CE-013

ChatGPT を用いた記事の構成要素を加味する長文文章生成手法の提案
Generating Long Texts Considering Article Structure Using ChatGPT

澤崎 夏希[1)]　遠藤 聡志[2)]
Natsuki Sawasaki　Satoshi Endo

1　はじめに

ChatGPT[1] などの Large Language Model（LLM）の登場よって，自然言語処理分野は大きく発展した．LLM により、従来は人手で行われていた多くのタスクが自動化されるようになってきた [2]. 特に、自然言語における文章のかさ増しは、従来の手法では困難であったため、LLM を用いたかさ増し手法が期待されている [3][4][5][6]．かさ増し手法は人手によるデータの作成コストが大きい場合に必要とされる．例えば，長文のようなデータセットは人手でのかさ増しコストが大きく，人手を介さないかさ増し手法が求められている．しかし、LLM の性能は言語により差があることが知られており，また短文の生成結果を評価されることが多い．特に，英語の短文に対しては高い精度が報告されている一方で，日本語の長文生成に関する評価は限定的である．

かさ増しにおいて，元データの文長を保持できない場合，特徴となっている語彙や表現が失われてしまうリスクが高くなるため，文長を維持できるようなかさ増し手法が必要となる．そこで本研究では、日本語長文生成のために、記事データから抽象化された構成要素を取り入れることで、日本語の長文文章のかさ増しを行う．記事の構成要素を加味することで，全体を把握した上で生成を行うことが可能になり，安定した文長の生成が行えることが期待される．記事毎に必要な構成要素は異なると考えられるが，全てのデータに対して人手で解析を行ったプロンプトを作成することは困難である．そこで，記事の構成要素を抽出する LLM を経由することで，自動的に最適な要素の解析と抽出を行う．記事の構成要素を抽出する LLM と，その構成要素を元にかさ増しデータを生成する LLM により，記事の構成要素を加味した日本語長文生成が行え，文長や元の文章の特徴を加味しつつ多様な生成が行えることが期待される．

生成した文章について、文長や共通語彙，固有語彙、文章の類似度を比較し、LLM による自動評価によって元データの特徴が保持されているか検証する．最後に分類実験によりかさ増し手法としての特徴をみる．比較として，記事の構成要素を加味した生成データと加味していない生成データを用意し，それらの傾向の差から記事の構成要素が文章生成に与える影響を評価する．

2　先行研究: Chataug

Dai らは、臨床データに対して ChatGPT[1] を用いたデータ拡張を行い、従来手法と比較して分類精度の向上を示した [7]．図 1 に示す通り、ChatGPT を用いたデータ拡張により臨床データの分類精度が向上し、ChatGPT の有効性が確認された．しかし、使用されたデータセットはいずれも短文であり、英語で構成されていた．これらのデータセットは、文章全体の文脈を考慮する必要がある記事データを対象としていなかった．本研究では、日本語の長文データセットに対して ChatGPT を用いたデータ拡張を行い、その有効性を検証する．

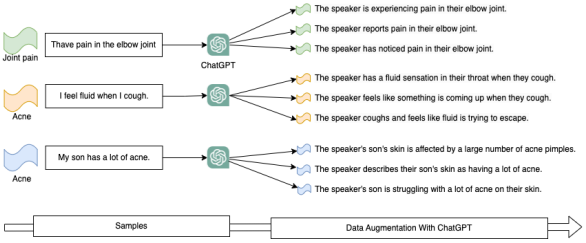


図 1: Chataug のデータかさ増しフロー

3　提案手法

記事の全体像を構造として抽出し，要素毎に生成を行うことで長文の生成が行えると考えた．また，記事毎の構造は一定ではなく，インタビューやコラム，紹介記事など多彩なフォーマットが想定され，それぞれについて人手でプロンプトを作成するのは困難である．そこで，元データの記事の構成要素を抽出するための LLM を用意し，抽出された構造を元に文章の生成を行う．生成工程を図 2 に示す．かさ増し元のデータはカテゴリ毎にランダムに記事を 100 件抽出し，記事 1 件毎にデータの生成を行う．データの生成は，LLM を用いて記事から構成情報を抽出し，抽出された構成情報をプロンプトとして LLM に入力し記事を生成する．

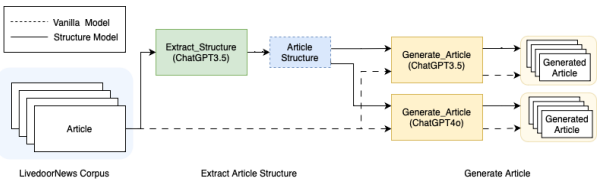


図 2: 提案手法

3.1　記事の構成要素抽出

記事の構成要素は以下のプロンプトを用いて抽出を行う．最適な要素の構成は記事ごとに異なると考えられ，人手でその全ての構成要素を定義するのは困難である．そこで，記事の構成要素を抽出するためのプロンプト設計した．LLM によって自動的に記事の構成要素抽出することにより，多様な記事に対しても安定した水準の生成が行えると考えた．

記事抽出に使用する LLM として *gpt-4o-2024-05-13*(ChatGPT4omni) と *gpt-3.5-turbo-0125*(ChatGPT3.5) が考えられるが，数件の出力結果を比較し精度的な問題

1)　†琉球大学 理工学研究科 総合知能工学専攻
Department of Integrated Intelligence Engineering, Graduate School of Science and Engineering, University of the Ryukyus.
2)　†琉球大学 工学科 知能情報コース
Intelligent Information Course, Engineering department, University of the Ryukyus

が見られないのを確認し，コストの観点から *gpt-3.5-turbo-0125* を使用した．使用したプロンプトを図 3 に示す．

以下の記事を読んで、記事の内容に応じて適切な構成要素を抽出してください．記事のフォーマット（インタビュー、紹介、コラムなど）を特定し、それに基づいて次のような代表的な構成要素を参考にしつつ、記事ごとに動的に適切な要素を抽出してください．
代表的なフォーマットと構成要素の例
1. **インタビュー** - 質問: インタビュアーが行った質問- 回答: インタビュー対象者の回答- 背景情報: インタビューの背景や対象者の紹介- 注目ポイント: インタビューで特に重要なポイントや興味深い点
2. **紹介記事** - 主題: 紹介している対象（人物、場所、イベントなど）- 特徴: 主題の特徴や重要なポイント- 背景情報: 紹介対象に関連する背景情報や歴史- 詳細: 紹介対象の具体的な詳細情報
3. **コラム** - 主題: コラムの中心となるテーマや議論のポイント- 背景: 主題に関連する背景情報や状況- 論点: コラム内で議論されている具体的な論点- 結論: コラムのまとめや筆者の結論

元データ

記事の内容に応じて、適切な構成要素を抽出し，抽出された構成要素を基に、元の記事の流れや意味を維持しつつ、自然な追加情報やエピソードを挿入するよう指示するプロンプトを作成してください．元の文章を使用する箇所は省略し，元の記事という表記に置き換えてください

図 3: 記事の構成要素抽出プロンプト

3.2 データセット

使用するデータセットは、livedoor news corpus を使用する [8]．このデータセットは、NHN Japan 株式会社が運営する「livedoor ニュース」の一部を基に、可能な限り HTML タグを取り除いて作成されたカテゴリ分類用のデータセットである．カテゴリ毎のデータ量と，平均文長，平均語彙数を表 1 に示す．商品名などそれぞれの記事に固有の語彙が存在し、カテゴリ名がファッション等の一般的なジャンルではなく、メディア独自のカテゴリ名を使用していることが特徴である．

表 1: 各カテゴリのデータ数と平均文長・平均語彙数

category	article count	length(ave)	vocabulary(ave)
dokujo-tsushin	870	1596.79	329.20
it-life-hack	870	1273.05	235.40
kaden-channel	864	970.62	221.05
livedoor-homme	511	1658.43	315.65
movie-enter	870	1519.68	324.62
peachy	842	1319.80	273.84
smax	870	1770.24	278.07
sports-watch	900	710.01	184.15
topic-news	770	749.81	183.29

4 実験

記事の構成要素を用いたかさ増しの影響を評価するため，記事構成を加味した生成データと，加味しない生成データを用意する．また LLM モデルを変えると傾向が変わると考えられるため，生成に使用するモデルは ChatGPT4omni と ChatGPT3.5 を使用し．生成データは以下の４通りとなる．

- Vanilla35: 生成指示のみの ChatGPT3.5
- Vanilla4o: 生成指示のみの ChatGPT4omni
- Structure35: 記事構成+生成指示の ChatGPT3.5
- Structure4o: 記事構成+生成指示の ChatGPT4omni

まず本研究の主題である長文生成について，文長・語彙の比較により評価する．その後，生成されたデータの特徴を，生成元のデータとの文章類似度比較と LLM による自動評価により評価し，最後に分類実験への影響を調査する．

4.1 生成文章の文長比較

生成元の記事と生成モデルによる生成文の文長を比較し，それぞれの生成文が生成元の文長を確保できているか評価する．カテゴリ毎に集計したものを表 2 に示す．

表 2: データの平均文長

category	Original	Vanilla35	Vanilla4o	Structure35	Structure4o
dokujo-tsushin	**1596.79**	857.81	881.34	753.32	1399.23
it-life-hack	1273.05	737.84	795.13	700.86	**1378.85**
kaden-channel	970.62	680.97	672.78	676.88	**1315.72**
livedoor-homme	**1658.43**	877.89	1032.45	784.57	1436.77
movie-enter	**1519.68**	791.66	757.13	771.32	1376.88
peachy	1319.80	871.36	924.44	777.99	**1420.20**
smax	**1770.24**	935.56	920.06	753.67	1514.89
sports-watch	710.01	605.09	634.57	659.33	**1177.64**
topic-news	749.81	580.71	604.34	672.91	**1202.57**
average	1284.53	770.72	803	727.63	1357.94

表からわかるように，記事の構成要素を加味した Structure4o による生成データが最も文長が長く，元データよりも平均文長が長くなっていた．特に、kaden-channel と、sports-watch，topic-news は増加量が大きく，平均約 400 文字も文長が増加している．これらのカテゴリは元の平均文長が短い傾向があることから，元の文章がある程度少量でも，記事の構成要素を加味することで，安定した量の文章が生成できることを示している．一方で元の文長が長い dokujo-tsushin や、smax については生成元の方が文長が長い．このことから，記事の構成要素を加味することで 1300 文字前後の範囲で安定した生成が行えるようになったと考えられ，長文生成において記事の構成要素を考慮することは有効であると言える．

4.2 生成文章の語彙比較

次に，得られた長文文章データがどのような語彙を有しているかを評価した．これは文長が確保されていても，繰り返しやモード崩壊 (mode colapse) のような多様性の消失が考えられるためである．表 3 にカテゴリ毎に語彙数の集計を行った結果を示す．

表 3: データの平均語彙数

category	Original	Vanilla35	Vanilla4o	Structure35	Structure4o
dokujo-tsushin	**329.20**	195.01	198.98	167.58	272.40
it-life-hack	235.40	153.74	161.05	147.12	**250.00**
kaden-channel	221.05	155.07	153.91	152.47	**250.53**
livedoor-homme	**315.65**	187.55	198.81	168.79	272.31
movie-enter	**324.62**	182.15	179.59	176.51	275.57
peachy	**273.84**	186.89	196.29	168.28	263.56
smax	**278.07**	177.08	177.69	155.49	265.63
sports-watch	184.15	149.54	152.84	152.93	**242.14**
topic-news	183.29	145.88	147.48	156.10	**242.72**
average	**260.44**	170.21	174.09	160.54	259.41

語彙数の比較でも，生成データの中では文長と同様に記事の構成要素を加味した提案手法が最も語彙を有していた．この語彙数は記事の構成要素を加味していない Vanilla4o と比較しても多く，特に表 2 で文長の増加が顕著だった kaden-channel と、sports-watch，topic-news では，同様に語彙数の増加が見られるため，繰り返しのような生成ではなく，多様な表現を得た結果文長が増加したことがわかる．このことから多様性を拡張する目的でも記事の構成要素のような全体像を考慮に入れることは有効だと考えられる．

4.2.1 生成文章の共通語彙比較

これらの語彙が，元データの語彙と大きく異なるドメインのデータとして生成されている恐れがあるため，元データと共通の語彙数を調べ同じカテゴリ性を持っているかを調査する．元データと共通する語彙数を図 4 に示す．

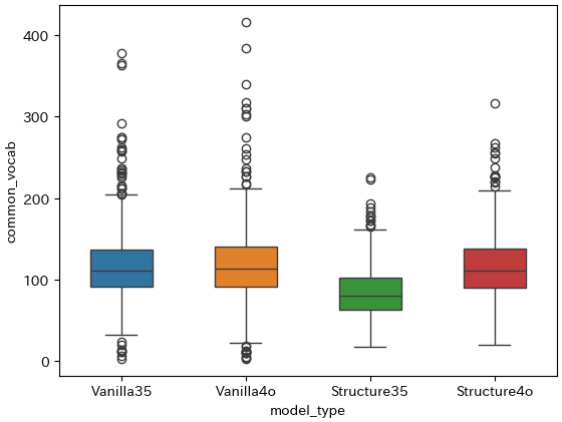


図 4: 共通語彙比較

この図から，多くの語彙が得られた記事の構成要素を加味した Structure4o モデルは，他手法と同程度に元データとの共通語彙を持ちつつ，他手法よりも固有の語彙を多く獲得していることがわかる．最も共通語彙を持っているのは Vanilla4o モデルだが，入力記事の要約等によって共通語彙が増えた可能性がある．特に Structure35 は元データとの共通語彙が少なく，情報が欠落している可能性がみられ，記事の構成要素を有効に利用するにはある程度の性能が必要になると考えられる．これらのことから，ある程度の性能を持つ LLM に対しては，記事の構成要素を加味しても元データの共通語彙は失われず，Vanilla モデルと同程度に元データの特徴を保持できるといえる．

4.2.2 生成文章の固有語彙比較

元データと共通の語彙を持つ必要がある一方で，かさ増し手法としては元データの特徴的な表現は使用しつつ，語彙を広げるようなデータであることが望ましいと考えられる．この語彙を広げた度合いを元データが持たない，生成データ固有の語彙数から評価する．図 3 にモデルごとの固有語彙数の箱ひげ図を示す．

固有語彙では Structure4o が大きく他のモデルを上回った．上図で見た通り，他モデルと比較し最も語彙数が多いモデルなので，元データの語彙を保持しつつ新たな表現も生成した結果語彙が増えたことがわかる．つまり元の語彙に追加する形で語彙が増えているので，提案手法を用いることでデータとしての多様性は拡張されるといえる．

また，最大最小の数字を見ると，Vanilla モデルは最大最小の差が大きいことがわかる．これは記事の複製や要約を行った場合は語彙が全て共通するため 0 になり，元の記事と異なる記事を生成した場合は大きく固有語彙が増えると考えられる．このことから，記事の構造を加味することで，ある程度元のデータの表現を用いつつ，独自性のある表現が可能になったといえる．

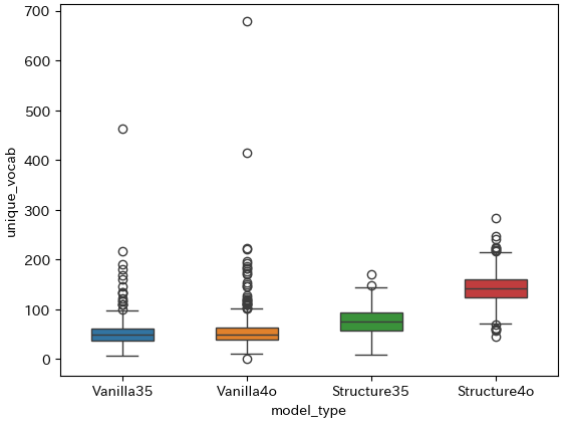
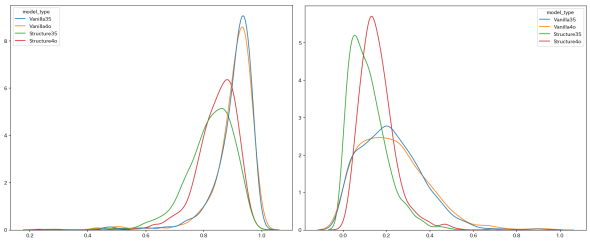


図 5: 固有語彙比較

4.3 文章類似度比較

次に，学習への使用を想定し確保された多様性を特徴量の観点から評価する．かさ増しは同じ意味で違う表現であることが望ましいため，同じ意味であることを cos 類似度で，異なる表現であることを BLUE スコアで評価する．それぞれの類似度はかさ増し元データと生成データとの比較で行われ，同じ意味であるとよいため cos 類似度は 1.0 にある程度近く，異なる表現を得たいため BLUE スコアは 0 に近いほど良い．ただし，cos 類似度は 1.0 に一致すると同一データとなるため，ある程度 1.0 に近い類似度であることが望ましい．BLUE スコアと cos 類似度についてカーネル密度推定 (KDE) で可視化を行ったものを図 6 に示す．文章ベクトルには日本語用 Sentence-BERT モデル[1)]を用いた．



Cosine similarity　　BLUE score

図 6: Similarity KDE

どの生成手法でもある程度，元データと類似したデータが生成されているが，記事の構成要素を加味したモデルの cos 類似度は 0.8 程度の位置に分布していることがわかる．BLUE スコアは 0 に近い位置に分布しており，これは記事の構成要素によって多様性が得られ，元データの特徴を持ちつつ異なるデータとして生成されたといえる．Vanilla モデルは cos 類似度が 1.0 に近い位置に分布が集まっていることから，記事の内容を複製や要約したような生成が行われた可能性が考えられる．

4.4 LLM による自動評価

生成した文章に対して，LLM による自動評価が広く扱われている．人の評価と相関があり，数値で評価が難しい評価を行うことが可能である [9]．本研究でも，

1） Sentence-BERT モデル: https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2

表 4: Classification Results

label	Original	Duplicate	Vanilla35	Vanilla4o	Structure35	Structure4o
dokujo-tsushin	92.60%	92.99%	93.25%	93.12%	92.73%	**93.64%**
it-life-hack	**87.66%**	86.88%	85.84%	86.36%	86.49%	86.49%
kaden-channel	88.61%	**89.27%**	**89.27%**	88.48%	86.39%	87.43%
livedoor-homme	**84.43%**	83.46%	83.46%	83.21%	82.97%	80.29%
movie-enter	95.97%	**96.36%**	95.84%	95.84%	96.49%	94.81%
peachy	**87.06%**	**87.06%**	86.12%	85.44%	86.39%	85.18%
smax	94.55%	94.81%	94.81%	94.81%	95.19%	**95.32%**
sports-watch	97.00%	97.50%	97.50%	**98.00%**	97.88%	97.38%
topic-news	**93.58%**	93.43%	92.99%	93.28%	93.13%	**93.58%**
total	91.67%	**91.76%**	91.45%	91.40%	91.31%	91.03%

文章の自然さや独自性といった数値評価が難しい 5 つの項目に対して，ChatGPT3.5 を利用して以下の評価を行った．

1. 一貫性: 文章全体の流れが一貫しており、追加された部分が元の内容と調和しているか.
2. 自然さ: 追加された部分が自然で違和感を与えないか.
3. 元の記事類似性: 元記事と同じ内容になっているか.
4. オリジナリティ: 元の記事と異なる表現になっているか.
5. 情報欠落: 生成によって失われてしまった情報がないか

それぞれ該当する，該当しないの 2 値で回答を行うように指示し，生成データ 900 件に対しての「該当する」と回答した結果を以下の表 5 に示す.

Evaluation	Vanilla35	Vanilla4o	Structure35	Structure4o
Consistency	856(95.1%)	857(95.2%)	880(97.8%)	**884(98.2%)**
Naturalness	856(96.1%)	857(95.2%)	879(97.7%)	**884(98.2%)**
Theme Maintenance	847(95.1%)	850(94.4%)	824(91.6%)	**853(94.8%)**
Originality	20(2.2%)	15(1.7%)	103(14.4%)	**349(38.3%)**
Lack of Information	**5(0.56%)**	10(1.1%)	10(1.11%)	38(4.2%)

表 5: 自動 LLM 評価

どのモデルも一貫性のある自然な生成が行えており，日本語としての破綻がないことがわかる．全体的に，元のデータのテーマ性も保持されているが，独自性が低い傾向が見られる．これは類似度評価で見られた同一の記事を生成してしまった場合と考えられ，記事の構成要素を加味した方が独自性のある記事を生成できていることが LLM を用いた評価からもわかる．一方で，情報欠落のスコアも Structure4o モデルが最も高くなっている．これは独自性の高い生成を行なった結果元のデータから離れたデータが生成された可能性がある．

4.5 カテゴリ情報の保持性

これまでの実験から，記事の構成要素を用いることで多様な語彙を持つ長文データが生成可能なことが示された．得られた語彙がかさ増しとしてどのように寄与するか分類実験により評価する．かさ増し元とかさ増しデータを学習データとして，残りを全て評価用として使用する．ベースラインとして，かさ増し元データを複製しただけの Duplicate データセットを追加し，分類した結果を表 4 に示し，精度が高い数値を太字で表した．

全体の精度を見ると，Duplicate のデータセットが最も精度が高い．これは複製されたデータで精度が向上するデータであったことを示し，多様な表現を得ることでむしろ精度が下がる可能性が示唆されている．生成データ内の全体精度としては Vanilla 系のモデルが高いが，最も高精度なカテゴリは Structure4o の方が多い．これは Vanilla 系のモデルでは複製に近い生成が行われたが，文長の減少から特徴が失われたと考えられる．一方で Structure4o は多様性の中に元カテゴリの特徴を保持されていたため，精度が向上したカテゴリが多く見られたと言える．

5 まとめ

本研究では記事の構造を LLM によって抽出し，その情報をプロンプトとして入力することで，長文文章生成を行なった．結果として元データと比較しても十分な文長が確保でき，語彙も増加したことから記事の構成要素は長文生成に有効であることが示された．LLM を用いた評価でも提案手法が最も独自性のある生成が行えており，文長だけでなく表現を拡張できたことがわかった．一方で，カテゴリ毎の増加すべき語彙を制御対象にしていなかったため，拡張された語彙が有効なカテゴリと，あまり精度に寄与しないカテゴリが存在することがわかった．よって，よりカテゴリの特徴を含むかさ増しを行う必要はあるが，文長を確保する方向の制御には一定の貢献を示せた．また，分類精度に寄与させる場合，カテゴリ全体の特徴も加味する必要があると考えられる．今後はかさ増しデータの品質についての評価指標を定め，かさ増しの目的に合わせてた制御が可能な手法を模索していく．

参考文献

[1] ChatGPT.https://openai.com/blog/chatgpt/

[2] Törnberg, Petter. ”Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning.” arXiv preprint arXiv:2304.06588 (2023).

[3] He, Haibo, and Edwardo A. Garcia. ”Learning from imbalanced data.” IEEE Transactions on knowledge and data engineering 21.9 (2009): 1263-1284.

[4] Bayer, Markus, Marc-André Kaufhold, and Christian Reuter. ”A survey on data augmentation for text classification.” ACM Computing Surveys 55.7 (2022): 1-39.

[5] SAWASAKI, Natsuki, et al. ”Sentence Generation Method by Extension of MolGAN Using Sentence Graph.” Journal of Japan Society for Fuzzy Theory and Intelligent Informatics 32.2 (2020): 668-677.

[6] Ding, Bosheng, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. ”Data augmentation using llms: Data perspectives, learning paradigms and challenges.” arXiv preprint arXiv:2403.02990 (2024).

[7] Dai, Haixing, et al. ”Chataug: Leveraging chatgpt for text data augmentation.” arXiv preprint arXiv:2302.13007 (2023).

[8] livedoor ニュースコーパス http://www.rondhuit.com/download.html#ldcc

[9] Liu, Yuxuan, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. ”Calibrating llm-based evaluator.” arXiv preprint arXiv:2309.13308 (2023).