モデル間転移へ向けた 複数 LLM の近傍構造に基づくハードプロンプトチューニング

岸本裕　小尾賢生　小杉哲　船越孝太郎　奥村学
東京科学大学
{ky,smalltail}@lr.first.iir.isct.ac.jp
{kosugi,funakoshi,oku}@first.iir.isct.ac.jp

概要

ハードプロンプトチューニングは，事前学習済み言語モデルのパラメータを固定したまま，入力に付与する文字列形式のプロンプトを最適化して下流タスクへ適応させる手法である．近年，ソースモデル上で最適化したプロンプトをターゲットモデルに適用するハードプロンプト転移が試みられているが，ターゲットモデル上で直接プロンプトチューニングを行った場合の性能には依然として及ばず，転移性能には改善の余地が残されている．本研究ではこの課題に対し，ハードプロンプトチューニングで選択される語彙を予め複数のソースモデル間で埋め込み空間における近傍構造が類似する対象に制限することを提案する．実験の結果，提案手法で最適化したプロンプトは転移性能を向上させるのみならず，プロンプトチューニングにおいても性能向上をもたらす傾向が確認された．

1　はじめに

現在，ファインチューニングを用いた大規模言語モデルの下流タスクへの適応が広く行われており，モデル規模の増加に伴う計算資源の増大は喫緊の課題である．これに対し，モデル本体のパラメータを固定したまま学習可能なトークン列を調整するハードプロンプトチューニング [1, 2, 3, 4, 5] がパラメータ効率の高い適応手法の 1 つとして注目されている．しかしながら，モデルの規模や種類が増大する昨今，モデルごとに個別にプロンプトチューニングを実施すること自体が，新たな計算コストの要因となりつつある．この問題に対処するため，ソースモデル上で最適化したプロンプトをターゲットモデルに適用するプロンプト転移 [5, 6, 7] が提案されている．しかしながら，既存のハードプロンプトチューニング手法で得られたプロンプトは転移後の性能がターゲットモデル上で直接プロンプトチューニングを行った場合の性能に及ばないことが多く，依然として改善に値する課題である．

本研究では図 1 に示すようにハードプロンプトチューニングの際，予め複数モデルで構造が類似した語彙を抽出する処理を提案する．これにより特定のモデルのみに有効な語彙をプロンプトの候補から排除し，ハードプロンプト転移性能の向上を図る．

2　関連研究

2.1　ハードプロンプトチューニング

ハードプロンプトチューニングは，入力文の前後に付与する文字列形式のプロンプトを最適化し，タスク性能を向上させる手法である．モデルがもつ語彙を $\mathcal{V}$，プロンプトトークン長を m とすると，ハードプロンプトは $T \in \mathcal{V}^m$ で表される．データセットを $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ とするとき，ハードプロンプトチューニングの目的は次式で定義される T^* を求めることである．

$$T^* = \underset{T \in \mathcal{V}^m}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(y_i \mid [T; x_i]) \tag{1}$$

離散トークン列 T そのものに対して目的関数の勾配を直接適用することは通常困難であるため，勾配情報に基づく置換で探索する AutoPrompt [1]，報酬に基づき離散トークン列を生成する方策を強化学習で最適化する RLPrompt [4]，連続埋め込みを更新しつつ各ステップで最近傍の語彙埋め込みへ射影して離散化する PEZ [5] などが提案されている．

2.2　ハードプロンプト転移

ハードプロンプト転移は，ソースモデル上で最適化したハードプロンプトをターゲットモデルの

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

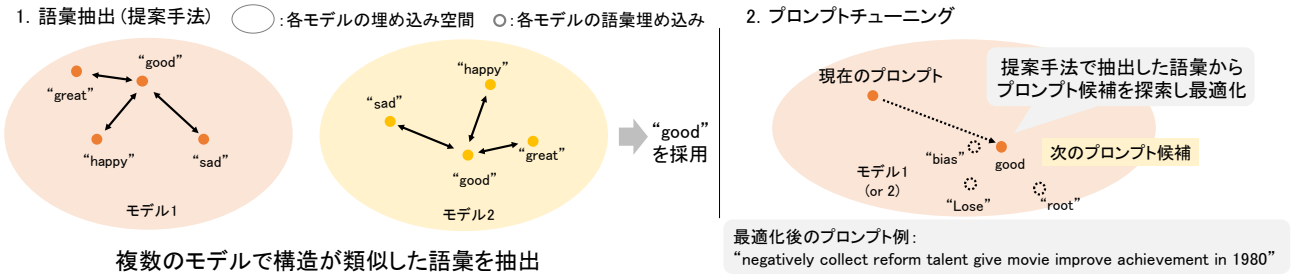

図 1: 提案手法のイメージ

入力として再利用する手法である．Rakotonirina ら [7] は，AutoPrompt [1] で最適化したハードプロンプトがモデルを跨ぐと性能低下しうることを示し，候補生成を担う generator model と候補評価を担う evaluator model を分離する mixed-training により，モデル間の一般化を改善する手法を提案した．

2.3 ソフトプロンプト転移

ソフトプロンプトは，モデルの埋め込み層に直接挿入された連続ベクトル形式のプロンプトを指す．ソフトプロンプト転移は，ソースモデルで最適化したソフトプロンプトをターゲットモデルの埋め込み空間へ写像し再利用する手法である．一般に，埋め込み空間の次元数や表現はモデル毎に異なるため，ソフトプロンプトの直接的な転移は困難である．この問題に対し Wu ら [6] は，ソース，ターゲットモデル間の共通語彙に対応する埋め込みと各モデルのプロンプトの相対的な幾何構造は概ね保存されるという仮説に基づき，ソースモデルにおけるソフトプロンプトと共通語彙埋め込みのコサイン類似度行列をターゲットモデルにおけるターゲットプロンプトと共通語彙埋め込みで模倣する転移手法を提案した．

3 提案手法

本研究では，Wu ら [6] の手法に着想を得て，複数モデル間で埋め込み空間の近傍構造が安定な語彙のみを用いてハードプロンプトを最適化する．

3.1 共通語彙埋め込みの抽出

語彙を抽出する N 個のモデルの内 $i(=1,\dots,N)$ 番目のモデルがもつ語彙を $\{\mathcal{V}_i\}$ とし，全てに共通する語彙を

$$\mathcal{V}_{\text{common}} = \bigcap_{i=1}^{N} \mathcal{V}_i \tag{2}$$

と定義する．ここで，共通語彙は文字列が一致し，かつ 1 トークンのものに限定する．$|\mathcal{V}_{\text{common}}| = c$ とし，モデルの埋め込み次元を d_i とする．共通語彙の埋め込みベクトルを $w_{iu} \in \mathbb{R}^{d_i}$ $(u = 1,\dots,c)$ とし，埋め込み行列 $W_i \in \mathbb{R}^{c\times d_i}$ の u 行目が w_{iu} となるように並べる．各行を ℓ_2 正規化した行列を $\tilde{W}_i$ とする．

$$\tilde{W}_i = \begin{bmatrix} \tilde{w}_{i1} & \cdots & \tilde{w}_{ic} \end{bmatrix}, \quad \tilde{w}_{iu} = \frac{w_{iu}}{\|w_{iu}\|_2}, \tag{3}$$

3.2 近傍構造一致度の算出

次に，$R_i = \tilde{W}_i \tilde{W}_i^{\top} \in \mathbb{R}^{c\times c}$ によりモデル次元に依存しない行列を獲得し，これを全モデルペアに渡って比較し構造が類似した語彙を抽出する．しかしながら本研究では計算資源的な制約により，一部の共通語彙をランダムにサンプリングした近似を行う．サンプリング時のばらつきを緩和するため反復回数を L とし，各反復 ℓ ごとに，$\{1,\dots,c\}$ からサイズ h の部分集合 $\mathcal{P}^{(\ell)}$ を一様ランダムにサンプルする．$\mathcal{P}^{(\ell)}$ に対応する行のみを抜き出した行列を $\tilde{W}_i^{(\ell)} \in \mathbb{R}^{h\times d_i}$ とし，

$$R_i^{(\ell)} = \tilde{W}_i \big(\tilde{W}_i^{(\ell)}\big)^{\top} \in \mathbb{R}^{c\times h} \tag{4}$$

と定義する．$R_i^{(\ell)}$ の第 u 行 $r_{iu}^{(\ell)} \in \mathbb{R}^h$ は，語 u と共通語彙 $\mathcal{P}^{(\ell)}$ に含まれる語とのコサイン類似度を並べたベクトルであり，これを語 u の相対的な近傍構造の近似とみなす．

次に，各モデル i に対して，$r_{iu}^{(\ell)}$ の上位 K 個の共通語彙集合 $\mathcal{N}_{iu}^{(\ell)} \subseteq \mathcal{P}^{(\ell)}$ を求める．ここで，$K \le h$ である．語 u について，モデル i と $j(j = 1,\dots,N \quad i \neq j)$ の近傍集合の一致度を Jaccard 係数 [8] で定義する．

$$J_u^{(\ell)}(i,j) = \frac{|\mathcal{N}_{iu}^{(\ell)} \cap \mathcal{N}_{ju}^{(\ell)}|}{|\mathcal{N}_{iu}^{(\ell)} \cup \mathcal{N}_{ju}^{(\ell)}|} \tag{5}$$

これを全てのモデル組に対して平均し

$$J_u^{(\ell)} = \frac{2}{N(N-1)} \sum_{1\le i<j\le N} J_u^{(\ell)}(i,j) \tag{6}$$

を得る．L 回の試行の平均

$$J_u = \frac{1}{L} \sum_{\ell=1}^{L} J_u^{(\ell)} \tag{7}$$

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

表 1: データセット：SST-2

Source Model: GPT-2-Large						
Method	Counterpart	Original	Transfer			
			GPT-2-XL	OPT-2.7B	LLaMA-3.2-3B	LLaMA-3.2-3B-Instruct
Empty	-	75.92	72.94	73.39	70.87	73.28
Random	-	$64.47_{\pm 4.57}$	$67.39_{\pm 3.99}$	$68.44_{\pm 2.95}$	$62.43_{\pm 2.40}$	$63.05_{\pm 5.28}$
Baseline (PEZ)	-	$81.54_{\pm 1.07}$	$71.83_{\pm 6.12}$	$71.51_{\pm 3.91}$	$64.13_{\pm 3.65}$	$66.44_{\pm 4.30}$
$\text{Baseline}_{\text{pruned}}$	-	$76.47_{\pm 5.96}$	$71.77_{\pm 7.31}$	$68.65_{\pm 4.13}$	$62.87_{\pm 5.44}$	$63.28_{\pm 3.78}$
$\text{CommonVocab}_{\text{pruned}}$	LLaMA-3.2-1B-Instruct	$80.30_{\pm 1.50}$	$72.20_{\pm 3.70}$	$72.13_{\pm 4.70}$	$66.58_{\pm 3.12}$	$68.07_{\pm 7.05}$
$\text{CommonVocab}_{\text{pruned}}$	OPT-1.3B	$77.16_{\pm 8.77}$	$70.11_{\pm 6.76}$	$71.88_{\pm 3.92}$	$65.67_{\pm 1.82}$	$66.88_{\pm 3.98}$
Ours	LLaMA-3.2-1B-Instruct	$\mathbf{82.41_{\pm 2.24}}$	$\mathbf{74.93_{\pm 4.03}}$	$\mathbf{73.69_{\pm 3.18}}$	$61.47_{\pm 4.55}$	$\mathbf{67.68_{\pm 4.59}}$
Ours	OPT-1.3B	$\mathbf{82.34_{\pm 2.51}}$	$\mathbf{73.88_{\pm 9.14}}$	$\mathbf{73.60_{\pm 8.65}}$	$\mathbf{68.35_{\pm 3.43}}$	$\mathbf{69.70_{\pm 5.55}}$

表 2: データセット：AGNEWS

Source Model: OPT-1.3B						
Method	Counterpart	Original	Transfer			
			GPT-2-XL	OPT-2.7B	LLaMA-3.2-3B	LLaMA-3.2-3B-Instruct
Empty	-	32.71	52.42	52.18	43.22	41.13
Random	-	$36.14_{\pm 2.44}$	$57.26_{\pm 4.37}$	$55.76_{\pm 2.76}$	$42.55_{\pm 1.26}$	$40.74_{\pm 0.70}$
Baseline (PEZ)	-	$53.33_{\pm 3.20}$	$56.61_{\pm 6.40}$	$49.79_{\pm 9.80}$	$42.71_{\pm 0.68}$	$41.02_{\pm 0.86}$
$\text{Baseline}_{\text{pruned}}$	-	$49.85_{\pm 2.18}$	$59.98_{\pm 4.27}$	$57.06_{\pm 4.59}$	$43.07_{\pm 1.88}$	$40.96_{\pm 0.46}$
$\text{CommonVocab}_{\text{pruned}}$	GPT-2-Large	$52.39_{\pm 2.02}$	$59.77_{\pm 5.15}$	$56.89_{\pm 5.80}$	$42.78_{\pm 1.96}$	$41.69_{\pm 1.35}$
$\text{CommonVocab}_{\text{pruned}}$	LLaMA-3.2-1B-Instruct	$52.49_{\pm 1.51}$	$56.87_{\pm 5.09}$	$48.60_{\pm 9.07}$	$41.81_{\pm 2.20}$	$41.82_{\pm 0.88}$
Ours	GPT-2-Large	$\mathbf{55.29_{\pm 1.55}}$	$\mathbf{63.28_{\pm 3.75}}$	$\mathbf{59.52_{\pm 5.03}}$	$42.68_{\pm 1.46}$	$\mathbf{43.47_{\pm 1.08}}$
Ours	LLaMA-3.2-1B-Instruct	$\mathbf{53.71_{\pm 1.76}}$	$\mathbf{60.89_{\pm 2.56}}$	$\mathbf{55.95_{\pm 5.09}}$	$41.92_{\pm 1.45}$	$\mathbf{42.38_{\pm 0.80}}$

を最終スコアとし，J_u が大きい順に上位 s 語彙からなる集合 $\mathcal{V}_{\text{sub}} = \text{Top-k}\left(\{J_u\}_{u=1}^{c}, s\right)$ を選択し，ハードプロンプトチューニングにおける探索空間を $\mathcal{V}_{\text{sub}}$ に制限する．

4 実験設定

ソースモデルには GPT-2-Large [9]，OPT-1.3B [10]，LLaMA-3.2-1B-Instruct [11] を用いた．ターゲットモデルはソースモデルよりも大規模なモデルを対象とし，GPT-2-XL [9]，OPT-2.7B [10]，LLaMA-3.2-3B，LLaMA-3.2-3B-Instruct [11] を用いた．データセットは 2 値感情分類の SST-2 [12, 13]，4 値カテゴリ分類の AGNEWS [14] を用いた．ハードプロンプトチューニング手法には AutoPrompt 等と比較して良好な性能が報告されている PEZ を用いた．[1] その他具体的なパラメータなどの情報は付録 A に示す．

5 実験結果

各タスクについて複数のソースモデルで実験を行った．紙面の都合上，本文では各タスクにおいて相対的に良好な結果を示したソースモデルの設定を抜粋して示す．本文で扱わない結果に関しては一部を付録 B に示す．本項では 5 つの異なるシード値で試行して得た精度の平均値と標準偏差を報告する．

次に表の用語について解説する．Original はソースモデルにおけるプロンプトチューニング自体の精度，Transfer はソースモデルで最適化したプロンプトを他モデルにそのまま適用したときの精度を示す．Empty は解答を誘導するテンプレートのみを入力に付したときの精度，Random はソースモデルの語彙をランダムに抽出しプロンプトとした際の精度を指す．また，Baseline は PEZ によるプロンプトチューニング，$\text{Baseline}_{\text{pruned}}$ は提案手法と同数語彙をランダムに抽出して PEZ を行った手法を指す．$\text{CommonVocab}_{\text{pruned}}$ は複数のソースモデルの共通語彙から提案手法と同数ランダムに抽出し PEZ を行った手法を指す．Ours は提案手法により構造が類似した共通語彙を抽出し PEZ を行った手法を指す．共通語彙を用いる手法は，ソースモデルと Counterpart に記されているモデルの 2 つで語彙抽出を行った．

表中の太字は Baseline と比較して優位な精度を示した Ours の数値を表す．

5.1 SST-2 の結果

GPT-2-Large をソースモデルとした転移結果を，表 1 に示す．Ours はいずれの Counterpart で語彙を抽出した場合においても Original の精度で Baseline

1） mixed-training は PEZ と最適化手続きが異なるため比較対象から除外した．

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

表 3: SST-2 における最適化後プロンプト

Source Model: GPT-2-Large		
Method	Counterpart	Prompt
Baseline (PEZ)	-	**seed1:** Internet hostedox LucMor affiliated hosted Hers appliances Facebook **seed2:** uf Garcia Kong Tor Hamp BaldwinondeMoore anime Singapore **seed3:** manga Mas SundaysolThis VillaberJapanese Crystalimm **seed4:** ress anime mindim accumulationHe Ins Breakfast office Hollywood **seed5:** ODSamsung {*unknown*}ort<pad> men Tend english brands Rand
Ours	OPT-1.3B	**seed1:** negatively collect reform talent give movie improve achievement in 1980 **seed2:** in crazy conservative independent SD criticizing job issues collaborative Dubai **seed3:** negatively caused create spiritual spirituality energy expectation enjoyment arrival artwork **seed4:** 2009 this mediocre 396 music utilize Music analyze Best competition **seed5:** create emotional fearful boring dull boring characters this an true

を上回った．転移について，Ours は Counterpart が OPT-1.3B の場合，4 つのターゲットモデルすべてで Baseline を上回った．Counterpart が LLaMA-3.2-1B-Instruct の場合，LLaMA-3.2-3B への転移の効用は限定的であったものの，他のターゲットモデルでは良好な結果が見られた．

5.2 AGNEWS の結果

AGNEWS において OPT-1.3B をソースモデルとしたプロンプトチューニングおよび転移結果を表 2 に示す．Ours は Original において，SST-2 と同様どちらの Counterpart においても Baseline の精度を上回った．また，$\text{Baseline}_{\text{pruned}}$ はターゲットモデルによっては Baseline を上回る場合もあり，本タスクにおいては探索空間の機械的な縮小が機能し得ることを示す．転移に関しては，GPT-2-XL，OPT-2.7B，LLaMA-3.2-3B-Instruct への転移において良好な性能を示す一方で，LLaMA-3.2-3B への転移では SST-2 と同様いずれの手法も Empty を下回った．

5.3 結果の解釈

Original の結果より，提案手法は PEZ プロンプトチューニング自体の性能を引き上げることが示唆される．

転移についても提案手法が良好な結果を示したが，Counterpart やターゲットモデルとの組み合わせに依存する傾向が見られた．これは，ソースモデルと Counterpart 間で整合的な語彙を選別できても，ターゲットモデル側の表現空間や語彙の使われ方が大きく異なる場合には，その語彙集合が必ずしも有効に機能しないためだと解釈できる．特に LLaMA-3.2-3B への転移で Empty を上回れない設定があることは，モデルアーキテクチャの差が大きい場合，プロンプト転移のみでは限界が残ることを示唆する．

6 プロンプトの比較

Ours，Baseline で作成されたプロンプトを比較する．Rakotonirina ら [7] は，実在する英単語がプロンプトに含まれる比率が高いほど，モデル間で汎化しやすい傾向があると報告している．そこで本項では，最適化されたプロンプトが実在する英単語として解釈可能である程度に着目し，その傾向を検証する．

表 3 に最適化後のプロンプトを示す．表では 5 つのシード値で試行して最適化されたプロンプトの結果をそれぞれ記述しており，文字化けが発生した場合は例外的に {*unknown*} と記述する．

Baseline で最適化したプロンプトは，文字化けするような語や"<pad>"のような特殊トークンを表す語が混入しているほか，"uf"や"LucMor"など一般的でない語が含まれる傾向がある．一方，Ours で最適化したプロンプトは単語単位で解釈できる語が殆ど全てを占め，転移結果と併せて考えると Rakotonirina ら [7] の主張と迎合する．

従って，本手法で最適化されるプロンプトは複数のモデルで汎化したプロンプトであることが示唆され，それが転移への有効性を示す一因の可能性があるが，詳細な検証は今後の課題である．

7 おわりに

本研究では，複数モデル間で埋め込み空間の近傍構造が類似する語彙に探索空間を制限したハードプロンプトチューニングが，モデル間転移性能に及ぼす影響を評価した．実験の結果，特定のモデル組み合わせでは転移後性能の改善が確認され，さらにプロンプトチューニング自体にも有効である可能性が示唆された．

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

8 謝辞

本研究は，東京科学大学のスーパーコンピュータ TSUBAME4.0 を利用して実施した．

参考文献

[1] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics.

[2] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[3] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: Learning vs. learning to recall. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 5017–5033, Online, June 2021. Association for Computational Linguistics.

[4] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 3369–3391, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[5] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2023.

[6] Zijun Wu, Yongkang Wu, and Lili Mou. Zero-shot continuous prompt transfer: Generalizing task semantics across language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[7] Nathanaël Carraz Rakotonirina, Roberto Dessì, Fabio Petroni, Sebastian Riedel, and Marco Baroni. Can discrete information extraction prompts generalize across language models? In **ICLR**, 2023.

[8] Johannes Hellrich, Bernd Kampe, and Udo Hahn. The influence of down-sampling strategies on SVD word embedding stability. In Anna Rogers, Aleksandr Drozd, Anna Rumshisky, and Yoav Goldberg, editors, **Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP**, pp. 18–26, Minneapolis, USA, June 2019. Association for Computational Linguistics.

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

[10] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

[11] Aaron Grattafiori, et al. The llama 3 herd of models, 2024.

[12] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing**, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[13] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. 2019. In the Proceedings of ICLR.

[14] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In **Advances in Neural Information Processing Systems**, 2015.

[15] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In **Proceedings of the 35th International Conference on Machine Learning**, Vol. 80 of **Proceedings of Machine Learning Research**, pp. 4596–4604. PMLR, 10–15 Jul 2018.

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

表 4: データセット：SST-2

Source Model: LLaMA-3.2-1B-Instruct						
Method	Counterpart	Original	Transfer			
			GPT-2-XL	OPT-2.7B	LLaMA-3.2-3B	LLaMA-3.2-3B-Instruct
Empty	-	82.11	72.94	73.39	70.87	73.28
Random	-	$75.48_{\pm 2.85}$	$62.27_{\pm 7.60}$	$61.90_{\pm 4.21}$	$63.51_{\pm 3.75}$	$63.26_{\pm 3.65}$
Baseline (PEZ)	-	$80.23_{\pm 1.68}$	$68.51_{\pm 9.52}$	$68.03_{\pm 5.96}$	$68.07_{\pm 6.31}$	$74.08_{\pm 2.37}$
$\text{Baseline}_{\text{pruned}}$	-	$80.34_{\pm 2.19}$	$63.58_{\pm 7.56}$	$67.57_{\pm 4.93}$	$68.30_{\pm 6.24}$	$71.01_{\pm 2.97}$
Ours	GPT-2-Large	$80.16_{\pm 2.00}$	**$72.89_{\pm 5.07}$**	**$72.06_{\pm 2.15}$**	$67.55_{\pm 3.19}$	$73.51_{\pm 3.54}$
Ours	OPT-1.3B	**$82.68_{\pm 3.29}$**	**$70.30_{\pm 7.43}$**	**$70.71_{\pm 6.92}$**	**$70.05_{\pm 4.96}$**	$72.84_{\pm 3.16}$

表 5: データセット：AGNEWS

Source Model: LLaMA-3.2-1B-Instruct						
Method	Counterpart	Original	Transfer			
			GPT-2-XL	OPT-2.7B	LLaMA-3.2-3B	LLaMA-3.2-3B-Instruct
Empty	-	41.13	52.42	52.18	43.22	41.13
Random	-	$26.34_{\pm 0.37}$	$55.03_{\pm 5.36}$	$49.47_{\pm 5.73}$	$43.24_{\pm 1.21}$	$40.61_{\pm 0.41}$
Baseline (PEZ)	-	$29.40_{\pm 1.01}$	$56.67_{\pm 3.22}$	$49.13_{\pm 1.94}$	$43.86_{\pm 2.25}$	$42.50_{\pm 2.15}$
$\text{Baseline}_{\text{pruned}}$	-	$28.93_{\pm 0.94}$	$53.73_{\pm 2.02}$	$45.51_{\pm 5.85}$	$41.93_{\pm 0.42}$	$41.57_{\pm 0.56}$
Ours	GPT-2-Large	$29.24_{\pm 0.34}$	**$60.63_{\pm 4.24}$**	**$55.62_{\pm 5.21}$**	$41.42_{\pm 0.87}$	**$42.69_{\pm 0.97}$**
Ours	OPT-1.3B	$29.10_{\pm 0.27}$	**$57.44_{\pm 6.32}$**	**$53.08_{\pm 4.87}$**	$41.86_{\pm 2.63}$	**$42.84_{\pm 1.35}$**

表 6: プロンプトチューニングのパラメータ

Parameter	Value
Learning rate	0.3
Training steps	5000
Train batch size	16
Optimizer	Adafactor[15]
The number of soft token	10

表 7: 提案手法のパラメータ

Parameter	Value
Vocabrary to be pruned m	8192
kNN value K	64
Number of iterations L	5
Top-k value s	8192

A 実験パラメータ

表 6 にプロンプトチューニングのパラメータを示す．また，表 7 に提案手法のパラメータを示す．今回は時間的な制約により提案手法のパラメータは固定したため，未だ改善の余地がある．

B LLaMA-3.2-1B-Instruct の結果

ソースモデルを LLaMA-3.2-1B-Instruct とした際の結果を表 4，表 5 に示す．

SST-2 について述べる．Original に関して，OPT-1.3B が Counterpart の際は提案手法が優位だが，Empty に対して改善量が微量であった．転移に関して，提案手法は GPT-2-XL，OPT-2.7B への転移は Baseline に比べ良好な精度を示したが，LLaMA 系への転移は効果が限定的であった．また提案手法を含めた殆どの手法が Empty を下回っており，Original の結果と併せて LLaMA 系に対してはプロンプトの挿入自体が有効とは限らないことが示唆される．

AGNEWS については，Original に関して全ての手法が Empty を下回った．従って SST-2 と同様にプロンプトチューニング自体が有効に機能していない可能性がある．転移では，どちらの Coutnerpart も LLaMA-3.2-3B 以外のモデルへの転移は提案手法が Baseline を大きく上回った．

以上より，手法に関わらず LLaMA 系に関してはプロンプト自体が有効でない可能性がある．しかしながら，転移に関してはターゲットモデルと同系統の Counterpart で語彙抽出をした際に転移性能が向上する傾向が示唆された．

C 開示事項

本研究の着想検討および構成の整理にあたり，対話型生成 AI（ChatGPT）を補助的に利用した．

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).