汎用分散表現BERTを用いたニューラル機械翻訳の検討

高橋 竜　　秋葉 友良　　塚田 元
豊橋技術科学大学
rtakahashi@nlp.cs.tut.ac.jp，akiba@cs.tut.ac.jp，tsukada@brain.tut.ac.jp

1 はじめに

機械翻訳では，通常，学習データとして対訳コーパスを必要とする．学習データが豊富なほど翻訳モデルの性能の向上が期待できるが，対訳コーパスは単言語コーパスと比較してデータの構築が困難であり，学習データの不足が問題となっている．そこで豊富かつ容易に入手が可能な単言語コーパスを用いて翻訳性能を向上させる手法が多く提案されてきている．最近，Bidirectional Encoder Representations from Transformers，BERT[1] と呼ばれる汎用の自然言語処理モデルが発表された．これは大規模な単言語コーパスで事前に学習されたモデルであり，学習済みのモデルに対して fine-tuning を行うことで高い性能を発揮する．実際，BERT を用いることで GLEU ベンチマークなどの様々なタスクで SOTA を達成している．しかし，報告されているタスクの多くは単言語で学習される自然言語理解であり，BERT を翻訳タスクに利用するという手法はあまり報告されていない．単言語コーパスで学習された強力な言語モデルである BERT をうまく翻訳タスクに利用することができれば，翻訳性能の大幅な改善が期待できる．

本論文では，事前学習済みの BERT をニューラル機械翻訳 (NMT) モデルに適用する．具体的には，NMT の入力言語側の単語分散表現に BERT の出力を用いる．BERT を分散表現に用いることで，従来の単語依存の分散表現ではなく，コンテキストも考慮された分散表現となる．これにより各単語の表現が柔軟になり，翻訳性能の向上が期待できる．また，BERT の出力についてのモデル拡張も行った．一般的に BERT は複数ある隠れ層の最終層の出力をそのまま使用するが，より翻訳に適した出力について調べるために出力の方法を変えていくつかのモデルを作成した．

翻訳モデルと比較して，BERT の学習には対訳コーパスを必要としないため，単言語コーパスがあれば，BERT を再学習させることができる．実験では，あらかじめ翻訳に用いるデータセットで BERT の fine-tuning を行った．これにより，BERT が fine-tuning する前と比べてデータセットのコンテキストにより強くなるため，データセットに対する翻訳性能が向上すると考えられる．

2 関連研究

2.1 BERT

BERT は 2018 年に Google が発表した自然言語処理モデルである．モデルは双方向の Transformer をベースとしたエンコーダーであり，トークン列 (単語またはサブワード) を入力として，入力の各トークンに対応する分散表現を出力する．BERT は転移学習を行うためのモデルであり，学習済みモデルに対してタスクに合わせたネットワークを追加して学習を行うことで様々なタスクに対処することができる．BERT は事前学習に MaskedLM(Masked Language Model) と次文予測 (Next Sentence Prediction) の２つのタスクを用いてモデルの学習を行っている．BERT は多くのタスクにおいて高い精度を記録していて，SQuAD や GLUE などの複数のベンチマークタスクにおいて State-of-the-art を達成している。学習済みのモデルが公開されているため，BERT は誰でも利用が可能となっている．最近は，BERT の問題点を改良したモデルの実装も行われている [2][3].

2.2 BERTを利用したニューラル機械翻訳

今村ら [4] は翻訳タスクに BERT を利用して性能の改善を達成している．具体的には，Transformer ベースの NMT モデルのエンコーダー部分を BERT に置き換えたモデルを提案した．ただし，このまま転移学習を行うと，NMT モデルのデコーダーのパラメータの数が非常に大きいため学習が発散してしまうので，安定した学習が難しいという問題が発生する．そこで

今村らは，未学習パラメータに対して 2 段階訓練を行うことで問題に対処した．これは，最初に BERT のパラメータは固定させた状態でデコーダのパラメーターのみを学習する．この状態での学習が収束したら次に BERT のパラメータを固定させず，エンコーダーとデコーダのパラメーターをまとめて学習する．これにより，従来の Transformer モデルに比べて BLEU スコアが向上している．

2.3 単言語コーパスを利用した NMT

Senrich ら [5] は単言語コーパスの利用方法として，逆翻訳を行って擬似対訳コーパスを作成する手法を提案した．これは，ターゲット側の単言語コーパスの逆翻訳を行い，ソース側の擬似文を生成し，これらを擬似対訳文として翻訳モデルの学習に利用するものである．西村ら [6] はソース側の大規模単言語コーパスを用いて分散表現を事前学習し，この分散表現を翻訳モデルに利用するモデルを提案している．しかし，この分散表現は文脈を考慮しない，単語依存の分散表現となっている．これに対して，BERT は同じ単語でも文脈毎に異なる分散表現を出力するため，より文脈に合わせた翻訳ができると考える．

3 提案手法

ベースラインの NMT は LSTM を用いたアテンション付きエンコーダーデコーダーモデルである．ベースラインモデルでは Embedding 層からの出力をそのまま使用している．

3.1 BERT モデル

3.1.1 BERT ベースラインモデル

入力側の単語の分散表現に BERT の出力を用いる．本論文で使用する BERT は隠れ層が 12 層あり，BERT ベースラインモデルでは通常の BERT の使い方にならって 12 層目の出力を利用する．BERT は入力単語列の各単語に対応した分散表現を出力するため，この出力を翻訳モデルでの分散表現として利用することができる．

3.1.2 BERT アベレージモデル

BERT ベースラインモデルでは 1 つの層の出力を利用していたのに対して，BERT アベレージモデルは複数の層の出力の平均を利用する．本研究では 2 つのモデルを作成した．1 つ目は出力に近い 4 つの層 (9〜12 層) のベクトルの平均を分散表現とするモデルである (BERT 4ave model)．2 つ目は 12 層全てのベクトルの平均を分散表現とするモデルである (BERT 12ave model)．

3.1.3 BERT ラーニングモデル

BERT 12ave モデルをさらに拡張したモデルを構築する．BERT 12ave モデルは各層の平均を出力としているため，全ての層に対する重みが等しくなっている．BERT ラーニングモデルは各層のベクトルごとに重み付けし，その重みを利用した加重平均を分散表現とするモデルである．重みは総和が 1 となっていて，学習可能なパラメータである．

3.2 BERT の fine-tuning

BERT は Wikipedia コーパスと BooksCorpus で事前学習された公開されている英語のモデルを使用している．本論文での翻訳実験では，モデルの学習に ASPEC コーパスを用い，テストに ASPEC コーパスと PATENT コーパスを用いている．そこで，翻訳モデルを学習する前に，学習済みの BERT を翻訳実験に用いるコーパスで fine-tuning を行った．fine-tuning された BERT を用いることで翻訳性能の改善が期待できる．BERT は単言語コーパスで学習可能なため，対訳コーパスを持たないデータでもこの手法が適用できる．単言語コーパスは直接翻訳モデルの学習に利用できないが，BERT の fine-tuning という点で貢献することが可能となる．

4 実験

提案した BERT モデルを使用して精度向上が図れるかを調べるために翻訳実験を行う．異なるドメイン上での性能低下を抑えれるかを調べるために複数のテストデータで性能評価を行った．さらに，ソース言語側単言語コーパスを利用して fine-tuning することでさらに精度向上が図れるかを検証していく．

表 1: 各モデルの実験結果

モデル		学習:ASPEC-200k BLEU テスト:ASPEC	テスト:PATENT	学習:ASPEC BLEU テスト:ASPEC	テスト:PATENT
ベースラインモデル	単語ベースライン	27.31	11.57	34.54	16.73
	BPE ベースライン	28.09	12.12	35.57	17.18
BERT モデル	BERT ベースライン	29.52	14.29	36.47	20.38
	BERT 4ave	30.26	16.15	36.66	20.72
	BERT 12ave	**31.07**	**17.26**	37.35	21.22
	BERT 12learn	30.90	17.03	**37.59**	**21.31**

4.1 データセット

学習には英日 ASPEC コーパスを使用する．また，学習データサイズを制限した場合 (ASPEC-200k) での実験も行う．2 つの学習データからデータサイズの違いによる影響を調べている．実験では，ASPEC コーパスで翻訳モデルを学習し，2 つのテストセットに対する結果を調べた．テストには ASPEC コーパスと PATENT コーパスを使用する．BERT の fine-tuning には英語 ASPEC コーパスと英語 PATENT コーパスを使用する．コーパスの詳細な情報を表 2 に示す．

英語側のコーパスは単語ベースと BPE ベースの 2 種類を用いてベースラインモデルを作成する．本実験で使用する BERT は BPE ベースのため，BERT モデルは全て BPE ベースとなっている．英語側のコーパスは前処理として NFKC 正規化を行った後，SMT ツールキット Moses に付属したトークナイザによりトークナイズを行い，小文字化を行った．BPE ベースはこれに加えて BERT の語彙に合わせてサブワード化を行った．日本語側のコーパスは NFKC 正規化を行った後，Mecab を使用して形態素解析を行い，小文字化を行った．

表 2: 各コーパスの文数

データセット	ASPEC	ASPEC_200k	PATENT
学習	1,000,000	200,000	3,000,000
テスト	1,812	-	899

4.2 モデルパラメータ

学習済みの BERT は GitHub で公開されているものを使用した．今回は，12 層の Transformer で隠れ層の次元数は 768 の基本モデルを使用する．本実験では NMT のエンコーダは 2 層の双方向 LSTM を，デコーダは 2 層の単方向 LSTM を使用した．分散表現と隠れ層の次元数は 768 とした．最適化には SGD を用いた．ミニバッチサイズは 64 とした．英語側の語彙は単語ベースではコーパスに頻出する 50000 語を使用し，BPE ベースでは BERT の語彙 30522 語と同じものを使用する．日本語の語彙はコーパスに頻出する 50000 語を使用する．10 エポック学習を行ったモデルの性能を評価する．BERT のパラメーターは学習中は固定させる．

4.3 各 BERT モデルの実験結果

各モデルの実験結果を表 1 に示す．ベースラインと比較すると全ての BERT モデルで性能が改善し，BERT を利用することの効果を確認できた．BERT モデル同士を比較すると，12 層全てのベクトルを利用した方が性能が高くなることがわかる．

BERT 12learn モデルの最終的な各層の重みを観察したところ，1 層目と 12 層目つまり最初の層と最終層の重みが特に高くなった．これは 4.4 節での実験で作成する各 BERT 12ave モデルでも同じ傾向であった．

4.4 BERT の fine-tuning による影響

4.4.1 実験設定

翻訳モデルを学習する前に事前学習された BERT を単言語コーパスを用いて fine-tuning する．3 種類のコーパスで fine-tuning を行う．

1. ASPEC(BERT-A)
2. PATENT(BERT-P)
3. ASPEC+PATENT(BERT-AP)

表 3: fine-tuning による性能の比較

モデル	BERT	学習:ASPEC-200k BLEU テスト:ASPEC	テスト:PATENT	学習:ASPEC BLEU テスト:ASPEC	テスト:PATENT
BERT ベースライン	BERT	29.52	14.29	36.47	20.38
	BERT-A	30.05	15.39	**37.82**	20.09
	BERT-P	29.93	14.49	36.13	20.26
	BERT-AP	**31.16**	**15.92**	36.78	**20.73**
BERT 12ave	BERT	31.07	**17.26**	37.35	**21.22**
	BERT-A	30.93	16.03	**38.23**	20.79
	BERT-P	**31.35**	16.80	37.42	20.62
	BERT-AP	31.2	16.95	37.76	21.02
BERT 12learn	BERT	30.90	17.03	37.59	21.31
	BERT-A	32.00	17.37	**38.31**	21.05
	BERT-P	31.34	17.35	37.8	21.10
	BERT-AP	**32.17**	**17.45**	37.91	**21.45**

コーパスサイズはそれぞれ 100 万，300 万，400 万となっている．学習率などの設定は文献にもある，BERT の fine-tuning の設定を参考にした．また，翻訳モデルには BERT モデル 4 つのうち、BERT 4ave モデル以外の 3 つのモデルを使用した．

4.4.2 実験結果

各モデルの実験結果を表 3 に示す．ASPEC コーパスで fine-tuning したモデルは，ASPEC テストデータの翻訳性能が特に改善した．これは fine-tuning によって，ASPEC ドメインに強い BERT になったためと考えられる．PATENT コーパスで fine-tuning したモデルは ASPEC テストデータと PATENT テストデータの性能の下がり幅が最も小さくなっている．これは学習で ASPEC コーパスを使用しているにもかかわらず，BERT-P が PATENT コーパスの情報を保持しているためであると考えられる．ASPEC コーパスと PATENT コーパスで fine-tuning した BERT-AP はどちらの特徴も持っているため，どちらのテストデータに対しても高い性能になった．BPE ベースラインと比較すると，最善の BERT モデルは ASPEC テストセットで +2.74BLEU，PATENT テストセット で +4.27BLEU の性能改善を示した．

5 おわりに

本研究では NMT に BERT の出力を分散表現として利用することで翻訳性能が向上することを確認した．特に，ドメイン外データに対しての改善が大きく，BERT のドメインに依存しない汎用性が確認できた．また，BERT の最終層だけを利用するより，BERT の隠れ層全ての出力を利用する方が性能が向上した．BERT weight モデルの重みの結果は重要な知見であり，BERT を用いたモデルのさらなる性能向上へ生かすことができると考える．今後の課題としては，LSTM だけではなく，Transformer ベースの NMT モデルへの適応や，ターゲット側の分散表現にも BERT を利用することができないかについての検討も行っていきたいと考えている．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[2] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. 2020. XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv preprint arXiv: 1906.08237.

[3] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

[4] Kenji Imamura, Eiichirou Sumita. 2019. Recycles a Pretrained BERT Encoder for Neural Machine Translation. Proceedings of the 3rd Workshop on Neural Generation and Translation, D19-5603, pp.23-31, Hong Kong. Association for Computational Linguistics.

[5] Rico Sennrich, Barry Haddow and Alexandra Birch. 2015. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, P16-1009, pp.86-96, Berlin, Germany. Association for Computational Linguistics.

[6] Tomoki Nishimura, Tomoyoshi Akiba. 2017. Addressing Unknown Word Problem for Neural Machine Translation using Distributed Representations of Words as Input Features. International Conference on Advanced Informatics: Concept Theory and Applications, Bali, Indonesia.