単一事例エキスパートの統合によるドメイン適応

清野 舜[◇,♠]　小林 颯介[♠,♡]　鈴木 潤[♠,◇]　乾 健太郎[♠,◇]
[◇]理化学研究所　[♠]東北大学　[♡]株式会社 Preferred Networks
shun.kiyono@riken.jp, {sosk, jun.suzuki, inui}@ecei.tohoku.ac.jp

1 はじめに

事前訓練済みモデルのファインチューニングは，今や NLP の各タスクにおけるデファクトスタンダードとも呼べる方法論となった．代表例として，巨大な生コーパス上で訓練された言語モデル（例：BERT[1]）を用いた転移学習や，符号化復号化モデル（Encoder Decoder; EncDec）のドメイン適応 [2] などが挙げられる．本研究では，ファインチューニングを実行する状況下で，従来のファインチューニングからさらなる性能向上が可能な方法論を模索する．こういった技術は，CoNLL[3] や WMT[4] などの各種シェアードタスクにおいて，他チームとの性能差を生むための有力な技術となりうる．

性能を引き上げるための有力な既存手法としてアンサンブルが挙げられる．アンサンブルでは，複数の乱数シードでモデルを独立に訓練し，各モデルの予測を統合することで，予測性能の向上をねらう．訓練済みモデルが同じであったとしても，アンサンブルはある程度の効果を示すが，性能向上を最大化するためには，モデルの事前訓練過程から別シードでの訓練が必要になることが知られている [5]. 事前訓練には数百から数千 GPU 時間規模の計算が必要であり [1, 6]，これは手元の限られた計算機資源では実現不可能な場合がある．

本研究では，事前訓練のやり直しを必要としない新しい方法論として，***k*-近傍アダプター平均法**を提案する．提案手法では，訓練データの中にはテストデータの推論に役立つ事例が存在するはずである，という仮定のもと，1 事例ごとに独立にエキスパートネットワークを作る．推論時は，各入力に対して類似する訓練データを求め，それに対応するエキスパートを用いて予測をおこなう．

EncDec のドメイン適応を題材とした実験を通して，提案手法が同一の事前訓練済みモデルを用いたアンサンブルよりも高い性能を達成できることを示す．

2 ドメイン適応

本節では，ドメイン適応のタスク定義を述べた後，ドメイン適応の既存手法 2 つ（全体ファインチューニングとアダプター）を概説する．

2.1 タスク定義

本研究では，機械翻訳のような系列変換タスクにおけるドメイン適応を題材として用いる．いま，(1) 汎用巨大コーパス $\mathcal{D}_{\text{pretrain}}$ で事前訓練されたモデル と (2) 対象ドメインの訓練データ $\mathcal{D}_{\text{domain}}$ の 2 つが与えられた際に，対象ドメインでの汎化性能を最も高くすることが目的である．

入力系列 $\boldsymbol{X}$ に対する出力系列 $\boldsymbol{Y}$ の条件付き確率を EncDec でモデル化することを考えると，その誤差関数は以下のように表される．

$$\mathcal{L}(\mathcal{D};\Theta) = -\frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{D}} \log P(\boldsymbol{Y}|\boldsymbol{X};\Theta) \qquad (1)$$

ここで，$\mathcal{D} = \{(\boldsymbol{X}_n, \boldsymbol{Y}_n)\}_{n=1}^{|\mathcal{D}|}$ は訓練データを表す．また，$\Theta \in \mathbb{R}^H$ は訓練対象のモデルパラメータを列ベクトルとして表したものであり，H はモデルの総パラメータ数である．

事前訓練においては，汎用コーパス $\mathcal{D}_{\text{pretrain}}$ 上で以下の最適化問題を解く過程でパラメータ $\Theta' \in \mathbb{R}^H$ を得る．

$$\Theta' \leftarrow \arg\min_{\Theta} \mathcal{L}(\mathcal{D}_{\text{pretrain}};\Theta) \qquad (2)$$

事前訓練済みパラメータ Θ' と対象ドメインの訓練データ $\mathcal{D}_{\text{domain}}$ を用いて，対象ドメインでの汎化性能を最も高くすることが目的である．

2.2 全体ファインチューニング

全体ファインチューニング（Full Finetuning; `FullFT`）は，ドメイン適応における最も素朴な手法である．事前訓練済みパラメータ Θ' をモデルの初期値として設定し，対象ドメインの訓練データ

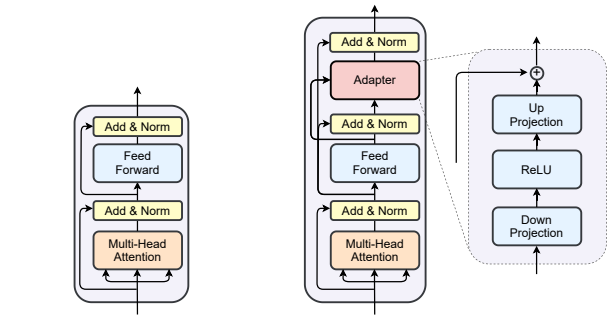

図 1　全体ファインチューニング（FullFT）とアダプターの比較：アダプターを用いる場合，新たに保存するパラメータはアダプターのみである

$\mathcal{D}_{\text{domain}}$ 上で訓練することで，パラメータ全体を更新する（図 1 左）．具体的には，以下の最適化問題を解く過程で，対象ドメインにより汎化したパラメータ $\Theta'' \in \mathbb{R}^H$ を獲得する．

$$\Theta'' \leftarrow \arg\min_{\Theta'} \mathcal{L}(\mathcal{D}_{\text{domain}}; \Theta') \tag{3}$$

FullFT は，素朴な手法でありながら高い性能が出せることから，ドメイン適応の研究におけるベースラインとして用いられる [2] ほか，各種シェアードタスク（例：WMT[7] や BEA[8]）でも常套手段として用いられる基盤技術である [9, 10]．

2.3　アダプター

アダプター [11] は，ファインチューニングを目的として事前訓練済みモデルの内部に追加される軽量なネットワークである（図 1 右）．以降の説明のため，アダプターのモデルパラメータの列ベクトル表現を $\Phi \in \mathbb{R}^O$ とする．ここで O はアダプターの総パラメータ数である．

アダプターのポイントは次の 2 点である：(1) ファインチューニングにあたっては，事前訓練済みモデルのパラメータ Θ' は固定し，アダプターのパラメータ Φ のみを更新する．(2) アダプターのパラメータ数 O は事前訓練済みモデルのパラメータ数 H よりも非常に小さく（$O \ll H$），通常 O は H の数%程度である [11]．FullFT ではモデル全体を保存しておく必要がある一方で，アダプターを用いる場合は新たに保存するパラメータはアダプターのみとなるため，保存に必要な記憶容量を削減できる．また，アダプターを取り除くことで事前訓練済みモデルの性能を復元可能であり，事前訓練で得た情報の破滅的忘却 [12] は原理的に生じない．

アダプターは，図 1 に示したように，非線形変換（例：ReLU）付きのフィードフォワード層から構成されることが多い．また，アダプターの大きさは中間層の次元数 d によって制御される．アダプターの追加箇所，Layer Normalization[13] やスキップ接続 [14] の位置などで様々な亜種が存在するが，本研究では Pfeiffer ら [15] の用いた構成に準ずる[1)]．

訓練時は以下の最適化問題を解くことによってアダプターを訓練し，訓練済みパラメータ $\Phi' \in \mathbb{R}^O$ を獲得する．

$$\Phi' \leftarrow \arg\min_{\Phi} \mathcal{L}(\mathcal{D}_{\text{domain}}; \Theta', \Phi) \tag{4}$$

推論時は Φ' と Θ' を用いて予測をおこなう．

アダプターを用いることで，BERT の転移学習 [11] や EncDec のドメイン適応 [2] などで，FullFT と同等の性能が達成できると報告されている．一方で，一部のデータセットにおいては，FullFT と比較して性能が数ポイント下回ってしまう場合もある [16, 2]．

3　提案手法

FullFT の性能を更に引き上げるための新しい方法論として，***k*-近傍アダプター平均法**（*k*-Nearest Neighbor Adapter Averaging; kAA）を提案する．

3.1　アイデアと概要

ドメイン適応において，各ドメインは更に細かいドメイン（サブドメイン）の集まりから構成されている．未知のデータが与えられた際，このサブドメイン単位に適応したモデルが手に入れば，より精緻な予測が可能になると考えられる．kAA の基本的なアイデアは，ドメインの粒度を事例単位にまで分解し，活用することである．

kAA の概要を図 2 に示す．kAA では，1 事例ごとに専用のエキスパートネットワーク（エキスパート）を訓練することで，各事例に適応したモデルを独立に作成する．この訓練は，FullFT 後のパラメータ Θ'' を用いて行なう．推論時は，入力データに応じて適切なエキスパート集合を抽出し，統合をおこなう．これにより，サブドメインに適応したモデルを作成し，FullFT からの性能向上をねらう．ここで，エキスパート集合の統合にあたっては，各エキスパートのパラメータを平均（アベレージング）す

1)　これは，著者らが実験によって見つけた経験的に最も良い構成であるとされる

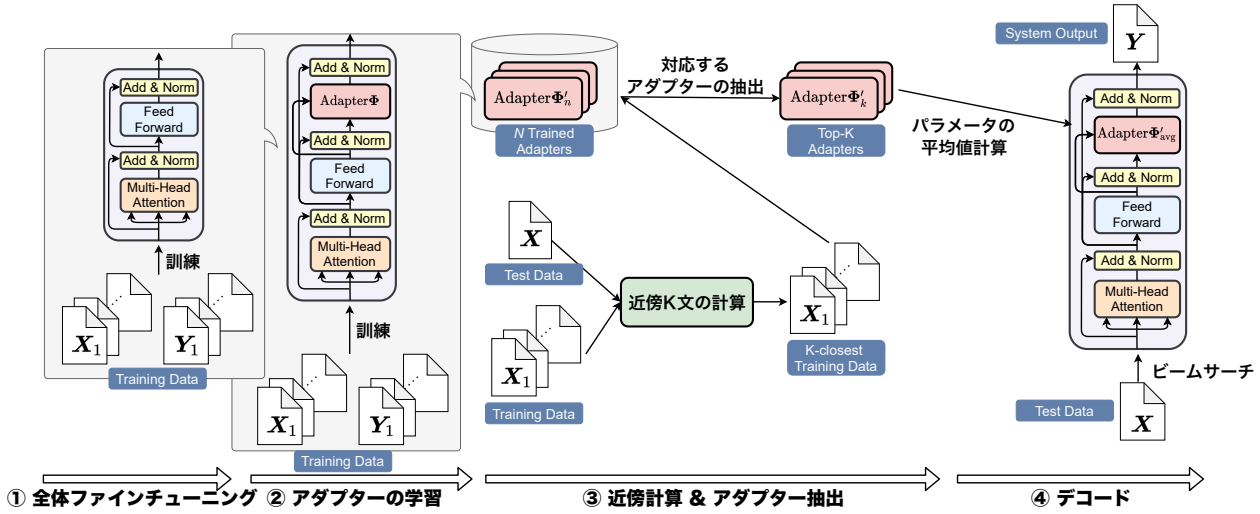

図 2　提案手法（k-近傍アダプター平均法）の概要

る．これにより，抽出したエキスパートの数によらず，推論時の計算時間は一定に保つことができる．

kAA の実現にあたって問題となりうるのが記憶容量である．通常，訓練済みモデルのファイルサイズは数百 MB から数 GB である．そのため，単純に各事例に対してモデルを保持しておくのは，記憶容量の観点から現実的ではない．そのため, アダプター（第 2.3 節）を用いて，必要な記憶容量を大幅に削減する．例えば，Transformer ベースの 6 層 EncDec[17] の各層にアダプターを追加する場合でも，$d = 16$ とすればファイルサイズは 1.6MB 程度となる．

3.2　アルゴリズム

訓練　いま，対象ドメインの訓練データ $\mathcal{D}_{\text{domain}}$ に N 個のデータが含まれるとして，$\mathcal{D}_{\text{domain}} = \{(\boldsymbol{X}_n, \boldsymbol{Y}_n)\}_{n=1}^{N}$ と表す．各データ $(\boldsymbol{X}_n, \boldsymbol{Y}_n)$ に対して，FullFT 済みのパラメータ Θ'' を初期値とした以下の最適化問題を解く過程で，合計 N 個のアダプター $\Phi'_1, \ldots, \Phi'_N$ を得る．

$$\Phi'_n \leftarrow \arg\min_{\Phi'} \mathcal{L}(\{(\boldsymbol{X}_n, \boldsymbol{Y}_n)\}; \Theta'', \Phi') \tag{5}$$

推論　学習したアダプターのうち $K (\ll N)$ 個を用いて予測をおこなう．具体的には，まず入力文 $\boldsymbol{X}$ と訓練データ $\mathcal{D}_{\text{domain}}$ の各文 $\boldsymbol{X}_n$ との類似度 $s_n \in \mathbb{R}$ を以下の通り計算する．

$$s_n = \text{sim}(f(\boldsymbol{X}), f(\boldsymbol{X}_n)) \tag{6}$$

ここで，sim と f はそれぞれ任意の類似度関数と，文をベクトル空間に埋め込む関数である．

次に，類似度上位 K 文に対応する K 個のアダプター $\Phi'_1, \ldots, \Phi'_K$ を入力文 $\boldsymbol{X}$ の推論に用いる．具体的には，各アダプターのパラメータの平均を求め，各アダプターに含まれる情報を統合する．

$$\Phi'_{\text{avg}} = \frac{1}{K} \sum_{k=1}^{K} \Phi'_k \tag{7}$$

最後に，Φ'_{avg} と Θ'' を用いて推論を行なう．

4　実験

4.1　実験設定

データセット　実験には，Aharoni ら [18] が提供している機械翻訳のドメイン適応用のベンチマークデータセットを用いた[2)3)]．このデータは英独のパラレルコーパスであり，5 個のドメイン（IT, Koran, Law, Medical, Subtitles）が含まれている．本研究では，各ドメインについて独 → 英の翻訳を行う．今回は，各ドメインのデータセットの量が限られている場合を想定し，各ドメインから訓練データとして 4000 文サンプルして用いた．各ドメインの開発・評価データはともに 2000 文である．モデルの評価には，sacreBLEU[20] で計算した BLEU 値を用いる．

モデル　EncDec の事前訓練済みモデル Θ' として，Ng ら [10] の公開している WMT2019[7] での優勝モデルを用いた[4)]．同モデルは，Transformer

2）https://github.com/roeeaharoni/unsupervised-domain-clusters
3）このデータは Koehn ら [19] の作成したデータセットから，訓練データと開発・評価データの重なりを除いたものである．詳細については Aharoni らの論文 [18] を参照されたい．
4）https://github.com/pytorch/fairseq/tree/master/examples/wmt19

表 1　各種手法の BLEU スコアとデコーダの計算回数：**太字**は同一カラム内での最高値を表す．† は提案手法を示す．

	IT		Koran		Law		Medical		Subtitles		Average		
	Valid	Test	Valid	Test	Valid	Test	Valid	Test	Valid	Test	Valid	Test	デコーダ計算回数
w/o finetuning	36.37	38.38	16.69	17.07	46.10	45.96	40.32	41.86	29.67	29.38	33.83	34.53	1 回
Adapter	38.70	40.17	19.87	20.15	48.43	48.22	43.98	44.44	**30.73**	30.37	36.34	36.67	1 回
FullFT	38.75	40.56	20.51	20.91	49.13	48.74	45.40	45.48	30.28	30.32	36.81	37.20	1 回
FullFT+Ens	38.92	**40.66**	**20.70**	21.04	49.28	48.81	45.54	45.61	30.44	30.32	36.98	37.29	M (モデル数) 回
FullFT+ModelAvg	**38.98**	40.58	**20.70**	20.99	49.27	48.76	**45.67**	45.67	30.41	30.37	**37.01**	37.27	1 回
FullFT+kAA†	38.90	40.60	20.67	**21.18**	**49.42**	**49.04**	45.61	**45.92**	30.32	**30.42**	36.98	**37.43**	1 回

(big)[17] をベースに数百万文対規模の対訳データで訓練されたものである．その他のハイパーパラメータに関しては付録 A を参照されたい．

比較手法　実験では，FullFT をベースラインとして用いて kAA の効果を検証する．kAA との直接の比較手法として，アンサンブル（Ensemble; Ens）とモデル平均法（Model Averaging; ModelAvg）を用いる．Ens では，単一の事前訓練済みモデルを用いて M 個の異なるシードで学習をおこない，推論時に各モデルの予測を平均する．また，ModelAvg は Ens から得られた M 個のモデルのパラメータの平均を用いて推論をおこなう手法である．ModelAvg との比較から，モデルパラメータの平均値を用いること自体による効果を調べる．また，FullFT が十分に強力なベースラインであることを示すために，事前訓練済みモデル単体（w/o finetuning）と，アダプターを用いてファインチューニングを行った場合（Adapter）の性能も報告する．

4.2　実験結果

表 1 に実験結果を示した．

ベースラインの性能確認　FullFT と Adapter は，どちらも w/o finetuning から性能が向上しているが，特に検証・テストセット両方の平均値において，FullFT のほうが Adapter より高い性能を示した．これは，EncDec にアダプターを用いた既存研究 [2, 16] と一致する結果である．この結果から，FullFT は適切なベースラインであるといえる．

スコアの引き上げ手法の比較　kAA，Ens と ModelAvg は全てベースラインである FullFT から性能が上がっている．特に，kAA が IT を除く全てのドメインのテストセット上で Ens や ModelAvg よりも高い性能を達成したことから，提案手法の有効性が示唆される．kAA と ModelAvg との比較から，性能向上はモデルパラメータの平均を用いることによるものではなく，kAA で訓練・抽出したアダプターの効果によるものだと言える．また，Ens の場合はモデル数 M 回のデコーダの計算が必要だが，kAA では 1 回で計算可能であることも大きなメリットである．

表 1 の右に示した各ドメインの平均値（Average）を比べると，kAA は，検証セット上の性能（36.98）では Ens（36.98）や ModelAvg（37.01）と同等であるが，テストセット上の性能ではこれら 2 つを上回る性能（37.43）を示した．これは，FullFT による学習過程における，検証セット上の性能を用いた早期終了が原因だと考えられる．具体的には，Ens と ModelAvg は M 回の FullFT のそれぞれで合計 M 回の早期終了を実行するため，意図せず検証セットに対して過学習している可能性がある．一方で，kAA では FullFT の早期終了が一度で良いため，過学習の影響が軽減されていると推察される．

同じ事前訓練済みモデルを用いたアンサンブル　Ens の性能が kAA に及ばないのは，Ens が同じ事前訓練済みモデルから複数個のモデルを作っていることも原因の一つだと考えられる．Ens による性能向上を最大化するためには，モデルの事前訓練過程をやり直す必要がある [5]．アンサンブルでは各モデルの持つ異なった局所解の情報を利用するが [21]，同じ訓練済みモデルを初期値として用いることで，それぞれの局所解が近傍に集まるため性能が伸びないと考えられる．

5　おわりに

本研究では，全体ファインチューニングの性能を更に引き上げることを目的とした新しい方法論（k-近傍アダプター平均法）を提案した．実験では，提案手法はアンサンブルよりも高い性能を達成できることを示した．この結果から，提案手法は単一の事前訓練済みモデルと少量（数千文規模）の訓練データを利用できる状況下で有効に動作することが示唆される．今後は入力データに対するアダプター選択の手法を改善し，さらなる性能向上を目指す．

参考文献

[1] 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 (NAACL 2019)*, pp. 4171–4186, 2019.

[2] Ankur Bapna and Orhan Firat. Simple, Scalable Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1538–1548, 2019.

[3] Raquel Fernández and Tal Linzen, editors. *Proceedings of the 24th Conference on Computational Natural Language Learning*, 2020.

[4] Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors. *Proceedings of the Fifth Conference on Machine Translation*, 2020.

[5] 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. *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020.

[6] Shar Narasimhan. NVIDIA Clocks World's Fastest BERT Training Time and Largest Transformer Based Model, Paving Path For Advanced Conversational AI, Aug 2019.

[7] Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pp. 1–61, 2019.

[8] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2019)*, pp. 52–75, 2019.

[9] Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. Tohoku-AIP-NTT at WMT 2020 News Translation Task. In *Proceedings of the Fifth Conference on Machine Translation (WMT 2020)*, pp. 145–155, 2020.

[10] Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. Facebook FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pp. 314–319, 2019.

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019.

[12] Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. Vol. 24 of *Psychology of Learning and Motivation*, pp. 109 – 165. Academic Press, 1989.

[13] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 770–778, 2016.

[15] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive Task Composition for Transfer Learning. *arXiv preprint arXiv:2005.00247*, 2020.

[16] Minh Quang Pham, Josep Maria Crego, François Yvon, and Jean Senellart. A Study of Residual Adapters for Multi-Domain Neural Machine Translation. In *Proceedings of the Fifth Conference on Machine Translation (WMT 2020)*, pp. 617–628, 2020.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems 31 (NIPS 2017)*, pp. 5998–6008, 2017.

[18] Roee Aharoni and Yoav Goldberg. Unsupervised Domain Clusters in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pp. 7747–7763, 2020.

[19] Philipp Koehn and Rebecca Knowles. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, 2017.

[20] Matt Post. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pp. 186–191, 2018.

[21] Lars Kai Hansen and Peter Salamon. Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 10, pp. 993–1001, Oct 1990.

[22] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[23] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, 2019.

A　ハイパーパラメータ

実験（第 4 節）で用いたハイパーパラメータの一覧を表 2 に示す.

表 2　ハイパーパラメータの一覧

設定名	値
最適化手法	Adam[22]
学習率	5e-05
ミニバッチサイズ	4,000 トークン
アンサンブル（Ens）のモデル個数 M	{5,10}のうち開発セットの性能から選択
モデル平均法（ModelAvg）のモデル個数	{5,10}のうち開発セットの性能から選択
kAA で用いるアダプターの個数 K	{5,10,20,30} のうち開発セットの性能から選択
kAA で用いるアダプターの次元数 d	16
訓練中のモデルパラメータのアベレージング [17]	最後 10 エポックのチェックポイントを使用
類似度関数 sim	cosine 類似度
埋め込み関数 f	EncDec のエンコーダの最終隠れ層の平均
Adapter で用いるアダプターの次元数 d	{16, 32, 64, 128, 256, 512}のうち開発セットの性能から選択
ビームサーチ	ビーム幅 5，長さペナルティ 1.0
フレームワーク	Fairseq[23]