Vol.2019-NL-241 No.1
2019/8/29

事前訓練済みBERTエンコーダーを再利用したニューラル機械翻訳

今村 賢治[1,a)]　隅田 英一郎[1]

概要：本稿では，事前訓練済みの BERT (Bidirectional Encoder Representations from Transformer) モデルを Transformer ベースのニューラル機械翻訳 (NMT) に適用する．単言語のタスクと異なり，NMT の場合，BERT のモデルパラメータ（訓練済み）に比べ，デコーダー（未学習）のパラメータ数が多い．そこで，まず BERT エンコーダーのパラメータを固定して，未学習パラメータのみを訓練し，その後，全体を微調整する 2 段階最適化を行う．実験では，直接微調整したときには BLEU スコアが極めて低くなったのに対して，2 段階最適化では訓練が成功した．その結果，Transformer の基本モデルや，モデル構造が同じ事前訓練なしの Transformer に比べても BLEU スコアが向上することが確認された．また，少資源設定で，より効果が高いことが確認された．

Neural Machine Translation that Recycles Pre-trained BERT Encoder

1. はじめに

Bidirectional Encoder Representations from Transformers, BERT [6] は，言語表現モデルである．大規模な単言語データで事前に訓練されたモデルであり，基本的にはそれを微調整 (fine-tuning) [8], [16] することで，所望のタスクに合わせる．BERT を用いたシステムは，The General Launguage Understanding Evaluation (GLUE) ベンチマーク [19] や，Stanford Question Answering Dataset (SQuAD) を用いた機械読解 (reading comprehension) ベンチマーク [12] など，さまざまなタスクで高い精度を達成している [6]．しかし，BERT が使われているタスクは，主に自然言語理解に関連するものであるため，基本的に単言語である．

一方，BERT の考え方を多言語に拡張したモデルも提案されている（cross-linugual language models; XLM と呼ばれている）[10]．このモデルは複数の言語で事前訓練されており，これをエンコーダー，デコーダーと見なして，機械翻訳を実現することもできる．

本稿では，事前訓練済みの BERT エンコーダーを，Transformer [18] ベースのニューラル機械翻訳 (NMT) に適用する．具体的には，NMT のエンコーダー部分を BERT に置き換える．BERT を用いたシステムや XLM に基づく教師あり機械翻訳は，タスクデータによる微調整で訓練される．しかし，本稿のシステムは，未学習であるデコーダーのパラメータ数が非常に多く，単純に微調整を行うと発散してしまうなど，安定した訓練が難しい．そこで本稿では，未学習パラメータの訓練と微調整の 2 段階で訓練する．

実験では，直接微調整したときには BLEU スコアが極めて低くなったのに対して，2 段階訓練では訓練が可能だった．その結果，Transformer の基本モデルや，モデル構造が同じ事前訓練なしの Transformer に比べても BLEU スコアが向上した．このように，他の目的（この場合，言語理解）に訓練されたネットワークでも，訓練を未学習部分の訓練，微調整の 2 段階にすると，別の目的（この場合，機械翻訳）に再利用することができる．その意味で，本稿は，ネットワーク再利用の一例となると考える．

以下，第 2 節では，BERT について簡単にレビューする．第 3 節では，提案方式である BERT 利用機械翻訳のモデルおよび訓練法を説明する．第 4 節では，実験を通じて提案方式の特徴を調査する．第 5 節では，関連研究や，逆翻訳とモデル再利用について議論し，第 6 節でまとめる．

[1] 国立研究開発法人　情報通信研究機構
National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
[a)] kenji.imamura@nict.go.jp

2. 関連研究

2.1 BERT

BERT [6] は，Bidirectional Encoder Representations from Transformers の略である．モデルの形式としては，Transformer ベースのエンコーダーで，単語列を入力とし，各単語に対応する分散表現 (representations) を出力する．内部では，マルチヘッドの自己注視機構により，入力の文脈を考慮した符号化が行われている．

BERT は，大規模コーパスから事前訓練された状態で配布されている．配布されているものは，12 層か 24 層と，Transformer の基本モデル（6 層）より深い．ユーザ（システム開発者）はモデルに，自分のタスクに合わせたネットワークを追加し，タスクデータで微調整することで，様々なシステムを構築する．たとえば文書分類タスクに使用する場合，BERT にクラス分類用の生成層（線形変換＋ソフトマックスから構成）を追加することで，分類器を構築する．同様に，固有表現抽出器を作る場合，各単語を分散表現から固有表現タグに変換する生成層を加え，微調整する．

事前訓練は，マスク言語モデルタスクと次文予測タスクの 2 種類で行われている．どちらも言語モデルとしての予測性能を上げるように訓練されている．

マスク言語モデルタスクでは，入力の一部を特殊単語 `[MASK]` または他の単語に置き換え，元単語を予測するように訓練する．たとえば，“`my dog is hairy`” という文に対しては，“`my dog is [MASK]`” という入力を与え，`[MASK]` 部分の単語が `hairy` であったことを予測する．予測には前方文脈と後方文脈の両方が使われる．

次文予測タスクでは，2 文を入力し，その 2 つが連続した 2 文かどうかを学習する．2 値分類タスクを Transformer で実現するため，入力の先頭に特殊単語 `[CLS]` を付与し，そこに対応する出力で分類する．また，文区切りは特殊単語 `[SEP]` で表す．

BERT は，The General Launguage Understanding Evaluation (GLUE) ベンチマーク [19] や，Stanford Question Answering Dataset (SQuAD) を用いた機械読解 (reading comprehension) ベンチマーク [12] など，さまざまなタスクで高い精度を達成している．しかし，これらは自然言語理解 (NLU) に関するものであるため．基本的に単言語である．

2.2 多言語モデル

BERT の考え方を多言語に拡張したモデル (cross-lingual language models; XLM) も提案されている [10]．XLM も形式的には Transformer モデルであるが，複数言語の単言語コーパスで訓練されている．また，対訳文を接続して入力することで，1 つの Transformer モデルの中で対訳関係も学習する．

XLM は，事前訓練された 2 つのモデルをエンコーダー，デコーダーとみなし，機械翻訳も実現することができる．本稿で述べる，BERT エンコーダーを使った NMT は，基本的にはエンコーダーのみ XLM を用いた機械翻訳と同じものである．ただし，我々の方式は，異なるシステムを接続して，効果を上げることを主眼においているため，モデル再利用 [13] を BERT エンコーダーと Transformer のデコーダーで行ったものともみなせる．

XLM による機械翻訳を含む，多くの事前訓練システムは，微調整のみでシステムを訓練する [6], [10] ．しかし，事前訓練モデルのパラメータに比べ，未訓練パラメータが多い場合，壊滅的忘却 (catestrophic forgetting)[9] と呼ばれる現象により事前訓練のパラメータが破壊され，その結果，訓練が発散する場合がある[*1]．この問題を抑制し，安定的に訓練を行う必要がある．

3. BERT 利用 NMT

本節では，BERT エンコーダーを使った NMT について説明する．

3.1 モデル

本稿の NMT は，基本的には Transformer モデルを用いたエンコーダー・デコーダーである．**図 1** に構成を示す．これに対し，BERT も Transformer エンコーダーの一種なので，我々はそのまま NMT のエンコーダーとして利用する．BERT によって符号化された各単語毎の分散表現 (encoder representations) は，Transformer デコーダーの文脈注視機構に入力され，翻訳文が生成される．なお，本稿では，BERT ではない従来のエンコーダーを Transformer エンコーダーと呼称する．また，Transformer デコーダーは 6 層に固定する．

BERT エンコーダーは，決められた前処理で事前訓練されているため，機械翻訳に利用する際も，BERT トークナイザーで単語分割する必要がある．このトークナイザーは WordPiece [20] に基づくサブワード化を含んでいる．入力は，このトークン列の先頭に特殊単語 `[CLS]`，最終に文区切り記号 `[SEP]` を付加したものになる．

今回，デコーダーには Transformer を使用するが，再帰ニューラルネットワークベースの NMT [1], [17] でも適用可能と考えている．

3.2 訓練

通常，事前訓練済みの BERT を使ってシステムを構築する際は，タスク特有のネットワークを追加して，タスク特

[*1] 初期値やハイパーパラメータの設定次第では，通常の訓練でも発散する場合があるが，われわれの実験では，BERT 利用 NMT でこの現象が顕著に発生した．

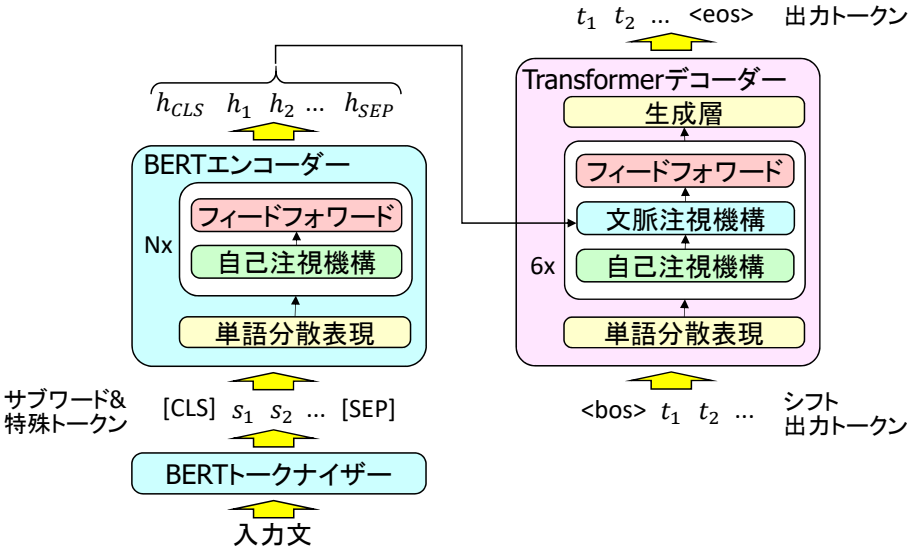

図 1　BERT エンコーダーを使った NMT の構成
Fig. 1　Structure of NMT Using BERT Encoder

有のデータで微調整するだけでよい．これは，追加したパラメータが十分に少なく，微調整で追加部分が学習できるためである．

本稿ではデコーダーが追加部分に相当するが，パラメータ数は非常に多い．実際，4 節の実験で使用するモデルのパラメータ数は，BERT エンコーダーが約 1.1 億個であるのに対し，Transformer デコーダーは約 8 千万個あり，微調整だけでは学習させることができない．そこで，本稿では，デコーダー訓練と微調整の 2 段階で学習する．

3.2.1　デコーダー訓練

この段階では，訓練データに対して，エンコーダーパラメータを固定 (freeze) し，デコーダーのパラメータのみを更新する．

具体的には，通常の翻訳器の訓練と同様に，対訳コーパスで訓練を行うが，誤差逆伝播はデコーダーのみにとどめ，エンコーダーへの伝播を行わない．このため，BERT エンコーダーは，逆伝播用の入出力の保存と，勾配計算が必要なくなり，モデルが巨大であっても比較的小さなメモリで訓練することができる．また，ドロップアウトは，Transformer デコーダーのみに適用し，BERT エンコーダーには適用しない．

なお，訓練の終了条件は，最終的には開発セットにおける損失が最低になったときであるが，微調整と組み合わせたときにどの程度デコーダーを訓練する必要あるのか，4.3 節で議論する．

3.2.2　微調整

微調整段階では，BERT エンコーダーのパラメータも通常どおり更新し，end-to-end の全体最適化を行う．

デコーダー訓練時に，開発セットの損失が最低になるまで学習済みだとしても，エンコーダーのパラメータも更新可能にした場合，デコーダーに合わせてエンコーダーが学習されるため，さらに最適化が行われる．なお，この段階では，ドロップアウトは BERT エンコーダー，Transformer デコーダー両方に適用する．

微調整段階では，すべての層の逆伝播計算を行うので，大きなメモリを消費する．

表 1　実験データのサイズ
Table 1　Sizes of Experimental Data

タスク	セット	文数	トークン数（英語）
WMT-2014 英独翻訳	訓練	4,468,840	1.16 億
	newstest2013	3,000	6.6 万
	newstest2014	2,737	6.3 万
	newstest2015	2,169	6.8 万
IWSLT-2015 英越翻訳	訓練	133,317	270 万
	tst2012	1,553	3.4 千
	tst2013	1,268	3.4 千

4.　実験

4.1　実験設定

4.1.1　データ

本稿では，Stanford NLP グループが前処理を行った，共有タスクデータ[*2]を使用した．データサイズを**表 1** に示す．

一つめのタスクは，Workshop on Statistical Machine Translation (WMT-2014) [2] の英独ニュース翻訳タスク (En-De) である．訓練データの他に，newstest2013 - 2015 ののべ 4 セットが公開されている．本稿では，newstest2013 を開発セットとし，newstest2014, 2015 をテストセットとして使用した．

[*2] https://nlp.stanford.edu/projects/nmt/

また，小規模データを対比させるため，International Workshop on Spoken Language Translation (IWSLT-2015)[3] の英語・ベトナム語翻訳タスク (En-Vi) も使用した．こちらは，tst2012 を開発セットとし，tst2013 をテストセットとした．

これらデータの目的言語は，バイトペア符号化 [15] を用いて，1.6 万のサブワードに分割した．原言語（英語）に関しては，Transformer エンコーダーを使う場合は 1.6 万サブワード，BERT エンコーダーを使う場合は，BERT トークナイザーに従い，3 万のサブワードに分割した．

4.1.2 BERT モデル

本稿では，GitHub で公開されている事前訓練済みモデル[*3]を使用した．今回使用したモデルは，BERT 基本モデルと呼ばれているもので，大文字小文字区別なし入力, 12 層，隠れ状態 768 ユニット，12 ヘッド，パラメータ数は約 1.1 億である．このモデルは，BooksCorpus [22] と英語 Wikipedia（合わせて約 33 億語）で訓練されている．

なお，このモデルは，TensorFlow ライブラリ用に訓練されたものなので，ツール[*4]を用いて PyTorch ライブラリ用に変換して使用した．

4.1.3 翻訳システム

ベースとなる翻訳システムは，fairseq [*5]を使用した．fairseq は，PyTorch 上に構築された NMT システムで，Transformer モデルを含む．このエンコーダー部分を BERT に置き換え，デコーダー部分は fairseq をそのまま使用した．

デコーダーモデルは，6 層 Transformer であるが，エンコーダー出力を文脈注視機構で混合するため，隠れ層のサイズやヘッド数は，エンコーダーと同じに設定した．

4.1.4 ハイパーパラメータ

表 2 に設定を示す．微調整時のハイパーパラメータは，学習率とウォームアップ以外はデコーダー訓練と同じ設定を使用した．

4.1.5 評価

評価は，BLEU [11] で行った．また，検定は，ブートストラップ再サンプリングに基づく MultEval ツール [5][*6] を使用し，危険率は $p < 0.05$ とした．

4.2 システム比較

本節では，WMT-2014 データでシステム比較を行う．

今回は，文献 [18] の Transformer 基本モデルをベースラインとし，それとの比較で提案法を評価する．また，参考として，エンコーダーのモデル構造を BERT モデルと同等にした場合 (Transformer BERT サイズモデル; つまり 12 層，隠れ状態 768 ユニット，12 ヘッド) も示す．これは，モデルの表現力による翻訳精度の影響を確認するためのものである．

表 2 ハイパーパラメータ設定
Table 2 Hyperparameter Settings

種別	オプション名	値
デコーダー訓練	バッチサイズ	約 500 文
	最適化方式	Adam
		$\beta_1 = 0.9, \beta_2 = 0.99$
	学習率	0.0004
	ウォームアップ	約 5 エポック
	学習率減衰	逆平方根
	ラベル平滑化	0.1
	ドロップアウト	0.15
	損失関数	ラベル平滑化クロスエントロピー
微調整	学習率とウォームアップを除き，デコーダー訓練と同じ	
テスト	ビームサイズ	10
	長さペナルティ	1.0 固定

提案法（BERT 利用 NMT と呼ぶ）は，まず開発セット (newstest2013) の損失が最低になるまで，デコーダー訓練を行い，その後微調整した．微調整の際の学習率は，デコーダー訓練時の 1/10 から 1 倍に変化させて，その影響をみた．なお，ウォームアップも学習率に合わせて 1/N 倍している．結果を**表 3** に示す．表中の開発 PPL は開発セット (newstest2013) のパープレキシティ，BLEU の年号は，それぞれ newstest2013, 2014, 2015 の BLEU スコアを表す．

まず，ベースライン同士を比較する．Transformer 基本モデルに比べ，Transformer BERT サイズモデルは，開発セットのパープレキシティが低下しており，良質なモデルが学習できている．しかし，BLEU スコアでは若干低下（有意差なし）しており，エンコーダーのパラメータ増加が，必ずしも翻訳品質に繋がっていないことがわかる．

次に，Transformer 基本モデルと BERT 利用 NMT を比較する．デコーダー訓練をまったく行わず，直接微調整した場合，訓練そのものは収束したが，BLEU スコアは著しく低下し，モデルが壊れていることが示唆されている．単言語タスクにおける BERT の利用方法とは異なり，BERT エンコーダーを利用する機械翻訳では，直接微調整を行うのは難しい．

一方，提案方式では，デコーダー訓練した直後（微調整前）は，ベースラインに比べ開発 PPL は高く，BLEU スコアは低い．BERT エンコーダーは，今回の訓練データとは異なるデータで訓練されているため，データの不整合から十分な学習ができいない．

しかし，微調整後は，開発 PPL は低くなり，それに伴い BLEU スコアは向上する．Transformer 基本モデルに比べると，BLEU スコアはほぼすべてのケースについて有意に向上している．Transformer BERT サイズモデルと比べて

[*3] https://github.com/google-research/bert
[*4] https://github.com/huggingface/pytorch-pretrained-BERT
[*5] https://github.com/pytorch/fairseq
[*6] https://github.com/jhclark/multeval

表 3　システム比較結果
Table 3　Results of System Comparison

太字は，全システムでの最良値を表す．(+)(-) は，Transformer 基本モデルに比べ，それぞれ有意に向上，有意に悪化（$p < 0.05$）を表す．

	システム	開発 PPL ↓	BLEU ↑ 2013	2014	2015	備考
ベースライン	Transformer 基本モデル	4.23	26.29	27.22	29.48	検定用ベースライン
	Transformer BERT サイズモデル	4.04	26.15	27.09	29.32	
BERT 利用 NMT	直接微調整（学習率=8e-5）	4.28	0.13 (-)	0.10 (-)	0.12 (-)	33 エポックで収束
	提案方式：デコーダー訓練直後	4.76	24.13 (-)	23.62 (-)	25.74 (-)	65 エポックで収束
	＋微調整（学習率=4e-5）	3.93	**27.14** (+)	28.27 (+)	30.68 (+)	21 エポックで収束
	＋微調整（学習率=8e-5）	**3.92**	27.05 (+)	**28.90** (+)	**30.89** (+)	9 エポックで収束
	＋微調整（学習率=1.2e-4）	3.93	27.03 (+)	28.50 (+)	30.51 (+)	11 エポックで収束
	＋微調整（学習率=1.6e-4）	3.94	26.64	28.59 (+)	30.51 (+)	11 エポックで収束
	＋微調整（学習率=2.0e-4）	3.95	26.89 (+)	28.67 (+)	30.24 (+)	12 エポックで収束
	＋微調整（学習率=4.0e-4）	学習が発散したため，測定せず				

も，開発 PPL は低下しているため，これはモデルパラメータが増えたためではなく，学習性能の差と言ってよい．

微調整時の学習率 (LR) を 4e-5 から 4.0e-4 まで変化させた場合，4.0e-4（デコーダー訓練時の学習率と同じ）では学習が発散してしまい，訓練できなかった．それ以外の学習率では，開発 PPL，BLEU スコアともに大きな差はないが，学習率 8e-5 の時，9 エポックで収束し，学習が最も早かった．以降の実験では，微調整の学習率は 8e-5 で固定する．

4.3　デコーダー訓練のエポック数

提案方式では，まずデコーダー訓練を収束するまで実施して，そののちに微調整する．しかし，この方法は，デコーダー訓練に時間がかかるため，効率はよくない．実際，前節の実験では，デコーダー訓練に 65 エポックかかった．本節では，デコーダー訓練を収束前に停止し，そこから微調整を行うことで，デコーダー訓練の効率化の可能性を調査する．

表 4 は，デコーダー訓練のエポック数を変えて微調整したときの結果である．表の最下行（収束するまで訓練）をベースラインとし，デコーダー訓練を途中で中断して微調整したときの開発 PPL，BLEU スコアの変化を示している．なお，デコーダー訓練 0 エポックとは，デコーダー訓練を行わず，直接微調整を行った場合で，表 3 の再掲である．

デコーダー訓練のエポック数を減らしてゆくと，微調整の収束速度が遅くなり，開発 PPL も増加する傾向があるが，BLEU スコアはほとんど変化しない．実際，デコーダー訓練 65 エポックと有意な差が出たのは，3 箇所だけである．

どこまでデコーダー訓練を行うべきか，おそらくデータやハイパーパラメータによって異なるため，明確な基準を示すことはできないが，少なくとも本稿のデータでは，以下の結果となった．

(1) 最も訓練時間を短くするためには，デコーダー訓練を 3 エポック程度行うと，微調整にかかる時間との総和が短い（$3 + 18 = 21$ エポック）．
(2) 最も良いモデル（開発 PPL が最低のモデル）がほしい場合は，デコーダー訓練を 20 エポック程度行うと十分である．

4.4　小規模データ実験

次に，IWSLT-2015 の英語・ベトナム語データ (約 13 万文）で実験を行い，少資源設定での BERT エンコーダーの効果を確認する．コーパス以外の実験設定は，4.1 節と同じである．結果を**表 5** に示す．

小規模データにおいては，BERT エンコーダーを適用し，デコーダー訓練を行うだけで，ベースラインに比べ，開発 PPL は低下するが，BLEU スコア上は，悪化する．これは，大規模データ（4.2 節）と同じ傾向である．

一方，微調整を行うと，開発 PPL，BLEU スコアともにベースラインから大きく向上し，BLEU スコアは tst2013 で+3.45 向上している．表 3 の実験では，同じ設定（Transformer 基本モデルと BERT 利用 NMT（微調整，学習率 8e-5）の差）で newstest2015 は+1.41 であったので，BERT エンコーダーは少資源設定における翻訳品質向上に，より効果的である．

5.　議論

5.1　逆翻訳との対比

単言語コーパスを用いて，翻訳器の性能を上げる技術の一つに，逆翻訳がある [14]．これは，目的言語の単言語コーパスを，機械翻訳で原言語に翻訳し，疑似対訳文を作成，他の対訳文と混合して原言語→目的言語の翻訳器を学習する技術である．一方，今回用いた BERT エンコーダーは，原言語の単言語コーパスから学習するものなので，逆翻訳

表 4　デコーダー訓練のエポック数を変えた場合の開発 PPL と BLEU スコアの変化
Table 4　Changes of Dev. PPL and BLEU Scores when Number of Epochs in Decoder Training Change

デコーダー訓練		微調整		BLEU ↑			
エポック	開発 PPL ↓	エポック	開発 PPL ↓	2013	2014	2105	備考
0	—	33	4.28	0.13 (-)	0.10 (-)	0.12 (-)	デコーダー訓練なし（直接微調整）
1	33.05	34	4.23	26.67	28.77	30.16 (-)	
2	11.47	20	4.20	26.80	28.58	30.82	
3	8.12	18	4.10	**27.41**	28.93	30.78	
5	6.69	18	4.02	27.20	28.43 (-)	30.80	
10	5.50	15	3.96	27.33	28.59	30.42	
20	5.08	18	**3.91**	27.00	28.87	**30.92**	
30	4.89	10	3.92	27.18	28.39 (-)	30.70	
40	4.86	12	**3.91**	27.01	**29.04**	30.70	
50	4.81	11	**3.91**	27.02	28.65	30.77	
65	4.76	9	3.92	27.05	28.90	30.89	デコーダー訓練を収束するまで行った場合．検定用ベースライン

表 5　IWSLT-2015 データにおける実験結果
Table 5　Results of IWSLT-2015 Data

	システム	開発 PPL ↓	BLEU ↑ 2012	2013
ベースライン	Transformer 基本モデル	11.54	24.03	26.12
BERT 利用 NMT	提案法：デコーダー訓練直後	11.45	21.77 (-)	23.23 (-)
	微調整（学習率=8e-5）	**8.98**	**26.77** (+)	**29.57** (+)

を補完する技術と位置づけられる．

しかし，必要とする対訳コーパスの量に関しては，両者に若干の違いがある．BERT エンコーダー自体は，その学習に対訳コーパスは必要ないので，対訳が入手しにくい言語対でも適用することができる．ただし，単言語コーパスサイズは大規模なものが必要となるので，英語のような資源が豊富な言語から，少資源言語への翻訳に適用するのが効果的である．

一方，逆翻訳方式は，目的言語→原言語に翻訳する逆翻訳器を作成するために，対訳コーパスが必要である．対訳コーパスサイズが少なすぎると，逆翻訳した結果自体が信頼できなくなるため，ある程度のサイズは必要である [7]．中規模以上の資源を持つ言語対に適用するのがよい．

なお，2.2 節で述べた XLM は，単言語コーパスを用いて，原言語→目的言語→原言語へのオートエンコーダーと，目的言語→原言語→目的言語のオートエンコーダーを訓練し，両者をエンコーダー・デコーダー形式で接続することによって，教師なし機械翻訳を実現する枠組みを持つ．そのため，逆翻訳も含んでいるとみなすこともできる．もともと逆翻訳は，デコーダーを強化するための技術であるので，事前訓練に含めるのもリーズナブルである．

5.2　原言語コーパスを用いた NMT

原言語コーパスを用いて，機械翻訳の性能を上げる試みには，XLM 以外には以下のものがある．

文献 [21] は，原言語コーパスで語順変換モデル，対訳コーパスで翻訳モデルをマルチタスク学習する方法を提案している．文献 [4] は対訳コーパスによる訓練と同時に，原言語コーパス，目的言語コーパスでそれぞれオートエンコーダーを訓練した．どちらの方法も，訓練時に対訳コーパスと単言語コーパスを併用する必要がある．

本稿の設定は，原言語コーパスでエンコーダーを事前訓練し，対訳コーパスでデコーダー訓練と微調整を行うので，訓練手順は明確に段階づけられている．つまり，大規模な単言語コーパスは，事前訓練だけに適用され，デコーダー訓練，微調整時には対訳コーパスだけ使用する．そのため，非常に大規模な単言語コーパスよるメリットを享受しやすい．

5.3　事前訓練と再利用

今回用いた BERT モデルは，元々自然言語理解 (NLU) に適したように訓練されており，機械翻訳は目的外使用である．その意味では，事前訓練モデルを適用したというより，自然言語理解用モデルを機械翻訳に再利用したと言った方が適切だと考えている．

事前訓練と再利用の境界は，未学習のパラメータ数で判断される．今回の実験に用いたモデルのパラメータ数は，BERT エンコーダーが約 1.1 億個，Transformer デコー

ダーが約 8 千万個である．未学習のパラメータが 8 千万個もあると，微調整だけでは最適化できない．その場合，モデル再利用と位置づけ，本稿のような 2 段階最適化が有効となる．

いずれにしても，本稿の研究は，モデル再利用の一例となると考えている．

6. おわりに

本稿では，BERT エンコーダーを組み込んだ NMT を構築した．事前訓練済みのモデルパラメータに対して，未学習パラメータが多かったため，デコーダー訓練と微調整の 2 段階最適化を行い，対訳データのみから学習した NMT より高品質の機械翻訳を構築した．提案システムは，小資源設定でより効果を発揮することを確認した．さらに，デコーダー訓練の適切なエポック数について調査し，数エポックから数十エポックで十分であることも確認した．

今後は，さまざまなシステムのハイブリッドについても調査したい．

謝辞 本研究は総務省の情報通信技術の研究開発「災害時における多言語音声翻訳システムの高度化のための研究開発」の一環として行われました．

参考文献

[1] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *CoRR*, Vol. abs/1409.0473 (online), available from ⟨http://arxiv.org/abs/1409.0473⟩ (2014).

[2] Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L. and Tamchyna, A.: Findings of the 2014 Workshop on Statistical Machine Translation, *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, Association for Computational Linguistics, pp. 12–58 (online), DOI: 10.3115/v1/W14-3302 (2014).

[3] Cettolo, M., amd Sebastian Stüker, K. M., Bentivogli, L., Cattoni, R. and Federico, M.: The IWSLT 2015 Evaluation Campaign, *Proceedings of the International Workshop on Spoken Language Translation*, Da Nang, Vietnam, (online), available from ⟨http://workshop2015.iwslt.org/downloads/proceeding.pdf⟩ (2015).

[4] Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M. and Liu, Y.: Semi-Supervised Learning for Neural Machine Translation, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp. 1965–1974 (online), available from ⟨http://www.aclweb.org/anthology/P16-1185⟩ (2016).

[5] Clark, J. H., Dyer, C., Lavie, A. and Smith, N. A.: Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, pp. 176–181 (online), available from ⟨http://www.aclweb.org/anthology/P11-2031⟩ (2011).

[6] Devlin, J., Chang, M., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *CoRR*, Vol. abs/1810.04805 (online), available from ⟨http://arxiv.org/abs/1810.04805⟩ (2018).

[7] Edunov, S., Ott, M., Auli, M. and Grangier, D.: Understanding Back-Translation at Scale, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 489–500 (online), available from ⟨http://aclweb.org/anthology/D18-1045⟩ (2018).

[8] Freitag, M. and Al-Onaizan, Y.: Fast Domain Adaptation for Neural Machine Translation, *CoRR*, Vol. abs/1612.06897 (online), available from ⟨http://arxiv.org/abs/1612.06897⟩ (2016).

[9] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A. and Bengio, Y.: An Empirical Investigation of Catastrophic Forgetting in Gradient-based Neural Networks, *arXiv preprint*, (online), available from ⟨http://arxiv.org/abs/1312.6211⟩ (2013).

[10] Lample, G. and Conneau, A.: Cross-lingual Language Model Pretraining, *CoRR*, Vol. abs/1901.07291 (online), available from ⟨http://arxiv.org/abs/1901.07291⟩ (2019).

[11] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, Pennsylvania, USA, pp. 311–318 (online), available from ⟨http://aclweb.org/anthology/P02-1040⟩ (2002).

[12] Rajpurkar, P., Jia, R. and Liang, P.: Know What You Don't Know: Unanswerable Questions for SQuAD, *CoRR*, Vol. abs/1806.03822 (online), available from ⟨http://arxiv.org/abs/1806.03822⟩ (2018).

[13] Ramachandran, P., Liu, P. J. and Le, Q. V.: Unsupervised Pretraining for Sequence to Sequence Learning, *CoRR*, Vol. abs/1611.02683 (online), available from ⟨http://arxiv.org/abs/1611.02683⟩ (2016).

[14] Sennrich, R., Haddow, B. and Birch, A.: Improving Neural Machine Translation Models with Monolingual Data, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-2016, Volume 1: Long Papers)*, Berlin, Germany, pp. 86–96 (online), DOI: 10.18653/v1/P16-1009 (2016).

[15] Sennrich, R., Haddow, B. and Birch, A.: Neural Machine Translation of Rare Words with Subword Units, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp. 1715–1725 (online), available from ⟨http://aclweb.org/anthology/P16-1162⟩ (2016).

[16] Servan, C., Crego, J. M. and Senellart, J.: Domain specialization: a post-training domain adaptation for Neural Machine Translation, *CoRR*, Vol. abs/1612.06141 (online), available from ⟨http://arxiv.org/abs/1612.06141⟩ (2016).

[17] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *Proceedings of Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 3104–3112 (online), available from ⟨http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf⟩ (2014).

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I.: Attention Is All You Need, *CoRR*, Vol. abs/1706.03762 (online), available from

⟨http://arxiv.org/abs/1706.03762⟩ (2017).

[19] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S. R.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/pdf?id=rJ4km2R5t7⟩ (2019).

[20] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. and Dean, J.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *CoRR*, Vol. abs/1609.08144 (online), available from ⟨http://arxiv.org/abs/1609.08144⟩ (2016).

[21] Zhang, J. and Zong, C.: Exploiting Source-side Monolingual Data in Neural Machine Translation, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-2016)*, Austin, Texas, pp. 1535–1545 (online), available from ⟨https://aclweb.org/anthology/D16-1160⟩ (2016).

[22] Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S.: Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books, *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 19–27 (2015).