大規模言語モデルによる 少数かつ短文の文書に対するトピックモデリング

土井 智暉 [1]　磯沼 大 [1,2]　谷中 瞳 [1]
[1] 東京大学　[2] エディンバラ大学
{doi-tomoki701, hyanaka}@is.s.u-tokyo.ac.jp　m.isonuma@ed.ac.uk

概要

短文の文書に対するトピックモデリングは，近年研究されているニューラルトピックモデルにとっても, 依然として挑戦的なタスクである．一方で，大規模言語モデルは様々なタスクで優れた性能を示しており，トピックモデリングにおいても優れた性能が期待できる．本研究では, 大規模言語モデルとして GPT-3.5, GPT-4 を取り上げ, 少数かつ短文の文書に対するトピックモデリングの性能を調査する. 実験の結果から, 大規模言語モデルは少数かつ短文の文書においては既存のトピックモデルよりも高性能であり, hallucination などの懸念についても影響は実用上無視できるほど小さいことを示す.

1　はじめに

トピックモデリングは文書の集合から潜在的なトピックを発見するタスクである[1)] [1, 2]. 近年ではニューラルモデルが活用され, 従来の統計的確率モデルより高い性能を示している [3, 4, 5]．しかしニューラルモデルは少数データに対しては性能が劣化する傾向があり，さらに短文に対するトピックモデリングは挑戦的であるといわれている [6, 7].

近年では， InstructGPT[8] や GPT-4[9] のような大規模言語モデル (large language models; LLM) が, 適切なプロンプトを与えることで, 様々なタスクで優れた性能を示している [10, 11, 12, 13]. そこで， LLM は少数かつ短文の文書集合に対するトピックモデリングにおいても, 優れた性能を示すことが期待できる．

本研究では，LLM として GPT-3.5, GPT-4 を想定し，少数かつ短文の文書集合をプロンプトで与えたときのトピックモデリングの性能を調査する. LLM に由来する懸念点として, 出力されたトピックが文書のごく一部のみを反映してしまう可能性や, 与えた文書集合に含まれないトピックを出力してしまう hallucination が想定される. そこで, トピックモデリングの評価で標準的な指標に加え, トピックが文書をどれだけカバーしているかを評価する指標と，トピックがどれだけ文書中の語で構成されているかを評価する指標を新たに導入する. また，出力の安定性として，適切なトピックモデリングの出力がどの程度得られるかについても，併せて評価する.

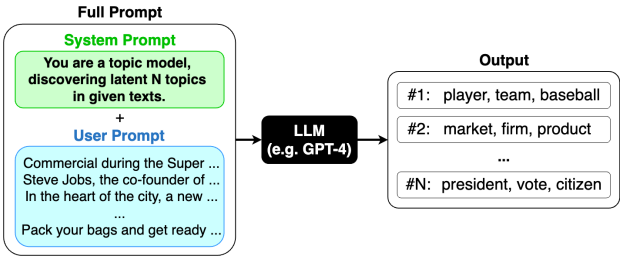

図 1　システムプロンプトおよびユーザプロンプトを通した, 大規模言語モデルによるトピックモデリング．全ての文書は 1 つのユーザプロンプト内で与えられる.

2　背景

トピックモデリングは，文書集合から，潜在的なトピックをトピックワードと呼ばれる語の集合の形で出力するタスクである [1, 2]．発見するトピック数やトピックワードを構成する語数は, 事前に決定する. 従来は Latent Dirichlet Allocation (LDA, [1]) に代表される統計的確率モデルを用いて取り組まれてきたが, 近年ではニューラルモデルを用いた手法が研究され, 高い性能を示している [4, 14, 15].

一方で, 短文からなる文書集合に対するトピックモデリングは, データのスパース性のために難しいことが知られている [6, 7]. TSCTM [7] はそのようなトピックモデリングにおける State-of-the-Art (SoTA) であり, VQ-VAE [16] に基づいた対照学習とデータ拡張を活用している.

1)　本研究では，各文書のトピック分布の推定は考慮しない.

LLM を活用したトピックモデルとしては BERTopic [15] が挙げられる. BERTopic では Sentence-BERT [17] を用いて文書の埋め込み表現を獲得し, UMAP [18] によってクラスタリング, TF-IDF ベースの手法でトピックワードを割り当てる. これに対し, 本研究では GPT-3.5, GPT-4 が文書からトピックワードを構成するまでを End-to-End で行うトピックモデリングを提案する. また, ChatGPT[2] を用いてトピックモデルの性能評価を行った研究 [19] も存在する. しかし, これまでの研究ではトピックモデリング自体における LLM の性能は調査されていない.

3 手法

本研究では, OpenAI が提供する API[3] を用いて, GPT-3.5 (gpt-3.5-turbo-1106) および GPT-4 (gpt-4-1106-preview) に対してトピックモデリングのタスク説明と文書集合をプロンプトとして与えることで, 文書集合からトピックワードを得る.

具体的には, 既存のトピックモデリング手法に倣い, 事前に文書を前処理済みの Bag-of-Words の形に変換する. そして, 各文書を改行記号で結合し, 1 つのユーザープロンプトとしてモデルに与える. 同時に, トピックモデリングのタスク説明を含めたシステムプロンプトを与え, 適切なトピックワード数, トピック数を満たしたトピックワードを出力するように指示する. ただし, モデルの出力が不適切であった場合は再度同様のプロンプトを与え, 適切な出力が得られるまで試行を繰り返す.

プロンプトについては予備実験を行い, [12, 13, 20] で使われているようなプロンプトにおける性能を調査した. 結果, 図 2 に示したシステムプロンプトを与え, ユーザープロンプトで文書集合を与える方法が最も高い性能を示したため, これを用いる.

4 実験

3 節で述べた手法について, 特に少数かつ短文の文書集合に対する性能を調査する. 本研究ではトピック数が 10 および 25, トピックワード数が 5 の条件下でトピックモデリングを行う.

4.1 データセット

5〜10 語程度しか含まない短文文書で構成されるデータセットとして GoogleNewsT [21], Tweet [22], StackOverFlow[4] が挙げられる. 本研究では, これらのサブセット[5] のうち, ランダムにサンプリングした 1000 事例をトピックモデリングの対象とする.

[7] に従い, 前処理として (i) 小文字化, (ii) 2 文字以下の単語の除去, (iii) 出現頻度が 5 回未満の低頻度語の除去 を行う.[6]

> You are a topic model, discovering latent topics in given texts depending on word co-occurrence.
> Inputs are texts of many documents. Each line corresponds to each document, separated by linefeed codes.
> Please discover *NUM_TOPICS* latent topics from input texts and show their meanings with precisely *NUM_TOPWORDS* words extracted from input texts.
> Outputs always should be in the format "Topic N: word1 word2 ..." where N is one to *NUM_TOPICS*. Only make formatted outputs.
> Make sure that the number of words for each topic is *NUM_TOPWORDS*, except for "Topic N:".

図 2 トピックモデリングのためのシステムプロンプト. *NUM_TOPICS* および *NUM_TOPWORDS* は実行時には "five" や "fifteen" など適宜置き換えられる,

4.2 ベースライン

従来の統計的確率モデルである LDA[7] と SoTA であるニューラルモデル TSCTM[7] をベースラインとする. さらに, データ拡張を適用した場合の各モデルについても, 併せてベースラインとする. これは, 短文を扱うタスクおけるデータ拡張の有用性が示されているためである [7, 23].

TSCTM のハイパーパラメータについては [7] に従う. データ拡張については WordNet Augmenter[8] と Contextual Augmenter[8] [24] を適用する. これらは, 文書中の語を, それぞれ WordNet で定義された同義語および BERT[25] モデルが同じ位置に存在しうると予測した語で置き換える拡張手法である. 本研究では [7] に従い, 各手法によって 30%の語を置き換えたのち, 前処理と同様に低頻度語を除去することで拡張データを作成する.

2) https://chat.openai.com/
3) https://platform.openai.com/docs/guides/text-generation (2024 年 1 月 12 日 参照)
4) https://www.kaggle.com/competitions/predict-closed-questions-on-stack-overflow/data?select=train.zip
5) https://github.com/rashadulrakib/short-text-clustering-enhancement/tree/master/data
6) 前処理後のデータセットの統計量については付録 A(表 6) に示す.
7) https://github.com/BobXWu/TopMost
8) https://github.com/makcedward/nlpaug

表 1 Coherence (*Cv*) と Diversity (*TU*) の平均値．“+Aug” はデータ拡張を適用した場合のモデルの性能を表す．Cv, TU はいずれも高いほど良く，TU はトピックワードがトピック間で完全に異なるとき最大値 1 をとる．

モデル	GoogleNews T				Tweet				StackOverFlow			
	$K=10$		$K=25$		$K=10$		$K=25$		$K=10$		$K=25$	
	Cv	*TU*	*Cv*	*TU*	*Cv*	*TU*	*Cv*	*TU*	*Cv*	*TU*	*Cv*	*TU*
LDA	0.374	0.827	0.382	0.712	0.417	0.693	0.419	0.579	0.310	0.633	0.339	0.483
LDA+Aug	0.304	0.940	0.360	0.869	0.428	0.820	0.419	0.736	0.385	0.847	0.411	0.712
TSCTM	0.337	**1.000**	0.455	**0.992**	0.399	**1.000**	0.471	**0.984**	0.364	**1.000**	0.438	**0.941**
TSCTM+Aug	0.357	0.973	0.408	0.947	0.420	0.987	0.438	0.947	0.353	0.953	0.369	0.901
GPT-3.5	0.471	0.947	0.505	0.849	**0.549**	0.879	0.565	0.792	0.439	0.860	0.441	0.685
GPT-4	**0.591**	0.973	**0.552**	0.856	0.524	0.987	**0.569**	0.952	**0.472**	0.960	**0.494**	0.808

表 2 Document Coverage (*DC*) と Factuality (*Fa*) の平均値．“+Aug” はデータ拡張を適用した場合のモデルの性能を表す．いずれの値も高いほど良い．データ拡張を適用しなかった場合のベースラインモデルにおいては，与えられた文書集合のみに基づいてトピックを出力するために，*Fa* の値は必ず最大値 1 をとる．

モデル	GoogleNews T				Tweet				StackOverFlow			
	$K=10$		$K=25$		$K=10$		$K=25$		$K=10$		$K=25$	
	DC	*Fa*	*DC*	*Fa*	*DC*	*Fa*	*DC*	*Fa*	*DC*	*Fa*	*DC*	*Fa*
LDA	0.474	**1.000**	0.700	**1.000**	0.593	**1.000**	0.768	**1.000**	0.769	**1.000**	0.879	**1.000**
LDA+Aug	**0.526**	0.993	0.755	0.963	**0.661**	**1.000**	0.850	0.978	**0.813**	0.976	0.907	0.895
TSCTM	0.471	**1.000**	**0.763**	**1.000**	0.604	**1.000**	**0.852**	**1.000**	0.740	**1.000**	**0.936**	**1.000**
TSCTM+Aug	0.383	0.966	0.682	0.930	0.520	0.926	0.698	0.873	0.459	0.756	0.675	0.66
GPT-3.5	0.406	0.976	0.659	0.970	0.559	0.986	0.748	0.977	0.662	**1.000**	0.827	0.988
GPT-4	0.394	**1.000**	0.610	0.959	0.537	0.966	0.793	0.961	0.666	**1.000**	0.902	0.974

4.3 評価

評価指標として，トピックの品質を評価する標準的な 2 つの指標に加え，LLM によるトピックモデリングで想定される懸念点を考慮するために，2 つの指標を新たに導入する．さらに，LLM によるトピックモデリングの出力安定性についても検討する．

Coherence, Diversity トピックの Coherence (各トピックを構成するトピックワードに一貫性があるか) と Diversity (トピックワードがトピック間で互いに異なっているか) はトピックモデリングの評価指標として標準的である [7, 14, 15]．[7] に従い，Coherence として，Wikipdeia 上でのトピックワードの共起性に基づく coherence value (*Cv*, [26]) を計算し[9]，Diversity として，各トピックにユニークなトピックワードの割合に基づく topic uniqueness (*TU*, [27]) を計算する．

Document Coverage LLM によるトピックモデリングにおいて想定される懸念点として，文書のごく一部のみを反映したトピックが出力される可能性が挙げられる．そこで，モデルが出力したトピックが文書集合をどれだけカバーしているかを評価するために，新しい指標 Document Coverage (*DC*) を提案する．

$$DC = \frac{\left|\{d \in D_{original} : \exists w \text{ s.t. } w \in \bigcup_k T_k, w \in W_d\}\right|}{|D_{original}|}$$

ここで d は Bag-of-words $\{w_1, ..., w_n\}$ で表された文書であり，W_d は文書 d に含まれる語の集合である．また，$D_{original}$ と T_k は，それぞれデータ拡張適用前の元の文書集合と k 番目のトピックにおけるトピックワードを表す．

Factuality LLM に由来するもう 1 つの懸念点は hallucination である．つまり，LLM は文書集合中に含まれないトピックを出力する可能性がある．そこで，モデルが出力したトピックワードが与えられた文書に基づいているかを評価するために，新しい指標 Factuality (*Fa*) を導入する．

$$Fa = \frac{\left|\{w \in \bigcup_k T_k : \exists d \in D_{original} \text{ s.t. } w \in W_d\}\right|}{|\bigcup_k T_k|}$$

9) https://github.com/dice-group/Palmetto

Fa の値が大きいほど，より多くのトピックワードが文書中の語彙で構成されていることになる．なお，データ拡張を適用したベースラインモデルにおいて，Fa の値が 1 未満になりうることを述べておく．なぜなら，データ拡張時の語置換によって，元の文書集合に含まれない語が拡張データに含まれている可能性があるからである．

出力安定性　3 節で述べたように，LLM によるトピックモデリングにおいては，指定したトピック数，トピックワード数に準拠した適切な出力が得られるまで，試行を繰り返す．本研究では，このとき必要だった試行回数を出力安定性の指標として報告する．なお，ベースラインを含む従来のトピックモデルにおいては，適切な出力が得られるのは自明である．

5　結果

各モデルについて 3 回ずつトピックモデリングを実行した．表 1，表 2 に結果を示す．出力安定性についても，3 回の平均値を表 3 に示す．

トピックの品質　表 1 から，GPT-3.5 および GPT-4 のスコアは Coherence (Cv) と Diversity (TU) のいずれにおいても比較的高い値を示し，ベースラインと比較して高品質なトピックを出力していることがわかる[10)]．特に Coherence については，ベースライン間のベストスコアよりも 20〜50% 程度高いスコアを獲得している．例えば，GoogleNewsT において GPT-4 は，トピック数が 10 のとき LDA の 0.374 に対して 0.591，トピック数が 25 のとき TSCTM の 0.455 に対して 0.552 であり，それぞれ約 58%，約 21%の性能改善を示している．

Document Coverage　表 2 から，GPT-3.5, GPT-4 の Document Coverage (DC) はベースラインモデルよりも低いスコアを示していることがわかる．このことから，LLM が出力したトピックがカバーする文書数は，ベースラインモデルのそれよりも少ないと考えられる．ただし，Coherence (Cv) と Document Coverage の間にはトレードオフがあることを述べておく．

Factuality　表 2 から，GPT-3.5, GPT-4 の Factuality (Fa) は，特にデータ拡張しなかった場合のベースラインモデルと比較して，低い値を示していることがわかる．したがって，これらのモデルは文書集合中に存在しない語をトピックワードとして出力しうると考えられる．一方で，多くの場合でこれらのモデルにおける Factuality は 0.98 以上であり，このとき 50 個のトピックワードにつき文書集合中に存在しない語は 1 個程度である．さらに，そのような語を分析した結果，多くは文書集合中に含まれる語の類義語や派生語 (*stream* と *streaming* など)，あるいは関連する語 (*actor* と *movie* など) であり，トピックの誤った解釈を誘因するような有害なものは少ないことがわかった[11)]．これらのことから，LLM の hallucination に関する懸念点は，実用的には無視できると考えられる．

表 3　LLM によるトピックモデリングにおいて適切な出力が得られるまでにかかった試行回数 (出力安定性) の平均値．例えば GPT-3.5 は，GoogleNewsT においてトピック数を 10 とした場合 ($K = 10$) は，平均して 4 回目の試行で初めて適切な出力が得られることを意味する．

モデル	GoogleNewsT		Tweet		StackOverFlow	
	$K = 10$	$K = 25$	$K = 10$	$K = 25$	$K = 10$	$K = 25$
GPT-3.5	4.00	9.00	4.67	15.00	1.67	2.67
GPT-4	2.33	4.67	5.00	8.00	1.33	2.33

出力安定性　4.3 項で述べた出力安定性について，3 回の平均値を表 3 に示す．より大きいトピック数の条件下 ($K = 25$) で，より多くの試行を必要としており，出力安定性が低下することがわかる．しかし一方で従来のトピックモデルにおいても，実際にテキスト分析に使用する際には，トピック数として 10〜15 程度を採用した上で分析しやすい結果が得られるまで試行を繰り返している [28, 29]．したがって，GPT-3.5 および GPT-4 の出力安定性については，実用的には十分であると考えられる．

6　おわりに

本研究では，少数かつ短文の文書集合に対する GPT-3.5 および GPT-4 によるトピックモデリングの性能を調査した．実験においては，トピックの品質を評価する標準的な指標に加え，2 つの新しい指標 Document Coverage と Factuality を導入し，想定される懸念点の影響の大きさを検討した．結果，本研究の実験設定においては，GPT-3.5, GPT-4 は既存のトピックモデルよりも高品質なトピックを出力できることがわかった．また想定される懸念点の影響は無視できるほど小さく，出力の安定性についても実用上十分に高いことを示した．今後の展望として，より文書数が多い文書集合に対する，LLM によるトピックモデリング手法の開発を行う予定である．

10)　トピックの具体的な出力例については付録 A.1 で示す．

11)　具体的な語や分析の詳細については付録 A.2 に示す．

謝辞

本研究は JST さきがけ JPMJPR21C8 の支援を受けたものである.

参考文献

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. **Journal of machine Learning research**, Vol. 3, pp. 993–1022, 2003.

[2] Rob Churchill and Lisa Singh. The Evolution of Topic Modeling. Vol. 54, p. 35, 2022. Issue. 10s.

[3] Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering Discrete Latent Topics with Neural Variational Inference. In **Proceedings of the 34th International Conference on Machine Learning - Volume 70**, 2017.

[4] Akash Srivastava and Charles Sutton. Autoencoding Variational Inference For Topic Models. In **International Conference on Learning Representations**, 2017.

[5] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. Topic Modeling in Embedding Spaces. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 439–453, 2020.

[6] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In **Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval**, 2016.

[7] Xiaobao Wu, Anh Tuan Luu, and Xinshuai Dong. Mitigating Data Sparsity for Short Text Topic Modeling by Topic-Semantic Contrastive Learning. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, 2022.

[8] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In **Advances in Neural Information Processing Systems**, 2022.

[9] OpenAI. GPT-4 Technical Report. **arXiv preprint arXiv:2303.08774**, 2023. version 3.

[10] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking Large Language Models for News Summarization. **arXiv preprint arXiv:2301.13848**, 2023. version 1.

[11] Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. Document-Level Machine Translation with Large Language Models. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, 2023.

[12] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szyd lo, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bart lomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Mi lkowski, Marcin Oleksy, Maciej Piasecki, Lukasz Radliński, Konrad Wojtasik, Stanis law Woźniak, and Przemys law Kazienko. ChatGPT: Jack of all trades, master of none. **Information Fusion**, Vol. 99, p. 101861, 2023.

[13] Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. A Systematic Study and Comprehensive Evaluation of ChatGPT on Benchmark Datasets. In **Findings of the Association for Computational Linguistics: ACL 2023**, 2023.

[14] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. Topic Modeling in Embedding Spaces. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 439–453, 2020.

[15] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. **arXiv preprint arXiv:2203.05794**, 2022. version1.

[16] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural Discrete Representation Learning. In **Advances in Neural Information Processing Systems**, 2017.

[17] 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**, pp. 3980–3990, 2019.

[18] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. **arXiv preprint arXiv:1802.03426**, 2020. version 3.

[19] Dominik Stammbach, Vilém Zouhar, Alexander Hoyle, Mrinmaya Sachan, and Elliott Ash. Revisiting Automated Topic Model Evaluation with Large Language Models. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, 2023.

[20] Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Towards Making the Most of ChatGPT for Machine Translation. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, 2023.

[21] Md Rashadul Hasan Rakib, Norbert Zeh, Magdalena Jankowska, and Evangelos Milios. Enhancement of Short Text Clustering by Iterative Classification. In **Natural Language Processing and Information Systems**, 2020.

[22] Jianhua Yin and Jianyong Wang. A model-based approach for text clustering with outlier detection. In **2016 IEEE 32nd International Conference on Data Engineering (ICDE)**, 2016.

[23] Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. Supporting Clustering with Contrastive Learning. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, 2021.

[24] Sosuke Kobayashi. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, 2018.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, 2019.

[26] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In **Proceedings of the Eighth ACM International Conference on Web Search and Data Mining**, 2015.

[27] Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. Topic Modeling with Wasserstein Autoencoders. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, 2019.

[28] 武富有香, Yuri NAKAYAMA, 須田永遠, 宇野毅明, 橋本隆子, 豊田正史, 吉永直樹, 喜連川優, ROCHA E C Luis, 小林亮太. 日本語の大規模 Twitter データからみる新型コロナワクチン接種に関する人々の関心の推移. 人工知能学会全国大会論文集, 2023.

[29] Viriya Taecharungroj. "What Can ChatGPT Do?" Analyzing Early Reactions to the Innovative AI Chatbot on Twitter. **Big Data and Cognitive Computing**, Vol. 7, No. 1, 2023.

A 付録

表 4 GoogleNewsT に対するトピックモデリングにおいて, トピック数を 10 とした時に各モデルが出力したトピックの例.

モデル	トピックの例
LDA	nokia lumia taylor prince swift
	black video friday comet time
	thanksgiving lakers monday kim friday
TSCTM	kim black west kardashian meningitis
	china moto smartphone spike syria
	market protester pill jos men
GPT-3.5	xbox nokia lumia microsoft hour
	black friday deal shopping cyber
	hiv aid testing study girl
GPT-4	black friday shopping deal cyber
	xbox console microsoft user launch
	nokia lumia smartphone video launch

表 5 GoogleNewsT に対するトピックモデリングにおいて, 各モデルが出力した文書集合中に含まれない語の例.

モデル	出力された文書集合中に含まれない語の例
LDA+Aug	project, champion, hold, number, fleet, tattle, unk
TACTM+Aug	support, website, also, champion, status, red, fleet
GPT-3.5	**playstation**, character, frosty, family, south, prison, brown
GPT-4	**coaliation**, **murdoch**, **philippines**, football, streaming, guilty, actor

表 6 データ拡張 (Aug) の適用有無それぞれにおけるデータセットの統計量. $|D|$ は文書数を示し, *Len* は 1 文書あたりの平均語数を示す. また, $|V|$ は語彙サイズを示す. データ拡張前の文書数が 1000 未満であるのは, 低頻度語の除去により一部の文書が除外されたためである.

Dataset	Aug	$\|D\|$	*Len*	$\|V\|$
GoogleNewsT	-	922	2.99	307
	+	2766	2.25	341
Tweet	-	981	4.59	361
	+	2943	3.73	442
StackOverFlow	-	948	2.85	219
	+	2844	2.39	322

A.1 出力されたトピックの例

実験において実際に得られたトピックの一部を表 4 に示す. 表 4 から, GPT-3.5, GPT-4 は, LDA, TSCTM と比較して, より解釈しやすいトピック (“black friday deal shopping cyber” など) を出力していることがわかる.

A.2 出力された文書中に存在しない語の定性的分析

GoogleNewsT に対するトピックモデリングにおいて, モデルが出力した文書中に存在しない語の例を表 5 に示す. 太字で示された GPT-3.5, GPT-4 の出力は, 文書中に存在しない固有名詞 (*playstation, murdoch, philippines*) や実世界に存在しない語 (*coaliation*) であり, このような語はトピックの誤った解釈を誘因する可能性があるという意味で, 有害であると考えられる. しかし, このような語は GoogleNewsT に対するトピックモデリングにおいてのみ出力されており, Tweet および StackOverFlow に対しては出力されなかった.