] では、既存の日本語コーパスを元にインストラクションデータを構築している[28]。

データセットの構築が完了後は、そのデータを用いて、LLMs を追加学習させることでインストラクションチューニングは実現できる。しかしながら、応答部以外 (指示文や入力文、###から始まるフォーマット定義されている部分) は、学習したいターゲットではない。例えば、指示文の部分で、「下記のトピックに関する著名人の言葉を教えてください。」という文において、「下記のトピックに関する」までを与えられた場合に、「著名人の言葉を教えてください。」という部分を生成する学習はインストラクションチューニングの趣旨には全く合わない。そこで、応答部以外に関しては、学習の対象にならないように、Cross Entropy Loss のマスクを行う必要がある。

インストラクションチューニングを行うことにより、より好ましい出力が得られることも明らかになっている[29]。

B.2.2 Reinforcement Learning from Human Feedback (RLHF)

RLHF とは、強化学習を用いて、人間の好むような出力を LLM に学習させる手法である。

本手法は、LLM の開発以前に OpenAI 社が要約モデルのチューニングにおいて使用した [128] ものがベースとなっており、その後、InstructGPT 用に使用した [129] のが始まりである。

RLHF は、大きく分けて二つのステップからなる。報酬モデルの構築と強化学習のフェーズである。

第一ステップの報酬モデルの構築においては、ある入力文に対する出力文の好ましさをスコア化するモデルを構築することを目指す。様々な入力文に対して、2 つの応答文のうち、どちらほうが好ましいかというデータセットを大量に作り、それらの組を対照学習させることにより、応答文の好ましさを判定する報酬モデルを作成する。これにより、入力文と出力文を与えると、その好ましさを定量化できるモデルが作成できる。例えば、この報酬モデルの例として、NVIDIA 社の Nemotron-4-340B-Reward[30] などが存在しており、報酬モデルもまた大規模なモデルとなっている場合も多い。

第二ステップでは、報酬モデルを用いて、LLM の応答がより好ましくなるように、LLM 自体をファインチューニングする。このステップでは、Proximal Policy Optimization (PPO) [130] という、方策ベースの強化学習手法を用いる。ここで、強化学習を用いるのは、出力の離散性ゆえである。つまり、LLM の出力は連続的に変化せず、どのトークンを次に出力するのかという離散的な問題であるため、報酬モデルを用いたときに獲得できる報酬を最大化する方向の勾配をうまく LLM に流すことができないためである。そのため、方策勾配という、離散空間を強化学習における行動空間とみなして、学習を行う強化学習由来の手法を使う必

[27] https://web.archive.org/web/20241004061813/https://drive.google.com/file/d/1v8i2jgI2yiUaKBoNla6A_zk0O0_vom4_/edit

[28] https://github.com/masanorihirano/llm-japanese-datasetで利用可能。ただし、後続で登場したベンチマークで使用されているタスクが含まれているため、使用にあたってはベンチマークに使用されていない部分のみを使用するなどの工夫が必要である。

[29] 例: https://engineering.linecorp.com/ja/blog/3.6b-japanese-language-model-with-improved-dialog-performance-by-instruction-tuning

[30] https://huggingface.co/nvidia/Nemotron-4-340B-Reward

要があるため、PPO が使用される。これにより、報酬モデルを通じて獲得できる報酬を最大化するような学習が行われ、LLM がアップデートされることにより、人間の選好に一致した出力を得られるようなファインチューニングが実施可能になる。

しかしながら、この PPO によるチューニングは、報酬モデル、LLM に続き、PPO の内部アルゴリズムで使用される参照モデルというチューニング前の LLM をも同時に使用するため、GPU のメモリ使用が激しいという問題点がある。そのため、次節で述べる DPO が使用されることが増えている。

B.2.3 Direct Preference Optimization (DPO)

DPO [131] は、RLHF の問題点を解決するために、大規模な報酬モデルを使うことなく、RLHF の損失関数を直接強化学習で最適化することにより選好チューニングを行う手法である。

大規模な報酬モデルは、人間の選好をサロゲートするモデルとして構築されていたものの、そこで使用される損失関数を直接強化学習の目的関数にすることで、直接的な学習を行えるというのが DPO のメリットである。これにより、報酬モデルの学習コストが削減され、RLHF で問題であった GPU のメモリ利用の問題も解消される。また、DPO の損失関数と RLHF の損失関数の同値性についてもすでに議論されており [131]、DPO を用いることで RLHF 相当の学習ができることも明らかにされている。

B.3 LLMs の活用

B.3.1 Prompt-turning

LLMs のパラメータ数の増加に伴い、事前学習により様々なタスクにおける性能が向上するとともに、パラメータを調整するためのコストが増加した。そのような背景もあり、学習データによってモデルのパラメータを調整するのではなく、入力である prompt を適切なものに調整する「Prompt-turning」は、非常に重要である。そのため、プロンプトの最適化を自動的に行う研究なども取り組まれている [132]。

B.3.2 RAG

LLMs の活用で課題となるのが、ハルシネーションの存在である。ハルシネーションは、LLMs がユーザーの入力に対して事実とは異なる情報を生成することを指す。例えば、大学レベルの数学の計算問題を入力として LLMs に与えても、正しい回答が返ってくる問題の方がまだ少なく、誤った数値や解法を出力する [133]。このような数学の問題のハルシネーションは、タスクが難しすぎるがゆえに起こるハルシネーションであるが、ハルシネーションの原因の大半は、モデルが回答に必要な情報を学習していないことに起因している。そこで、モデルが回答に必要な情報を学習していない典型例として、社内の Q&A システムでの LLMs の活用を考える。

提出書類の作成方法、出張申請、勤怠システムの使い方など、従来であれば、担当者にメールで問い合わせをするのが一般的であったが、人的コストの削減や質問のしやすさの向上を目的に、多くの会社で Q&A システムの導入が進んでいる。しかし、社内独自の資料などは当然 LLMs の学習に使用されていないため、質問に答えることができず、場合によってはハルシネーションが起こる。LLMs の訓練は、社内独自の情報を外部の環境に出すことがリスクとなることやコストの問題で難しい。仮に、多大なコストを払って LLMs を訓練できたとしても、社内のルールは時間の経過に伴って変化していくものである。変更の度に LLMs

を訓練するのは不可能であり、継続学習のような tuning では過去に学習させてしまった古い情報が残ったままになる。

このような問題を解決するのが、RAG (Retrieval Augmented Generation) である。RAG は、回答に必要な情報をデータベースから検索してくることで、適切な回答を可能にする手法である。したがって、社内ルールが変わった場合には、データベースのデータを入れ替えるだけで解決する。ただし、RAG をシステムに組み込むためには、別途検索システムを構築する必要があり、検索アルゴリズムの精度や速度がシステムのボトルネックになる可能性がある。

金融業界においては、常に最新の情報に注意を向ける必要があるが、直近数ヵ月の情報を LLMs に訓練させるだけでもコストは膨大であるため、LLMs がどんなに発展したとしても RAG の適用はこの先も必要であり続けるだろう。