事前学習済みモデル T5 における近傍分布の有効性の調査

丹羽彩奈 [1]　岡崎直観 [1]
[1] 東京工業大学情報理工学院
{ayana.niwa[at]nlp, naoaki.okazaki[at]}c.titech.ac.jp

概要

大規模なデータストアから近傍事例を検索し，テキスト生成モデルを補強するメモリ拡張手法が有望視されている。特に最近では、時間ステップごとのモデルの予測分布を近傍事例に基づく分布で補間することで追加の学習なしに機械翻訳性能を向上させるシンプルかつ効果的なアプローチ（kNN-MT）が提案された [1]。本研究は、そのアプローチを学習コストの観点から捉え、事前学習済みモデルに k 近傍モデルを組み込むことで少ない学習コストで性能を向上させることを目的とする。本稿ではそのための一歩として、kNN-MT の枠組みを事前学習済みのエンコーダ・デコーダモデル T5 に導入し、様々な入出力設定をもつ自然言語生成タスクや、zero-shot などの複数の学習設定における近傍事例の有効性を調査した。その結果、事前学習で得た生成能力と近傍分布を組み合わせることで、少ない学習コストでタスク特化の生成を行える可能性が示唆された。

1　はじめに

メモリ拡張モデルは、ベースとなるモデルと大規模なデータストアから検索した近傍事例（kNN）を組み合わせて最終出力を決定する。このアプローチではモデルパラメータの学習による暗黙的な記憶とデータストアによる明示的な記憶の両方が性能向上に寄与するため [2]、品詞タグ付け [3] や構文的曖昧性解消 [4]、用例ベース機械翻訳 [5] など、幅広いタスクで活用されてきた。

特に最近では、デコーダのみからなる言語モデル GPT-2 [6] の時間ステップごとの予測分布を k 近傍モデルで補間する kNN-LM が追加の学習なしに性能を向上させることに成功し、後続の研究に大きなインパクトを与えた [2]。kNN-LM のアプローチはエンコーダ・デコーダモデルに拡張され、言語モデリングだけではなく機械翻訳（kNN-MT）[1, 7] や自然言語推論 [8]、文法誤り訂正 [9] など個別のタスクに適用されている。

これらのアプローチは、特に低頻度語などの学習時に覚えきれなかった知識に有効であり、モデルがもつ知識を補間できることが性能向上のひとつの要因であると指摘されている [2]。この知識の補間は、近年自然言語処理タスクで主流になりつつある、事前学習時とは異なるタスク特化の知識を fine-tuning で獲得するアプローチと共通する考え方である。そこで、本研究では fine-tuning で得るタスクの知識を近傍分布で補間することで学習コストを削減することを目的とし、事前学習済みのエンコーダ・デコーダモデルに kNN-MT のアプローチを組み込む。

本稿は、機械翻訳や自動要約などあらゆる自然言語処理タスクを系列から系列への変換としてみなして統一的な枠組みで扱う T5 モデル [10] をベースのモデルとして採用する。そして、この T5 モデルをさまざまな自然言語生成タスクに適応させる際に、近傍分布を活用してタスク特化の知識を補間する。既存研究では同一タスク内でのドメイン適応に対する有効性が示されてきた。本研究は事前学習済みエンコーダ・デコーダモデルを別タスクに適応するために近傍分布を活用するはじめての研究である。

本項では、そのための一歩として、まず kNN-MT の枠組みを組み込んだ T5 を機械翻訳だけではなく自動要約、対話生成、Data-to-text など様々な入出力をもつタスクに適用し、その近傍分布の有効性を調べた。また、学習コスト削減に関する有効性について調査するため、全学習事例を用いた fine-tuning だけではなく、zero-shot や few-shot 設定において近傍分布を組み込み、少ない学習でより良い性能を達成するアプローチが有望か調査した。実験結果より、現状では近傍分布が有効なタスクは限定的であるものの、翻訳タスクでは T5 においても比較的小規模なデータストアで性能を改善できることを確認した。また、特定のタスクにおいて近傍分布を活用することで、少量の学習事例を用いた few-shot の性能を全学習事例で fine-tuning した場合の性能に近づけ

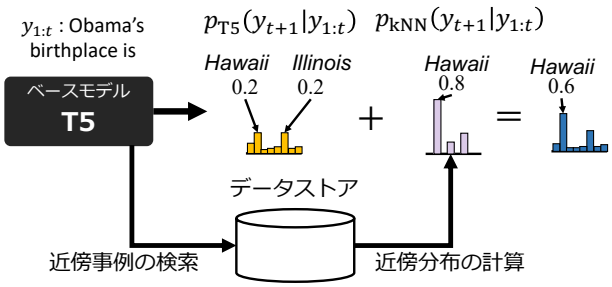

図 1　生成済みの系列 "Obama's birthplace is" に続くトークンを予測する際の手法の概要図

られることを確認した。

2　近傍分布を用いたテキスト生成

本研究では、 Khandelwal ら [1] が提案した近傍事例を用いたモデル kNN-MT を T5 に適用し、任意のテキスト生成タスクに用いる（図 1）。ベースとなる自己回帰モデルは、入力系列 x と既に生成済みの系列 $\hat{y}_{1:i-1}$ に基づき、i 番目のトークンの確率分布 $p(y_i|x, \hat{y}_{1:i-1})$ を予測する。

2.1　kNN-MT

kNN-MT は、k 近傍モデルをエンコーダ・デコーダアーキテクチャに統合した機械翻訳のための Transformer モデルである [1]。学習済みの翻訳モデルの出力分布 $p_{\mathrm{MT}}(y|x)$ をノンパラメトリックな k 近傍モデルによって求めた近傍分布 $p_{\mathrm{kNN}}(y|x)$ で線形補間することで最終的な分布 $p(y_i|x, \hat{y}_{1:i-1})$ を求める。ここで、$\hat{y}$ は既に生成されたトークンを表す。

$$p(y_i|x, \hat{y}_{1:i-1}) = \lambda p_{\mathrm{kNN}}(y_i|x, \hat{y}_{1:i-1}) + (1-\lambda) p_{\mathrm{MT}}(y_i|x, \hat{y}_{1:i-1}) \quad (1)$$

ここでハイパーパラメータ $0 \le \lambda \le 1$ は補完係数であり、値が大きいほど近傍分布が最終的な分布に大きな影響を及ぼす。

データストアの構築　データストア $(\mathcal{K}, \mathcal{V})$ は、全ての学習事例 $(\mathcal{X}, \mathcal{Y})$ のトークンごとの隠れベクトル $\boldsymbol{h}(x, \hat{y}_{1:i-1})$ をキー、その隠れベクトルに対応する正解トークン y_i を値とするペアの集合である。

$$(\mathcal{K}, \mathcal{V}) = \{(\boldsymbol{h}(x, y_{1:i-1}), y_i) \mid \forall y_i \in y, y \in \mathcal{Y}\} \quad (2)$$

近傍分布 p_{kNN} の計算　まず、モデルは入力系列 x と生成済みの系列 $\hat{y}_{1:i-1}$ を受け取り、隠れベクトル $\boldsymbol{h}(x, \hat{y}_{1:i-1})$ を語彙数次元のベクトルに線形変換することで次の単語に対する出力分布 $p_{\mathrm{MT}}(y_i|x, \hat{y}_{1:i-1})$ を得る。次に、k 近傍モデルが $\boldsymbol{h}(x, \hat{y}_{1:i-1})$ をクエリとしてデータストアから距離関数 $d(\cdot,\cdot)$ に従って近さ Top-k の近傍事例 $\mathcal{N} = \{(\boldsymbol{k}_j, v_j) \in (\mathcal{K}, \mathcal{V})\}_{j=1}^{k}$ を検索する。この距離関数 $d(\cdot,\cdot)$ には L^2 距離を用いる。最後に、その負の距離に温度 T のソフトマックス関数を適用することで、語彙数次元の近傍分布 $p_{\mathrm{kNN}}(y_i|x, \hat{y}_{1:i-1})$ を得る。

$$p_{kNN}(y_i|x, \hat{y}_{1:i-1}) \propto \sum_{(\boldsymbol{k}_j, v_j) \in \mathcal{N}} \mathbb{I}_{v_j = y_i} \exp\left(\frac{-d(\boldsymbol{k}_j, \boldsymbol{h}(x, \hat{y}_{1:i-1}))}{T}\right) \quad (3)$$

2.2　kNN-T5

本研究では、事前学習済みモデルに k 近傍モデルを統合して低コストで良い性能を達成することを目的とする。ベースとなる事前学習済みのエンコーダ・デコーダモデルには、Text-to-Text Transfer Transformer (T5) [10] を採用する。T5 は、教師なしタスクと教師ありタスクで事前学習された Transformer ベース [11] のモデルであり、あらゆる自然言語タスクを text-to-text タスクとして見なし、プレフィックスを用いることで生成タスクを切り替える。

kNN-MT のベースである Transformer モデルを T5 モデルに置き換え、各ステップのモデルの出力分布 $p_{\mathrm{T5}}(y_i|x, \hat{y}_{1:i-1})$ を近傍分布で補間することで最終的な分布を求める。推論時およびデータストア構築に用いる隠れベクトル $\boldsymbol{h}(x, \hat{y}_{1:i-1})$ の計算時には、入力系列の先頭にタスクごとのプレフィックス（例：英独翻訳では "`translate English to German:` "）を連結してモデルに入力する [1)]。隠れベクトルにはデコーダ側のフィードフォワード層の最終層への入力を用いる。データストアの構築方法や近傍分布の計算方法は kNN-MT に従う。

3　実験

実験では、事前学習された T5 モデルに対する近傍分布の有効性について、幅広い自然言語生成タスクと学習設定で調査する。

3.1　実験設定

データセット　自然言語生成タスクのベンチマークである GEM [2)] から自動要約（XSum データセット、以下 XSum [12]）、対話生成（Schema-Guided Dialog データセット、以下 Dialog [13]）、Data-to-Text（DART データセット、以下 DART [14]）のタスクを

1)　出力系列に変更はないため、データストアのサイズは変わりない。
2)　https://gem-benchmark.com/

表 1 実験で用いるデータセットの統計値

	学習データ	開発データ	評価データ
XSum	23,206	1,117	1,166
Dialog	164,982	10,000	10,000
DART	62,659	2,768	5,097
WMT'16	4,548,885	2,169	2,999

選定した。XSum は短い単文への自動要約、Dialog は直前の発話とエージェントの対話行為から応答を生成するタスク志向型の対話生成、DART は Wikipedia に含まれるテーブルからのテキスト生成である。表 1 に示した各データセットの統計情報の通り、これらは比較的小規模なデータセットである。さらに、当ベンチマークセットには含まれない生成タスクとして機械翻訳（WMT'16 英独）も採用する [3]。これらのデータは全て Huggingface Datasets [4] からダウンロードした。

モデル 二種類のサイズの T5 モデルである t5-small と t5-base を用いる[5]。パラメータサイズはそれぞれ 60M、220M である。入出力設定が特殊な Dialog、DART、XSum はそれぞれ fine-tuning を行った。学習や推論の詳細については付録 B に示した。実装は全て Huggingface Transformers [6] で行った。

近傍分布の統合 近傍探索ライブラリ faiss [15] を用いてデータストアから近傍事例を検索した。なお、統合方法に関するハイパーパラメータ (k, T, λ) は 、$k \in \{32, 128, 256, 512\}$、$T \in \{10, 50, 100, 200\}$、$\lambda \in \{0.2, 0.4, 0.6, 0.8\}$ の組み合わせから開発データでの性能に基づき決定した。

評価 XSum、Dialog、DART は GEM-metrics[7] で求めた BLEU [16]、ChrF [17] と ROUGE [18] で評価した。WMT'16 英独は、既存研究と同様に BLEU で評価する。BLEU の算出では SacreBLEU [19] を用い、大文字小文字を区別した。

3.2 実験結果

まず、全ての学習事例で fine-tuning した場合 [8] の近傍分布の統合なしとあり（+kNN）の場合の性能と近傍分布の統合方法に関するハイパーパラメータ、またデータストアのサイズ D を表 2 に示した。結果として、事前学習済みエンコーダ・デコーダモデルにおける近傍分布の有効性は、タスクによって大きく異なる傾向を示した。まず、自動要約・対話生成・Data-to-text タスクでは近傍分布が性能向上にほとんど寄与しない。逆に既存研究でも有効性が示されている機械翻訳タスクでは 2.6 ポイント以上 BLEU 値が向上した。機械翻訳タスクとその他のタスクの最も明示的な違いは、機械翻訳タスクの 1/50 以下しかないデータストアの小ささである。既存研究 [1] で用いられている WMT'19 独英のデータストアのサイズが 770M であることを踏まえても、今回近傍分布が有効でなかった 3 つのタスクのデータストアは相対的に著しく小さい。そこで、データストアのサイズが WMT'16 英独の性能に与える影響を調べるため、データストアのサイズを小さくした時の BLEU 値の推移を調べた（図 2）。これによると、データストアの大きさと性能の間に相関関係があることが確認できる。よって、事前学習モデルに近傍分布を統合する際にもデータストアの大きさは重要であることが示唆される。しかしながら、任意のタスクでデータストアのサイズを大きくすることは容易ではない。

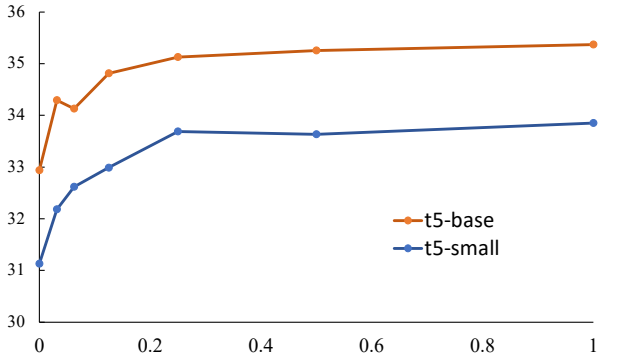

図 2 用いるデータストアの事例の割合 (横軸) を変化させた時の WMT'16 英独の評価データでの BLEU 値（縦軸）

そこで、次はデータストアのサイズを大きくすることなく事前学習済みモデルで近傍分布を有効に活用する方法について調査する。具体的には、事前学習時の知識のみを使ってベースモデルの出力分布 p_{T5} を計算させ、タスクの知識を近傍分布のみから得る zero-shot 設定での性能を調べた（表 3）。その結果、zero-shot 設定の性能もタスクによって大きな違いが見られた。特に顕著なのは、Dialog と DART で近傍分布を統合した場合の性能が大きく向上した点である。Dialog では自然言語文に加えてエージェ

3） 機械翻訳は WMT の Shared task でのベンチマークとして確立しているため GEM から除外されている。
4） https://github.com/huggingface/datasets
5） https://huggingface.co/t5-small, https://huggingface.co/t5-base
6） https://github.com/huggingface/transformers
7） https://github.com/GEM-benchmark/GEM-metrics
8） 事前学習に含まれる翻訳タスクは除く

表 2 自然言語生成タスクでの性能と近傍分布に関する最適なハイパーパラメータおよびデータストアのサイズ D

タスク	自動要約			対話生成			Data-to-text			機械翻訳
データセット	XSum			Schema guided dialog			DART			WMT'16 英独
評価指標	R-1	R-2	R-L	BLEU	ChrF	ROUGE	BLEU	ChrF	ROUGE	BLEU
t5-small	0.343	0.263	0.118	33.96	49.89	0.522	45.99	61.94	0.587	31.13
+ kNN	0.343	0.266	0.123	34.13	49.94	0.525	44.90	61.40	0.578	**33.85**
(k, T, λ)	(512, 50, 0.4)			(128, 50, 0.4)			(512, 100, 0.6)			(128, 50, 0.4)
t5-base	0.401	0.168	0.316	34.48	50.49	0.527	48.81	64.83	0.610	32.94
+ kNN	0.407	0.175	0.324	34.98	50.77	0.531	48.22	64.32	0.602	**35.37**
(k, T, λ)	(32, 100, 0.2)			(512, 200, 0.4)			(128, 200, 0.6)			(32, 200, 0.4)
D	0.66M			2.49M			1.71M			125.41M

表 3 t5-small の実験。B は BLEU 値、R は ROUGE 値を示す。$\lambda = 1$ は近傍分布のみを用いてテキストを生成した場合の性能である。

タスク	自動要約		対話生成		Data-to-text	
評価指標	R-2	R-L	B	R	B	R
zero-shot	0.043	0.152	3.10	0.150	5.59	0.207
+ kNN	0.073	0.205	17.43	0.357	20.33	0.321
($\lambda = 1.0$)	0.069	0.190	17.84	0.362	20.30	0.314
(k, T, λ)	(128, 100, 0.8)		(32, 50, 0.8)		(32, 50, 0.8)	
few-shot	0.043	0.152	11.57	0.225	5.59	0.208
+ kNN	0.070	0.201	26.20	0.453	20.69	0.322
($\lambda = 1.0$)	0.066	0.190	25.89	0.452	20.42	0.314
(k, T, λ)	(128, 100, 0.8)		(32, 100, 0.8)		(32, 50, 0.8)	

ントの対話行為情報が、DART では自然言語文ではなく構造化データが入力として与えられる。そのため、両タスクとも T5 の事前学習時の入出力設定とは異なり生成の難易度が高く、zero-shot ではほとんど解くことができない。しかし、近傍分布を統合することで事前学習時にはないタスク固有の知識を追加学習なしに活用できるようになる。一方で、これらの性能は $\lambda = 1$、つまり近傍分布のみで生成した結果とほぼ変わらない性能であり、事前学習で獲得したテキスト生成能力をほとんど使えていない。そのため fine-tuning ありの場合に比べて性能は大きく劣り、実用性に欠ける。

そこで、次は学習コストを抑えつつベースモデルの生成能力と近傍分布の両方を有効活用する設定を探るべく、ランダムサンプリングした 32 件の学習事例を用いて few-shot 学習を行った。結果を表 3 の下半分に示した。特筆すべきは、Dialog で少数の学習を追加することでベースのモデルの BLEU 値が近傍分布なし・ありともに 8.5 ポイント以上向上した点である。これは、入力に自然言語文が含まれることから比較的事前学習された知識を転移しやすい設定であるためだと考えられる。学習コストについて考えると、全ての学習データで学習する場合に比べてステップ数を 1/250、延べ学習事例数を 1/4000 にまで削減しつつ、GEM のベースラインの報告値である 50.0（ROUGE-L）[9] に対して差を 5 ポイント以下にまで近づけられたことになる。つまり学習コストを減らすという目的で近傍分布を組み込むことが少なくとも特定の条件下では有望であることがわかった。近傍分布の有効性は顕著である一方で、ベースモデルの性能は依然として低く、学習コストを下げることで性能を必要以上に犠牲にしてしまう。また、XSum と DART では少量の学習事例だけではモデルの生成能力を引き出すことはできなかった。事前学習で獲得した生成能力と近傍分布を幅広い入出力設定のタスクにおいて活用できる few-shot 手法の検討は今後の課題である。

4 結論

本研究では、メモリ拡張モデル kNN-MT の枠組みを事前学習済みのエンコーダ・デコーダモデル T5 に適用し、任意のテキスト生成に適用可能な手法として一般化した上で、低学習コストで高い性能を達成するための近傍分布の活用法の検討に取り組んだ。機械翻訳、自動要約、対話生成、Data-to-text という幅広い入出力の自然言語生成タスクを用いた実験により、近傍分布の統合は少なくとも特定のタスク設定においては有効であり、学習コストを大幅に抑えつつ事前学習済みモデルを各タスクにノンパラメトリックに適応させられる可能性が示された。今後は、近傍分布が有効であるタスクの性質の更なる分析や、学習コストと性能のトレードオフを改善する近傍分布の統合方法について調査を進めたい。

9) https://gem-benchmark.com/results

謝辞

本研究成果は、国立研究開発法人情報通信研究機構（NICT）の委託研究「自動翻訳の精度向上のためのマルチモーダル情報の外部制御可能なモデリングの研究開発」（課題 225）により得られたものです。

参考文献

[1] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In **International Conference on Learning Representations**, 2021.

[2] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In **International Conference on Learning Representations**, 2020.

[3] Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. MBT: A memory-based part of speech tagger-generator. In **Fourth Workshop on Very Large Corpora**, Herstmonceux Castle, Sussex, UK, June 1996.

[4] Claire Cardie. Domain-specific knowledge acquisition for conceptual sentence analysis. **Computer Science Department Faculty Publication Series**, p. 60, 1994.

[5] Eiichiro Sumita and Hitoshi Iida. Experiments and prospects of example-based machine translation. In **29th Annual Meeting of the Association for Computational Linguistics**, pp. 185–192, Berkeley, California, USA, June 1991.

[6] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[7] Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. Fast nearest neighbor machine translation. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 555–565, Dublin, Ireland, May 2022.

[8] Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. knn-prompt: Nearest neighbor zero-shot inference. arXiv, 2022.

[9] Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. Interpretability for language learners using example-based grammatical error correction. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, pp. 7176–7187, Dublin, Ireland, May 2022.

[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. **J. Mach. Learn. Res.**, Vol. 21, No. 1, jun 2022.

[11] 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**, pp. 5998–6008, 2017.

[12] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 1797–1807, Brussels, Belgium, October-November 2018.

[13] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, No. 05, pp. 8689–8696, Apr. 2020.

[14] Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. DART: Open-domain structured data record to text generation. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 432–447, Online, June 2021.

[15] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, Vol. 7, No. 3, pp. 535–547, 2019.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In **Proceedings of the 40th Annual Meeting on Association for Computational Linguistics**, p. 311–318, USA, 2002.

[17] Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In **Proceedings of the Tenth Workshop on Statistical Machine Translation**, pp. 392–395, Lisbon, Portugal, September 2015.

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

[19] Matt Post. A call for clarity in reporting bleu scores. **WMT 2018**, p. 186, 2018.

[20] Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. Investigating pretrained language models for graph-to-text generation. In **Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI**, pp. 211–227, Online, November 2021.

A　実験設定の詳細

実験で用いる入力系列の設定　DART の入力系列は、Wikipedia のテーブルに含まれる主語、述語、目的語の三つ組のセットを既存研究 [20] と同様に各要素の前に特殊トークン<H>, <R>, <T>を付けて系列化した。また、T5 モデルに用いたプロンプトは以下の通りである。

自動要約　“summarize:”

対話生成　“Prompt: _____, Response Type: _____, Type of Slot: _____, Agent: _____”

Data-to-text　“translate graph to English:”

機械翻訳　“translate English to German:”

対話生成のプロンプトは、ベンチマーク GEM で推奨されているものを用いた。

B　実験の詳細

B.1　学習と推論のパラメータ設定

XSum はバッチサイズ 16、学習率 2e-05 で 50 エポック、Dialog はバッチサイズ 256、学習率 0.001 で 5000 ステップ、DART はバッチサイズ 24、学習率 7e-05 で 40 エポック学習した。また、推論時のビーム幅は全て 5 に設定した。

B.2　zero-shot 学習に近傍分布を組み込む際の留意点

近傍分布のハイパーパラメータである k, T, λ をそれぞれ固定した時の全学習事例による fine-tuning あり（full）の設定と zero-shot（zero）の設定の BLEU 値の分散値の比率 $\mathrm{Var}(\mathrm{BLEU}_{\mathrm{zero}})/\mathrm{Var}(\mathrm{BLEU}_{\mathrm{full}})$ をを図 4 に示した。値が大きいほど zero-shot における性能がハイパーパラメータに敏感であることを示す。これによると、データセット間で違いがあるが特に Dialog と DART の zero-shot における性能が両パラメータに大変敏感であり、温度パラメータや近傍事例数によって大きく性能が異なることがわかる。そのため、事前学習済みモデルの出力にそのまま近傍分布を統合する場合は入念なハイパーパラメータの探索が大変重要である。

B.3　実験結果

t5-base で zero-shot と few-shot を行った時の性能を表 5 に示した。t5-small と同様に、Dialog では少量の事例を学習することで性能が大きく改善するが、XSum と DART では近傍分布の有効性は見られない。ただ、XSum の性能は few-shot 学習すると近傍分布によって性能が低下する。近傍分布の統合方法に関する最適なハイパーパラメータも大きく変化し、特に用いる近傍事例の数 k が大幅に削減されたことから、少量の学習データで隠れベクトルの空間が大きく変化し、生成に役立つ近傍事例以外の事例が多く検索されるようになってしまったことが推測される。この挙動の不安定さについては更なる調査が必要である。

表 4　各ハイパーパラメータの値を固定した時の fine-tuning した時（full）と zero-shot の時（zero）の BLEU 値の分散の比率 $\mathrm{Var}(\mathrm{BLEU}_{\mathrm{zero}})/\mathrm{Var}(\mathrm{BLEU}_{\mathrm{full}})$

		XSum	Dialog	DART
k	32	0.2201	24.8678	146.9497
	128	0.2598	19.1575	287.9740
	256	0.1983	14.0274	358.2954
	512	0.1869	10.6157	337.1579
T	10	0.0227	4.9278	18.3659
	50	0.2229	144.5953	155.8636
	100	1.4857	387.9873	209.9756
	200	0.9654	241.9247	318.0549
λ	0.2	0.0730	0.1111	96.5000
	0.4	0.1921	0.0082	29.8099
	0.6	0.1003	1.5240	5.0379
	0.8	0.5000	2.4268	1.6748

表 5　t5-base の実験。B は BLEU 値、R は ROUGE 値を示す。

タスク	自動要約		対話生成		Data-to-text	
評価指標	R-2	R-L	B	R	B	R
zero-shot	0.044	0.154	1.31	0.081	1.91	0.102
+ kNN	0.076	0.206	22.70	0.410	18.96	0.317
(λ = 1.0)	0.069	0.193	22.19	0.407	19.37	0.308
(k, T, λ)	(256, 200, 0.8)		(32, 200, 0.8)		(32, 100, 0.8)	
few-shot	0.044	0.154	17.10	0.289	1.92	0.103
+ kNN	0.064	0.192	25.42	0.444	19.05	0.309
(λ = 1.0)	0.053	0.173	25.00	0.439	18.07	0.296
(k, T, λ)	(32, 200, 0.6)		(32, 200, 0.8)		(32, 50, 0.8)	