InstructSum: 自然言語の指示による要約の生成制御

西田光甫　西田京介　斉藤いつみ　齋藤邦子
日本電信電話株式会社　NTT 人間情報研究所
kosuke.nishida.ap@hco.ntt.co.jp

概要

GPT-3 などの事前学習済み言語モデルは，訓練データを使うことなく指示を入力するだけで出力をタスクに適合させることができる．自然言語の指示による言語モデルの出力制御の従来研究では，複数のタスクの中の 1 つとして要約タスクへの適合を扱い，どのような要約を出力するかまで制御する取り組みがなかった．そこで本研究では，3309 個の要約タスクを持つ InstructSum データセットを作成した．さらに，長いソーステキストと指示の関係性を効率的にモデリングする手法を提案した．評価実験で InstructSum と提案手法の有効性を確認した．

1　はじめに

GPT-3 などの事前学習済み言語モデルは，言語モデルにタスク定義や入出力例を自然言語で指示することで，モデルの Fine-Tuning を行わずに出力を制御することを可能にした [1]．自然言語の指示による出力の制御は，非専門家や訓練データのないサービスに AI 開発の門戸を開いたことで，大きな注目を集めている．近年では，モデルが指示に従う能力を高めるため，大量のタスクを用意し，タスクごとに異なる指示を与えて教師ありマルチタスク学習を行う Instruction Tuning が提案された [2]．Instruction Tuning を加えた事前学習済み言語モデルは，指示を入力するだけで通常の訓練データを用いた教師あり学習モデルに匹敵する性能を達成することが報告されている [2, 3, 4]．

本研究では，Instruction Tuning を要約タスクに導入し，自然言語の指示によって要約を制御することに取り組む．第一の貢献として，要約に特化した Instruction Tuning のデータセットとして InstructSum を作成した．このデータセットでは，クラスタリングによって書き方が共通している要約の集合を得て，クラウドソーシングによって要約の書き方を自然言語の指示に書き下した．第二の貢献として，要約対象のテキスト（ソーステキスト）のみを先にエンコードして内部表現として保持することで，指示だけを高速にエンコード可能なモデルを実現した．Instruction Tuning の従来研究に比較して要約タスクでは入力長が長くなるが，提案モデルは精度を落とさずかつ効率的に指示とソーステキストを処理できる．評価実験により，InstructSum と提案手法の有効性を確認した．

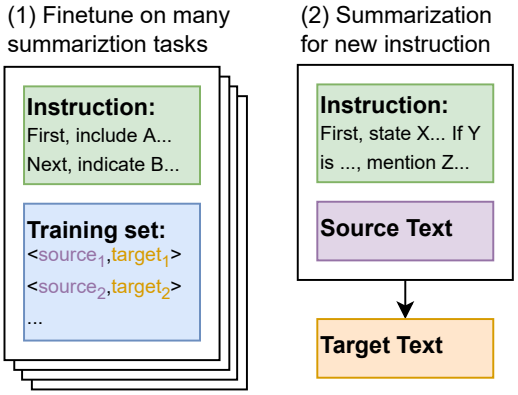

図 1　要約タスクにおける instruction-tuning.

2　InstructSum

2.1　問題定義

学習　ある要約タスクを自然言語で表現する指示 I が与えられる．各タスクの訓練データとして（ソーステキスト S，ターゲットテキスト $T_{S,I}$）のペアの集合が与えられる．複数個のタスクから要約モデルを学習する．

推論　要約モデルは，未知の指示（タスク）I と未知のソーステキスト S が与えられると，ターゲットテキスト $T_{S,I}$ を出力する．

2.2　データセットの収集

概要　前節で定義した問題定義に従うデータセットを構築するため，まず類似した書き方に基づくターゲットテキスト（および対応するソーステキス

ト）のクラスタを作成し，クラスタ内のターゲットテキストに共通する指示を作成する．ここで，クラスタ内のソース・ターゲットテキストの集合がタスクの訓練データに相当する．

ソース・ターゲットテキストの収集　[5] に従い，Wikipedia の概要部分をターゲットテキストとした．彼らは複数文書要約のために外部リンクの情報や Google 検索 API の結果を使ってソーステキストを収集したが，本研究では同じ記事の概要以外の部分をソーステキストとして利用した．

ターゲットテキストのクラスタリング　次にターゲットテキストをクラスタリングし，書き方が類似した記事のクラスタを得る．複数の概要部分に共通する下書きを作成するタスクを提案した [6] の手法を参考にした．まず記事タイトルを単語分割し，同じ箇所の 1 単語以外が一致するクラスタを作成する．次にターゲットテキストを単語分割し，クラスタ内の他のテキストに対してレーベンシュタイン距離を計算する．全てのテキストと単語長の 0.7 倍以上離れている場合にクラスタから削除する．クラスタの大きさが 4 以下の場合はクラスタを削除する．

指示文の作成　上記の手段で収集したクラスタのうち 900 クラスタ 8993 概要部分に対してアノテーションを付与した．まず，以下の操作を繰り返しクラスタに対して 5 つの指示文を作成する．この操作では，ワーカの負荷軽減のため 1 クラスタ当たり 5 つのターゲットテキストを読む．

- 同一クラスのキーワード（Donald Trump と Joe Biden など）を各ターゲットテキストから抜き出し，キーワードのクラス名（President's Name など）を考える．
- クラス名に言及する指示文を書き下す．
- クラスが現れる順番が共通する場合，順序に言及する．
- クラスを含むターゲットテキストが 3 つ以下である場合，そのクラスを含む条件に言及する．

指示文とターゲットテキストの関係情報の付与　次に，ターゲットテキストを文に分割したターゲット文と 5 個の指示文の関係フラグを付与する．関係フラグは，ある指示の内容をターゲット文が含んでいるかどうかを $\{0,1\}$ で表現する．

ここで，指示文 1 から指示文 j まで（$j = 1, \cdots 5$）を連結することで，1 つのクラスタから 5 通りの指示 I を作成する．関係フラグを用いて，指示 I の 1 指示文でも関連があるターゲット文を繋げたテキストを作成し，改めて指示 I とソーステキスト S に対応するターゲットテキスト $T_{S,I}$ と定義する．ここで，ターゲットテキストが連続した文でないときはデータセットから取り除く．

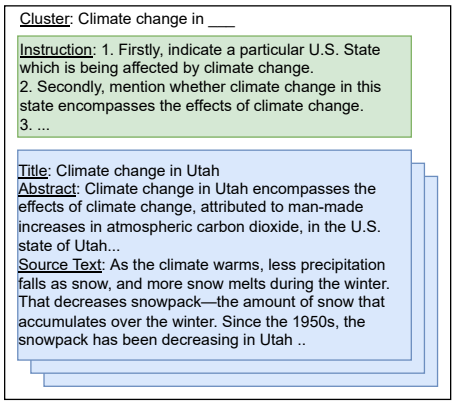

図 2　データセットの例．指示 I とターゲットテキスト $T_{S,I}$ に処理する前の指示文・Wikipedia 記事の例を示す．

表 1　統計値．

$\#S$	$\#I$	$\#T_{S,I}$	Ave(L_S)	Ave(L_I)	Ave($L_{T_{S,I}}$)
8993	3309	24344	1299.2	41.7	86.9

データの分割　クラスタ内の Wikipedia 記事のカテゴリ情報を用いて，評価セットに未知の指示が含まれるように 746/50/104 クラスタを訓練/開発/評価セットに分割した．詳細は付録に示す．

2.3　データ分析

図 2 にデータ例を示す．表 1 にデータの統計値を示す．指示，ソーステキスト，ターゲットテキストのトークン長を L_I, L_S, L_T と書く．

指示文　作成した 4500 指示文がどのような指示なのか調査するため，最初の単語の分布を調べた．If から始まる条件付きの指示が 13.0%，順序に言及する指示が 36.7%あるため，単純なキーワードクラスだけを指定するよりも複雑な指示ができる．残りの 50.4%はキーワードクラスを指定する指示であり，多様な表現の指示が含まれる．

ソース・ターゲットテキスト　ソース・ターゲットテキスト間の Rouge-L [7] を評価すると，28.1 であった．外部リンクページと Google 検索結果に基づきソーステキストを構築した Wikisum [5] は Rouge-L を 17.0 と報告しており，InstructSum は WikiSum に比べてテキスト間の整合性が高いデータセットであると言える．ソース・ターゲットテキストの長さ 1299.2 と 86.9 は要約データセットとしては標準的か少し長い程度である．代表的なデータ

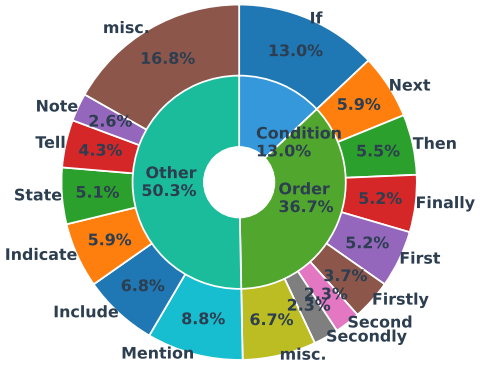

図 3　指示の分類と最初の単語の分布．

セットである CNN/DM [8] は 789.9 と 55.6 である．

3　モデル

同一のソーステキストに異なる指示を与えて複数回要約を生成する現実的な設定において，ソーステキストをモデルの内部表現として保持して計算効率を高める手法を提案する．Transformer 構造 [9] の Encoder-Decoder である T5 [10] をベースとする．

入出力形式　モデルの入力は，'Instructions: {I} According to the above instructions, summarize the following article. Title: {Ti} Article: {S}' とした．ここで，波括弧は代入操作，Ti は Wikipedia 記事のタイトルを示す．出力はターゲットテキストのみである．入力は最初の 1024 トークンのみを用いた．

内部表現保持モデル　まず，Transformer の self-attention について再確認を行う．Query, Key, Value の行列 $Q, K, V \in \mathbb{R}^{L \times d}$ を使って隠れ状態

$$H = \mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

を得る．L は系列長，d は次元サイズである．この計算量が Transformer Encoder のボトルネックであり，$\mathcal{O}((L_S + L_I)^2)$ の計算量が必要である．

そこで，提案モデルでは Q, K, V を指示 I の系列 $Q_I, K_I, V_I \in \mathbb{R}^{L_I \times d}$ とソーステキスト S の系列 $Q_S, K_S, V_S \in \mathbb{R}^{L_S \times d}$ に分割する．ソーステキストを $H_S = \mathrm{Attn}(Q_S, K_S, V_S)$ とエンコードする．K_S, V_S はソーステキスト表現としてモデルが保持する．

指示が入力されたときは $H_I = \mathrm{Attn}(Q_I, K, V)$ とエンコードできる．K_I, V_I は K_S, V_S と繋げることで K, V とする．この操作では，最も大きい $\mathcal{O}(L_S^2)$ の計算を 2 回目以降の指示入力で省略できる．

学習・推論　学習時の損失は Teacher-Forcing と Cross-Entropy によって計算した．推論時の生成はビーム幅を 2 とした Beam-Search で行った．

表 2　自動評価値．Zero-shot は InstructSum による学習を行わない．No Inst. は指示 I を入力しない．Full は通常の Transformer モデル（full self-attention）である．

	# Param.	R-L	B-4
FLAN-T5-Base (Zero-shot)	250M	13.35	0.61
FLAN-T5-Base (Trained, No Inst.)	250M	36.60	7.63
FLAN-T5-Base (Trained, Full)	250M	**42.38**	**13.53**
FLAN-T5-Base (Trained, Proposed)	250M	41.11	12.37
FLAN-T5-Large (Zero-shot)	780M	20.05	2.00
FLAN-T5-Large (Trained, No Inst.)	780M	38.41	11.12
FLAN-T5-Large (Trained, Full)	780M	**45.56**	**16.40**
FLAN-T5-Large (Trained, Proposed)	780M	44.93	16.31
FLAN-T5-XL (Zero-shot)	3B	21.06	2.68
FLAN-T5-XXL (Zero-shot)	11B	27.90	5.32

4　評価実験

ベースモデルである T5 に対して Instruction Tuning を実施した FLAN-T5 [4] を初期値として InstructSum で学習した．評価指標には，要約タスクで一般的な Rouge-L（R-L） [7]，BLEU-4（B-4） [11] を用いた．その他の設定の詳細は付録に示す．

4.1　結果と議論

InstructSum により指示型要約の性能は向上するか？　表 2 に結果を示す．InstructSum で訓練を行うことで，Base, Large モデルであっても Zero-shot の XXL モデルの性能を大きく上回った．InstructSum による訓練は，既に Instruction Tuning 済の FLAN-T5 を初期値とした場合でも指示によって要約を制御する能力を高める効果があった．また，RTX8000 4GPUs で計算した場合，XXL モデルの評価データの推論時間が 161 分に対し Large モデルの訓練・推論時間は計 94 分であるから，時間・空間計算量の観点からも InstructSum の利用は有用である．

内部表現の保持により性能を維持し効率性を高められるか？　内部表現保持モデルを用いた場合でも，Large モデルの Rouge-L では 0.63，BLEU-4 では 0.09 ポイントと，性能が僅かに下がるに留まった．指示を入力した Large モデル Encoder の演算数は提案手法を用いることで Base モデルの 174GigaFLOPS より小さい 24GigaFLOPS に抑えられた．入力 1024 トークンのうち指示は冒頭の平均 41 トークンであるため，通常の self-attention は計算量が多いにも関わらずソーステキストと指示の離れた位置間の相互作用のモデリングへの貢献が少なく，提案手法が精

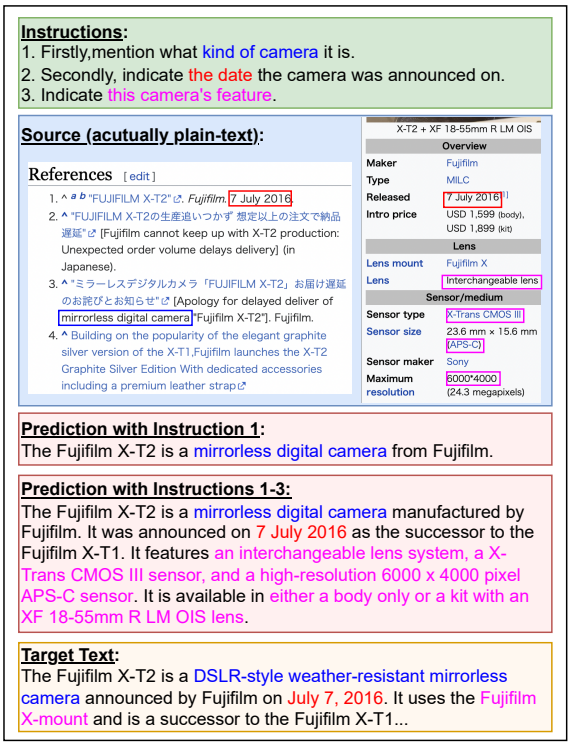

図 4　提案モデルの生成例．ソース・ターゲットテキストの全文は https://en.wikipedia.org/wiki/Fujifilm_X-T2 にて確認できる．

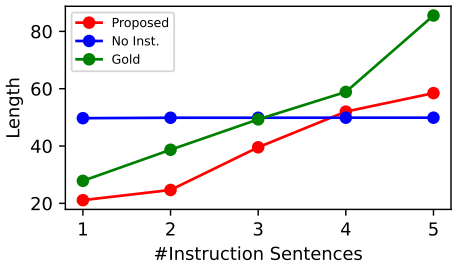

図 5　指示文数ごとの平均長．

度・計算量の面でバランスが良いと言える．

指示によって生成結果は変化するか？　表 2 に示す様に，指示を入力せずに InstructSum で訓練・評価した場合，性能が下落した．次に，図 4 に生成例を示す．要約の制御を学習する Instruction Tuning により，提案手法は指示の内容に従う要約を生成出来ている．さらに，図 5・図 6 に I の指示文数ごとに評価した結果を示す．提案手法は，正解要約の長さに合わせて生成要約の長さも変化している．自動評価値は短い指示・正解要約に対して性能が高い．特に precision のみで計算する BLEU-4 で短い正解要約に対して性能が高いことから，指示に対して簡潔に要約を生成していると言える．指示を入力しない場合はソーステキストのみに依存するため，平均的な長

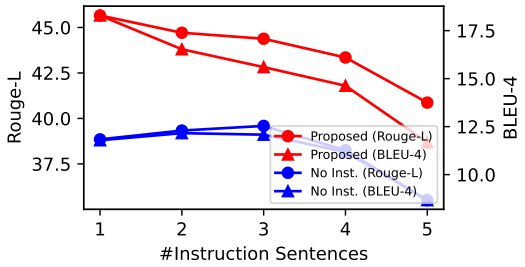

図 6　指示文数ごとの自動評価値．

さの出力を学習してしまう．

5　関連研究

クエリ依存要約　本研究と最も関わりの深いタスクである [12, 13, 14, 15, 16, 17]．しかし，1 つの指示（タスク，クエリ）に複数のソース・ターゲットテキストのペアがあり，さらに複数タスクの学習ができるという点で本研究は新しく，指示による要約の制御の研究に適している．

要約の生成制御　自然言語の指示・クエリ以外にもキーワード [18, 19] や長さ [20, 21, 22] に関して要約の生成を制御する研究があるが，InstructSum は自然文で要約の内容や記載順序を指示できる点に新しさがある．一方で，長さは指示数で間接的に制御するのみであるため，今後の課題と言える．

Instruction Tuning　Instruction Tuning のために多くのタスクを含むデータセットが提案されており [2, 23, 24, 25, 26, 27, 28]，それらでは要約データセットも用いられている．しかし，これらの研究では要約タスク自体への適合を学習する．InstructSum は要約の制御のための Instruction Tuning に取り組んだ初めての研究である．

6　おわりに

本研究では，要約における Instruction Tuning を実現するため，InstructSum の作成と内部表現保持モデルの提案を行った．本研究の貢献を以下に示す．

本研究の独自性．　InstructSum は，ターゲットテキストのクラスタリングと組み合わせることで要約における Instruction Tuning に初めて取り組んだ．

本研究の重要性．　指示による要約の制御の実現により，自動要約の事業領域を広げることができる．InstructSum 及び内部表現保持モデルは，この実現に資すると考える．Instruction Tuning は ChatGPT [29] にも利用されており，言語モデル研究における重要技術として注目されている．InstructSum は Instruction Tuning の高度化・評価に貢献できる．

参考文献

[1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. **arXiv preprint arXiv:2005.14165**, 2020.

[2] 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.

[3] Long Ouyang et al. Training language models to follow instructions with human feedback. In **NeurIPS**, 2022.

[4] Hyung Won Chung et al. Scaling instruction-finetuned language models. **arXiv preprint arXiv:2210.11416**, 2022.

[5] Peter J. Liu*, Mohammad Saleh*, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In **ICLR**, 2018.

[6] Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, and Bill Dolan. Automatic document sketching: Generating drafts from analogous texts. In **ACL-IJCNLP (Findings)**, pp. 2102–2113, 2021.

[7] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, 2004.

[8] Karl Moritz Hermann et al. Teaching machines to read and comprehend. In **NIPS**, p. 1693–1701, 2015.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **NIPS**, pp. 5998–6008, 2017.

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **JMLR**, Vol. 21, pp. 1–67, 2020.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **ACL**, pp. 311–318, 2002.

[12] Hoa Trang Dang. Overview of duc 2005. In **document understanding conference**, Vol. 2005, pp. 1–12, 2005.

[13] Hoa Trang Dang. DUC 2005: Evaluation of question-focused summarization systems. In **Task-Focused Summarization and Question Answering**, pp. 48–55, 2006.

[14] Preksha Nema et al. Diversity driven attention model for query-based abstractive summarization. In **ACL**, pp. 1063–1072, 2017.

[15] Haichao Zhu, Li Dong, Furu Wei, Bing Qin, and Ting Liu. Transforming wikipedia into augmented data for query-focused summarization. **TASLP**, Vol. 30, pp. 2357–2367, 2022.

[16] Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. AQuaMuSe: Automatically generating datasets for query-based multi-document summarization. **arXiv preprint arXiv:2010.12694**, 2020.

[17] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir R. Radev. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In **NAACL-HLT**, pp. 5905–5921, 2021.

[18] Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. **arXiv preprint arXiv:2003.13028**, 2020.

[19] Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Fatema Rajani, and Caiming Xiong. Ctrlsum: Towards generic controllable text summarization. **arXiv preprint arXiv:2012.04281**, 2020.

[20] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In **EMNLP**, pp. 1328–1338, 2016.

[21] Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, Atsushi Otsuka, Hisako Asano, Junji Tomita, Hiroyuki Shindo, and Yuji Matsumoto. Length-controllable abstractive summarization by guiding with summary prototype. **arXiv preprint arXiv:2001.07331**, 2020.

[22] Yizhu Liu, Qi Jia, and Kenny Q. Zhu. Length control in abstractive summarization by pretraining information selection. In **ACL**, pp. 6885–6895, 2022.

[23] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. **arXiv preprint arXiv:2104.08773**, 2021.

[24] Yizhong Wang et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ tasks. In **EMNLP**, 2022.

[25] Victor Sanh et al. Multitask prompted training enables zero-shot task generalization. In **ICLR**, 2022.

[26] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In **ICLR**, 2022.

[27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In **ICLR**, 2021.

[28] Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. **arXiv preprint arXiv:2003.13028**, 2022.

[29] OpenAI. Chatgpt: Optimizing language models for dialogue, 2022. https://openai.com/blog/chatgpt/.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

[31] Adam Paszke et al. Automatic differentiation in pytorch. In **Autodiff**, 2017.

[32] Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In **ACL: System Demonstrations**, pp. 38–45, 2020.

A InstructSum

A.1 データ収集

Wikipedia は CirrusSearch の 2021 年 10 月 11 日の dump データを用いた [1)]．CirrusSearch は Wikipedia をテキストに前処理したデータを提供しているため，表から自動変換したテキストもソーステキストに含めることができる．

A.2 データ分布とデータスプリット

図 8 にソース・ターゲットテキストで作成したワードクラウドを示す．地名違いや年度違いのタイトルの記事のクラスタが作成されやすいため，特に地理・政治・スポーツのデータ数が多くなっている．

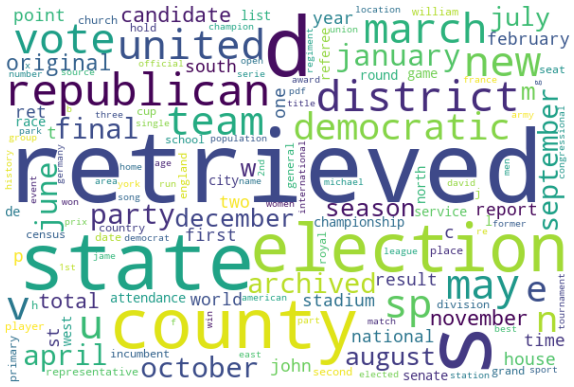

図 7 ソーステキストのワードクラウド．

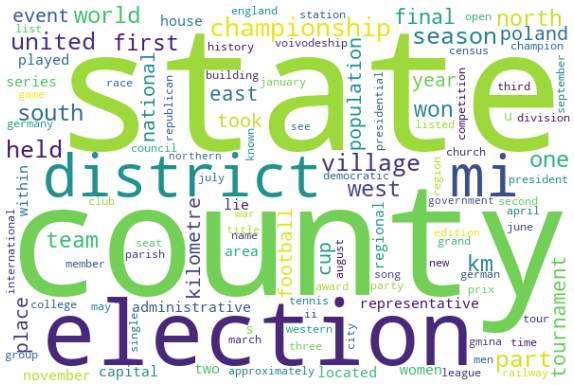

図 8 ターゲットテキストのワードクラウド．

データセットが含む Wikipedia ページについて，Cirrus Search が自動分類したカテゴリの分布を表 3 に示す．1 つのページは複数のカテゴリに属する．

本研究では未知の指示に従う能力を評価するため，Geography カテゴリをクラスタ内に含まない 104 クラスタを評価セットとした．残り 796 クラスタのうちランダムな 50 クラスタを開発セット，残りを訓練セットとした．

表 3 Wikipedia 記事が属するカテゴリ/サブカテゴリのトップ 10.

Geography/Regions	6920
Culture/Sports	2808
Culture/Media	2157
Culture/Biography	1969
History and Society/Politics and Government	1458
Culture/Media	958
Geography/Geographical	833
STEM/STEM	629
Culture/Visual Arts	523
History and Society/Society	485
History and Society/Transportation	431

B 実験

B.1 実装

実験には NVIDIA Quadro RTX 8000 (48GB) GPUs 4 枚を用いた．ハイパーパラメータを表 4 に示す．訓練時は入力長をデフォルトの 512 として学習し，推論時に T5 の relative positional embedding の最大入力長を 1024 に変更した．最適化手法には Adam [30] を用いた．実装には PyTorch [31] と Transformers [32] を用いた．指示を入力しないときの入力は ‘Summarize the following article. Title: {*Ti* } Article: {*S*}’ とした．図 5 と図 6 の実験では，データ数確保のため，2.2 節の最後で行ったターゲットテキストが不連続な場合のデータ削除を評価データでは行わなかった．

表 4 ハイパーパラメータ．

Batch Size	256
# Steps	500
Learning Rate	5e-5
Max Target Length	192
Min Target Length	0

1） https://dumps.wikimedia.org/other/cirrussearch/