対話型 AI（ChatGPT）を用いた論文要旨の生成に関する検討

加藤貴之*1

ChatGPT は人間らしく自然な回答を生成する能力を持ち，様々な応用が期待されているが，最適な利用方法はまだ開拓段階にある．そこで本報告では，熊谷組技術研究報告に投稿された論文等のアブストラクト生成に ChatGPT を使用し，その生成能力の限界と可能性を調査する．アブストラクトの生成は次の 3 つの手順とした：1) 論文を分割して要約，2) 要約からアブストラクトを生成，3) 文章の体裁調整．その結果，ChatGPT は一定の文章生成能力と解釈能力を持つが，専門的な内容を誤りなく解釈して表現するのには限界があるとわかった．しかし，文章案として使うなどの使い方次第では十分に効果的な活用が可能であると考えられる．

キーワード：NLP，Generative AI，LLM，Prompt Engineering

1．はじめに

AI に関する研究開発を行っている企業である OpenAI から 2022 年 11 月に公開された ChatGPT は，人間らしい自然な回答を生成するという革新的な能力を持つ大規模言語モデルであり，2 ヶ月余りで全世界でのユーザ数が 1 億人を超えるなど，注目を集めている．さらに，その後 2023 年 3 月に公開された新しいモデルである GPT-4 は，模擬司法試験を受験者の上位 10%前後のスコアで合格するなど，その性能の高さを示している [1)]．この大規模言語モデルの高度な文章生成能力は，複雑な問題や課題に対する解答を得られる可能性を秘めており，各種業界での応用が期待されている．たとえば，小説や漫画などのアイデア出し，英語の学習サポート [2)]，プログラミングのコード生成や自動補完 [3)]，など多種多様な形でその活用が試みられている．一方で，具体的にどの程度の品質が期待でき，どのように活用すれば最大の効果を得られるのかという問いは，今なお課題となっている．

このような背景から，本研究では具体的な事例の一つとして，熊谷組技術研究報告に投稿された論文等のアブストラクト作成を対象に，ChatGPT の能力と精度の検証を行う．論文のアブストラクトは論文の主旨や重要な結果を簡潔にまとめるため，深い理解と緻密な表現力が求められる．この作業を ChatGPT に任せることで，その文章生成能力と解釈能力を試し，ChatGPT の限界と可能性を探ることを目的とする．また，本報告は ChatGPT を用いて文章案を作成し，その内容を参考にする形で作成を行った．

なお，本報告は ChatGPT の出力をそのまま用いて論文やそのアブストラクトを作成することを推奨するわけではない．大規模言語モデルは Hallucination（幻覚）と呼ばれるもっともらしい嘘やでっち上げを生成する可能性があることが知られている [4)]．そのため，出力結果が正しいかどうかを使用者自身が判断する必要があり，そのまま使用することは推奨されないことに注意してほしい．

Fig.1　LLM の仕組み．入力の次に続くテキストを予測し，確率の高いものを出力することでテキストを生成している．質問に答えられるのも，質問のあとに回答が来るというパターンを学んでいるためである．

＊1　技術本部　技術研究所　環境工学研究室

2．LLM の仕組みと特徴

ChatGPT を効果的に使用するためには，その仕組みや特性を理解することは重要である．そこで本章では ChatGPT とその基盤となる大規模言語モデル（Large Language Model：LLM）の仕組みと特徴について解説する．

2．1　LLM の仕組み

LLM は大量のテキストデータを学習させた非常に多くのパラメータを持つ言語モデルのことである．そして言語モデルとは，入力されたテキストデータに続く次の単語を予測するためのモデルである（Fig.1）．LLM はこの言語モデルを大規模化させることで，人間らしい自然な回答を生成する能力を獲得している．なお，何をもって大規模とするかの明確な定義は存在しないが，一般的に数十億以上のパラメータを持つモデルを指すことが多い [5)]．

ここで重要な点として，LLM はテキストの内容を理解して回答しているわけではない，ということである．LLM が人間のように振る舞い，結果的に意味のある文章を生成することができるのは，大量のテキストデータから統計

的なパターンを学ぶことで，確率的に出現する可能性の高い単語を予測できるためである．また，LLM は自然な文章を生成することに長けており，人間が自然に使うようなフレーズや表現を模倣する能力を持つが，複雑な計算や専門知識が必要な問題を解くことは苦手としている．これは LLM がテキストの表面的なパターンを学んでいるだけであり，人間のような深い意味理解や論理的思考を行う能力は持っていないためである．そのため，LLM が知識を持つ人間のような返答をする場合でも，それはあくまでモデルが学習したデータに基づくパターン生成の結果であり，モデルが知識や意識を持って返答しているわけではないということを理解しておくことが重要である．

２．２　プロンプトエンジニアリング

LLM は単純なプロンプト（入力テキスト）でも回答を得ることはできるが，回答の品質を上げるための様々な手法が存在する．たとえば，具体的で明確な指示，出力形式の指定，回答する人物像や役割（いわゆるペルソナやロール）の指定，などを行うことで品質を向上させることができる．このプロンプトを工夫することで回答を最適化する技術を「プロンプトエンジニアリング」といい，様々な研究がされている．例を挙げると，少数の質問とその回答を例示することで回答方法を指示する Few-Shot [6]，論理的思考過程を回答と共に出力するように求める Chain-of-Thought [7] や Zero-Shot CoT [8]，複数の回答を生成し，その多数決をとる Self-Consistency [9]，質問内容に関する知識を事前に生成させる Generated Knowledge Prompting [10]，回答の生成，その内容の改善点を指摘して修正，ということを繰り返す Recursively Criticizes and Improves [11]，などといった技術がある．これらの技術イメージを Fig.2 に示す．この図は説明の都合上，論文の内容と若干異なる部分も含むが，こうした工夫を行うことで回答の品質を上げることができる．また，こういった品質を上げるプロンプトそのものを LLM に生成させようという Automatic Prompt Engineer [12] という手法もある．更には，回答するために必要な行動を思考し，それをもとに Web 検索などの外部ツールを使用することで回答精度を向上させる ReAct [13] という試みも存在する．

これらの研究例が示すようにプロンプトの工夫次第で生成精度が大きく変わる．したがって，より質の高い回答を得るためには，プロンプトの検討が必須となる．

２．３　ChatGPT の制約や仕組み

本報告で使用する ChatGPT は，基本的には前述の LLM と同様の特性を有しているが，それに加えて ChatGPT 特有の特性や制約も存在する．そこで本節では，ブラウザで操作できるユーザ向けのアプリケーションである ChatGPT と OpenAI の LLM である GPT モデルについて説明する．なお，本報で使用した ChatGPT は ChatGPT May 24 Version であり，モデルは GPT-4 を使用した．また，次の GPT モデル

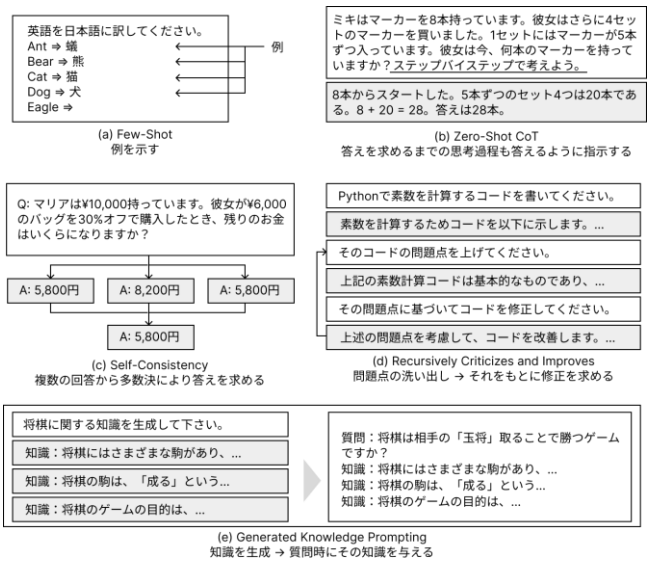

Fig.2　プロンプトエンジニアリングの例．白背景がプロンプトでグレー背景が LLM の回答である．このようなプロンプトの工夫により，回答の品質を上げることができる．

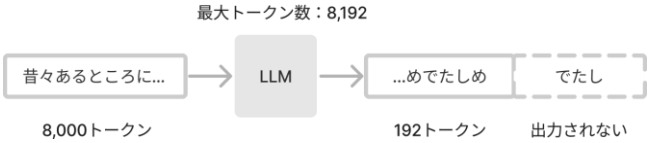

Fig.3　最大トークン数の制約．入出力の合計トークンが上限を超えると，超えた分のトークンは出力されない．

ルの説明は gpt-4-0613 についての内容であり，以降これを GPT と表記する．

２．３．１　GPT

GPT は OpenAI の LLM であり，人間が好むように対話に特化した形で学習されている．そのため前述の LLM と基本的に特性は同じではあるのだが，実際に使用する際の注意事項があるためその説明を行う．

（1）最大トークン数

GPT には最大トークン数という制約がある．自然言語のテキストはそのままでは GPT 上では扱えないため，テキストをトークンという単位に変換する．このトークンという単位を基準として GPT が一度に扱える上限が存在する．そして GPT の最大トークン数は，8,192 である．これは，入力トークン数と出力トークン数の合計が 8,192 を超えることはできないということを意味する．たとえば，入力トークンが 8,000 だとすれば出力は最大 192 トークンしかなく，その後に続くトークンは出力されない（Fig.3）．そのため，一定以上の長さのテキストを生成するためには，使用する側で入力トークンを抑制する必要がある．

（2）記憶

GPT は入力したテキストを記憶することはできない．たとえば GPT に 2 度質問する場合，2 回目の回答は 1 回目の質問および回答結果を考慮することはない．過去のやり取りを考慮して出力を行う必要があれば，別途処理が必

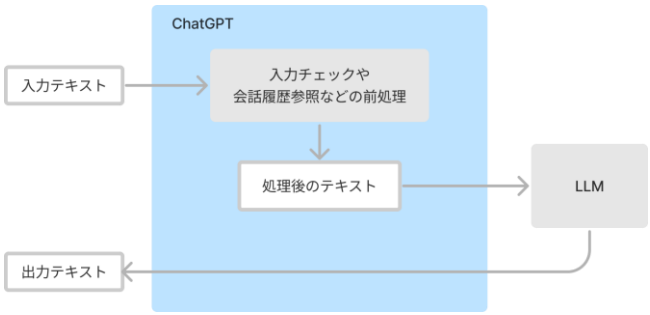

Fig.4　ChatGPT の動作．LLM をそのまま使用しているわけではなく，前処理を行っている．

要になる．

（3）再現性

確率的にテキストを生成するという特性上，同じ入力を与えても出力は異なる可能性があるため，出力に再現性はない．

2．3．2　ChatGPT

ChatGPT のアプリケーションは，この GPT を使用して構築されている．そのため，GPT のトークン数上限や再現性の制約はアプリケーションにも引き継がれる．しかし，GPT をそのまま使用しているわけではなく，アプリケーションとして動作するために様々な処理が行われている．たとえば，GPT の最大トークン数近くまでユーザが入力すると，ほとんど何も返答できなくなるため，アプリケーション上ではユーザの入力上限が GPT の最大トークン数よりも少なく制限されている．実際に ChatGPT を使用してみたところ，入力するテキスト内容にもよるが日本語ではおおよそ 2,600～2,800 文字程度が入力上限であった．また，前述の通り GPT に記憶能力はない．あくまで GPT に入力された内容に従って出力を返すだけである．そのため ChatGPT はそれまでの会話履歴を記憶して回答をしているように振る舞うために，前処理を行っている．会話履歴が短ければ，それまでの会話履歴すべてを GPT の入力とする，といった方法が考えられるが，やり取りが長くなったり，ユーザの入力や ChatGPT の出力が長くなると，会話履歴すべてを GPT に入力することはできない．GPT の最大トークン数を超えてしまうと，テキストが生成できないためである．つまり，ChatGPT はそれまでの会話履歴をすべて GPT に入力しているわけではなく，アプリケーション上で会話履歴の加工を行ってから GPT に入力している，と考えられる（Fig.4）．また，後処理については結果がストリーミングで返ってくる（徐々に表示される）特性などから，実施されていないと考えられる．ただし，これらの挙動は ChatGPT を使用してみての推測であり，アプリケーション内部で具体的にどのような仕組みが働いているのかは明らかではないことに注意してほしい．

このように ChatGPT はアプリケーションとして成立させるために様々な処理を行っていると思われ，それはブラックボックスである．わかることは，入力上限が設定されていること，何らかの前処理を行ってからテキストを生成していること，でありこれを念頭に置いてアブストラクトの生成の検討を行う．

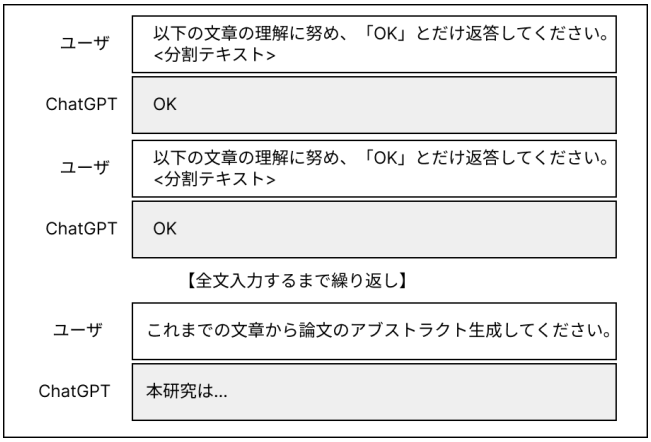

Fig.5　分割したテキストをすべて入力し終えてからアブストラクトを生成するイメージ．今回この方法は採用しなかった．

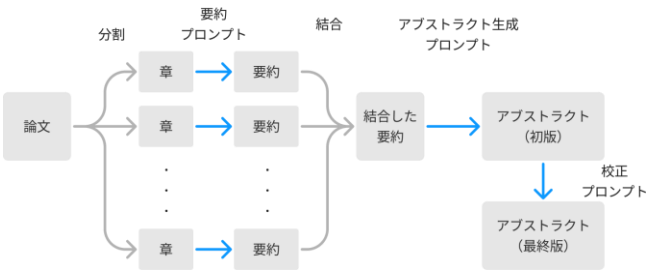

Fig.6　アブストラクト生成手順．論文を分割して要約，アブストラクト生成，校正，という手順を踏んだ．

3．アブストラクトの生成プロセス

本章では前章で説明した LLM の特性および ChatGPT のアプリケーション上の制約を考慮した上で，ChatGPT を活用して論文のアブストラクトを生成する手法について検討を行う．

3．1　論文からテキストの抽出

ChatGPT への入力はテキスト形式であり，本研究で対象とした熊谷組技術研究報告は PDF 形式である．そのため，まず PDF から手作業でテキストをコピーして本文の抽出を行った．テキストの抽出を手作業で行った理由は，自動処理において図や表のテキスト，ヘッダーやページ番号が本文と混ざって抽出されてしまう問題があるためである．さらに，自動抽出は正確さに欠けるため，チェックと修正が不可欠となる．特に，テキスト抽出の精度はアブストラクトの生成結果に影響を及ぼす可能性が高いため，これらの要因を考慮して手作業を採用した．また，論文の構成要素としてタイトル，要旨，キーワード，参考文献など，本文以外の情報も存在するが，これらはアブストラクトの生成には使用しない．なお，本文中に含まれる数式については文字化けなどを起こしてしまうため，PDF から正

確に抽出することが難しい．そのため，別行立て数式については抽出しなかった．しかし，本文中に組み込まれたインライン数式については，抽出の手間の都合上，文字化けなどを起こしていてもそのまま扱った．

３．２　アブストラクトの生成手法の検討

論文の本文からアブストラクトを生成する手法の検討過程について説明する．特に，ChatGPT の入力文字数の上限問題とそれに対する解決策について詳述する．

論文は一般的に長文であり，全文を ChatGPT に入力すると文字数上限を超えてしまう．この問題は，長文要約タスク全体に共通する課題である．そこで，論文を ChatGPT に入力可能な文字数上限以下になるように分割し，分割したテキストを順に入力して全文入力し終えるまで処理を待ってもらうという方法が考えられる（Fig.5）．しかし，この方法は必ずしも理想的な解決策とは言えない．ChatGPT の応答生成メカニズムは，過去の会話履歴全体を使用して回答を生成するわけではない．それまでの会話内容を加味した回答が生成されることから，会話履歴を要約する，会話履歴から質問に関連する箇所を抽出する，などといった前処理を行っていると推測はできる．しかし具体的な処理はブラックボックスであり，それが長文要約用途で使用することを想定して処理が行われているかどうかも不明である．

上述の問題を考慮し，分割したテキストを順に入力し，入力し終えてからアブストラクトを生成するという方法は適切ではないと判断した．そこで今回選択した方法は，論文を適切な単位で分割し，分割したテキストに対して要約を生成するというものである．そして得られた要約を結合し，この結合した要約から最終的なアブストラクトの生成を行う(Fig.6)．この一連の手順を踏むことで，ChatGPT の入力上限を超える論文に対してアブストラクトの生成を行う．また，要約した内容を結合してさらに要約するという手法は，書籍などの長文を要約する手順としても有効であると報告されている [14] ため，この方法を採用した．次節以降で，この手法の詳細について述べる．

３．３　分割要約

まず文章の分割については基本的に章単位に行った．各章はそれぞれ独立した主題を扱うため，章単位の分割は自然な選択である．しかし，一つの章が入力上限を超える長さの場合は，さらに細かく分割した．この細分化は適切な切れ目（たとえば，節または段落の区切り）で行った．

そうして分割した文章から Fig.7 に示したプロンプトを使用して要約を生成した．このプロンプトは，優秀な研究者であるという役割を与え，指示や制約条件，出力内容などを明示し，200 文字以内での簡潔な要約，重要なキーワードの確保，そして文章の意味を忠実に保つことを要求することで，生成される要約の精度を向上させることを目指した．

```
# 命令書:
あなたは、多くの論文を発表している優秀な研究者です。
以下の制約条件と入力文をもとに、最高の要約を出力してください。

# 制約条件:
- 要約は200文字以内
- 重要なキーワードを取りこぼさない
- 文章は簡潔に
- 文章の意味を変更しない

# 入力文：
<ここに文章>

# 出力文：
```

Fig.7　要約生成プロンプト．次のアブストラクト生成時の入力文が長くならないように，簡潔にまとめるような指示にした．このプロンプトにより分割した論文を要約する．

```
あなたは優秀な研究者です。以下の手順に従って論文のアブストラクトを作成してください。
1. 優れた論文のアブストラクトを書くにあたってのベストプラクティスをあげてください
2. そのベストプラクティスを念頭に置いて、制約条件と入力文から論文のアブストラクトを作成してください

# 制約条件
- 200-400文字にすること
- 文章の意味を変更しないこと

# 入力文
<ここに文章>

# 出力内容
- ベストプラクティス
- アブストラクト
```

Fig.8　アブストラクト生成プロンプト．ベストプラクティスを先に生成することで，それを考慮したアブストラクトを生成する．

```
論文のアブストラクトを要件に従って修正してください。
# 要件
- 「である調」にする
- 句読点は「，」「．」にする
- 日本語として自然な文章にすること
- 文章の意味を変更しないこと

# 修正前のアブストラクト
＜ここに文章＞

# 修正後のアブストラクト
```

Fig.9　校正プロンプト．論文のアブストラクトとしての形式に文章の体裁を整える．

また，必要な情報が抜け落ちないように，「箇条書きで情報を網羅的に出力すること」を要求するプロンプトも試みた．しかし，この方法では出力された文章が長くなり，最終的なアブストラクトの生成結果が悪化する傾向が観察された．これは要約内容が長くなることでアブストラクトに重要な項目がどの部分にあるのかを判断することが難しくなるためと考えられる．このプロンプトが長くなると精度が低下する傾向があるという結果は，Liu et al.の論文にも報告されている [15]．このため，要約を短く簡潔にして重要な内容を適切にまとめることを強調する

プロンプトにした．

3．4　アブストラクト生成

分割された要約の結果を結合した後のアブストラクト生成のプロンプトについて説明する．Generated Knowledge Prompting のアプローチを参考に，ベストプラクティスの生成を要求し，その後にその情報を考慮してアブストラクトを生成するという二段階のプロセスを採用した（Fig.8）．ベストプラクティスとは最適な方法やセオリーのことであり，今回の場合，目的や結果を明確に説明すること，簡潔に記述すること，などに当たる．これによりベストプラクティスが反映されたアブストラクトの生成が期待できる．

ベストプラクティスとアブストラクトの両方を一度に生成するのではなく，ベストプラクティスは予め ChatGPT で生成しておき，アブストラクト生成時の入力プロンプトに含める形にすることも検討した．しかし，ChatGPT より出力されるベストプラクティスは実行ごとに変わる．恣意的に内容を選択しないためにも，それは採用せずアブストラクトとともに生成する形にした．

3．5　体裁の調整

最後に，生成されたアブストラクトに対する体裁の調整を行った．ChatGPT により生成されたアブストラクトは，「ですます調」であったり，句読点が「、」「。」であったりと，論文作成における指定と異なる場合があった．また，日本語として若干不自然な表現も見られた．これらの点を解消するため，最後に文章の体裁を整えるプロンプトを実行し，その結果を最終的なアブストラクトの出力とした（Fig.9）．このプロンプトにより，生成されたアブストラクトは「である調」に整形され，句読点は「，」「．」に変更される．さらに，文の構造や語彙選択が日本語として自然な形に調整されることが期待される．

4．アブストラクトの評価とその結果

本章では，前章で説明した手法を用いて，熊谷組技術研究報告に投稿された論文等のアブストラクトの生成を試み，その結果のアンケートを筆頭執筆者にとることにより評価を行った．対象とした論文および研究報告は過去 3 年間に発行された 79 号から 81 号までのデータから執筆者が技術本部所属（2023 年 4 月時点）であり，かつ執筆者の重複がない形で選定した計 16 件の論文とした．

また，アブストラクトの生成は各論文一度のみ行い，再生成は行わなかった．同じ入力でも出力に再現性はないため，何度も生成してより良い結果を選択するということも考えられるが，恣意的な結果となってしまう．本報告の目的である ChatGPT の能力の検証という観点からも，一度の生成での結果を評価する方が適切であると考えた．

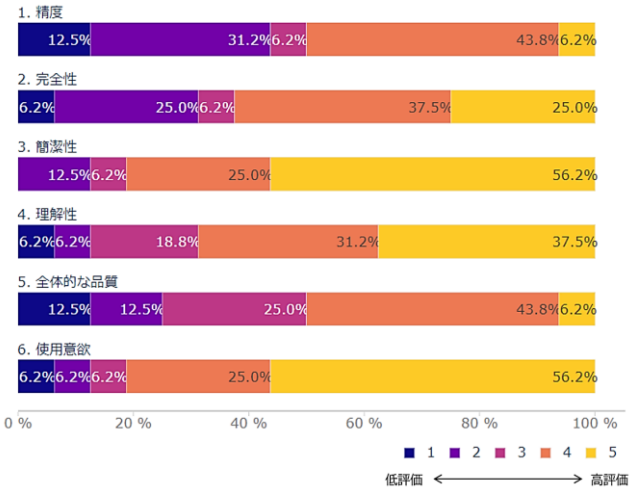

Fig.10　アンケート結果．

4．1　アンケート内容

生成されたアブストラクトを評価するために，それぞれの論文の筆頭執筆者に対してアンケートを実施した．アンケートは「精度：内容は正しいか」「完全性：必要な項目が網羅されているか」「簡潔性：文章が簡潔か」「理解性：理解しやすい文章か」「全体的な品質」「使用意欲：アブストラクトの作成に ChatGPT を今後使用したいと思うか」の 6 つの項目について 5 段階のリッカート尺度（1：低評価～5：高評価）で選択する設問と自由記述という形式で行った．なお，アンケート項目は ChatGPT により何度か生成させ，修正を加えることで決定した．

4．2　アンケート結果

アンケート結果を Fig.10 に示す．精度については 1，2 が 44%，4，5 が 50%，完全性については 1，2 が 31%，4，5 が 63%であり，どちらも高評価と低評価で分かれる結果となった．この 2 つは，内容の正確さを確認する設問であり，結果が分かれたのは正しいものと誤っているものが混在していたためと思われる．簡潔性については，4，5 の評価が 81%と高めの評価であり，理解性については，4，5 の評価が 69%と比較的高めの結果となった．この 2 つは，文章としての良し悪しの評価項目であり，どちらも比較的高めの評価となったのは ChatGPT の文章生成能力の高さが反映された形になったと思われる．また，全体的な品質については 1，2 が 25%，4，5 が 50%であった．使用意欲については，4，5 の評価が 81%と高評価であり，新しい技術への期待感が反映された形になったと思われる．

自由記述では「誤解を与える表現となっていた」「ニュアンスに違和感を感じた」「細かい部分で説明が足りていない」などの意見が見られた．これは論文という専門的な内容を扱う関係上，文章の細かなニュアンスを表現しきれなかったのだと思われる．また，プロンプト上で文字数制限をかけて簡潔に出力するように指示をしていることもあり，主語や接続詞などが省略されてしまったケース

があったことも起因していると思われる．そして，細かな内容や数値的な誤りの指摘もあり，もっともらしく見える誤った内容を生成する「Hallucination」の問題も発生している．逆に「草稿や見比べなどには使える」「執筆者が修正すれば，十分使用可能」「自身で修正するなどの使用方法に留まる」といった意見もあり，不足部分や表現の誤りなどを執筆者が修正すれば，活用できるというものになっていると思われる．しかし「根本的なところで間違っている」「日本語的におかしい」といった意見もあり，すべての論文についてうまく生成できたわけではないこともわかった．これについては，対象論文の専門的な内容を ChatGPT が学習していないことや，ChatGPT の性質上，出力にブレがあるため初回の生成で悪い出力を引き当ててしまっていた，などということが考えられる．また，今回使用した論文は研究だけではなく製品開発の報告も含まれている．今回プロンプトを「研究者として論文のアブストラクトを作成する」という研究の論文が入力される前提として作成してしまったため，開発経緯を説明している報告文には適していなかった，ということも考えられる．

5．まとめ

本報告では，ChatGPT を用いて熊谷組技術研究報告に投稿された論文等の本文からアブストラクトを生成する手法を検討し，生成した結果の評価を行った．生成手法は大規模言語モデルの特性と ChatGPT の制約を考慮して，最終的に次の方法を取った．1）論文のテキストを分割して要約し，2）要約を結合した内容からアブストラクトを生成し，3）文章の体裁を整える，という 3 段階の手順を踏むことで生成を行った．

この手法で生成したアブストラクトについて，筆頭執筆者にアンケートをとることで結果を評価した．また，アンケートでは 79 号から 81 号までの熊谷組技術研究報告から 16 件を対象とした．アンケートの結果「全体的な品質：1～5（低品質～高品質）」の設問では 1，2 が 25%，4，5 が 50%と良い評価が若干多いという結果であった．また，自由記述を見ると表現やニュアンスについての指摘が多く見られたが，全体としてはまとまっており草稿や参考には利用できるという意見もいくつかあった．

この結果から，ChatGPT は一定程度の文章生成能力と解釈能力はあるものの，論文のような専門的な内容を誤りなく解釈して表現するのには限界がある，ということがわかった．しかし，そのまま使用するのではなく文章案として使うなど，使用するユーザの工夫次第では十分に効果的な活用ができると思われる．

謝辞

本報告での ChatGPT 生成のアブストラクト評価に当たって，アンケートにご協力いただいた技術本部の方々に感謝いたします．

参考文献

1) OpenAI: GPT-4 Technical Report, 2023.
2) Kohnke, Lucas, et al.: ChatGPT for language teaching and learning, RELC Journal, 00336882231162868, 2023.
3) Chen, Mark, et al.: Evaluating large language models trained on code, arXiv preprint arXiv:2107.03374, 2021.
4) Ji, Ziwei, et al.: Survey of hallucination in natural language generation, ACM Computing Surveys 55.12, pp.1-38, 2023.
5) Zhao, Wayne Xin, et al.: A Survey of Large Language Models, arXiv preprint arXiv:2303.18223, 2023.
6) Brown, Tom, et al.: Language Models are Few-Shot Learners: Advances in neural information processing systems 33, pp.1877-1901, 2020.
7) Wei, Jason, et al.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Advances in Neural Information Processing Systems 35, pp.24824-24837, 2022.
8) Kojima, Takeshi, et al.: Large Language Models are Zero-Shot Reasoners, Advances in neural information processing systems 35, pp.22199-22213, 2022.
9) Wang, Xuezhi, et al.: Self-Consistency Improves Chain of Thought Reasoning in Language Models, arXiv preprint arXiv:2203.11171, 2022.
10) Liu, Jiacheng, et al.: Generated Knowledge Prompting for Commonsense Reasoning, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.3154-3169, 2022.
11) Kim, Geunwoo, Pierre Baldi, and Stephen McAleer.: Language Models can Solve Computer Tasks, arXiv preprint arXiv:2303.17491, 2023.
12) Zhou, Yongchao, et al.: Large Language Models Are Human-Level Prompt Engineers, arXiv preprint arXiv:2211.01910, 2022.
13) Yao, Shunyu, et al.: “ReAct: Synergizing Reasoning and Acting in Language Models”, arXiv preprint arXiv:2210.03629, 2022.
14) Wu, Jeff, et al.: Recursively Summarizing Books with Human Feedback, arXiv preprint arXiv:2109.10862, 2021.
15) Liu, Nelson F., et al.: “Lost in the middle: How language models use long contexts”, arXiv preprint arXiv:2307.03172, 2023.

Study on generation of abstracts of scientific papers using conversational AI (ChatGPT)

Takayuki KATOH

Abstract

ChatGPT is able to generate human-like natural replies and expected to be applied for various uses. However, the optimal uses have yet to be explored. In this study, we used ChatGPT to write abstracts of past papers published in Kumagaigumi Technical Research Reports and examined the limits and potential of its generating capabilities. Abstracts were generated in the following three steps: (1) Divide a paper into several parts and summarize each part; (2) generate an abstract from the summaries; and (3) adjust the text style. As a result, we found that ChatGPT is capable of generating and interpreting text to some degree, but that there are limits to its capabilities to interpret and express technical content unerringly. However, it can be sufficiently effective for some applications: for instance, for creating rough drafts.

Key words:　NLP, generative AI, LLM, prompt engineering