機械翻訳における単語埋め込み共有時の問題の言語埋め込み導入による緩和

毛 剣楠 [1]　　松本 忠博 [1]
[1] 岐阜大学 大学院
z4525087@edu.gifu-u.ac.jp　　tad@gifu-u.ac.jp

概要

単語埋め込みの共有はニューラル機械翻訳モデルのパラメータ数を大幅に削減し，翻訳精度を向上させる．本研究では，単語埋め込み共有時に両言語間の共通単語が部分的に負の影響を与えることを予備実験から予想し，その緩和ために言語埋め込みを導入して単語埋め込みに加えることを提案する．これにより 2 言語間の共通単語の埋め込みはエンコーダーとデコーダーで異なる分布を示した．日中，日英言語ペアの翻訳に対して提案手法を適用したところ，言語や言語埋め込みの付与方法にもよるが概ね翻訳精度が向上する傾向が見られた．

1　はじめに

強力なアーキテクチャ [1] の登場により，ニューラル機械翻訳（NMT）が最も有望な機械翻訳方式となっており，それに伴って，NMT による GPU メモリの必要性が高まっている．従来の NMT モデル [1, 2] では，単語埋め込みは NMT モデルの最も重要なモジュールの 1 つであり，単語埋め込みを用いて単語の構文的，意味的属性を捕捉する．NMT モデルでは一般的に，エンコーダー入力埋め込み，デコーダー入力埋め込み，デコーダー出力埋め込みの 3 つの行列が使用される．これらの埋め込みはモデルパラメータの大部分を占め，トレーニング時に多くの GPU メモリを占有している．

単語埋め込みの GPU メモリの使用を減らすために，NMT 単語の表現に用いるパラメータを減らすいくつかの方法が提案されている．Press[3] は単語埋め込みのパラメータを大幅に減らすために「three-way weight tying」と呼ばれる重みの共有方法を提案したが，これは実用 NMT の新しい事実上の標準となった．「three-way weight tying」は 3 つの単語埋め込みを表すために 1 つの行列を使用し，源言語と目的言語で共通の単語が 1 つの単語ベクトルを共有する．この方法は源言語と目的言語の共通サブワードの NMT にも適用でき，英独仏など，同じ文字が多い言語ペアではうまく機能した [1]．

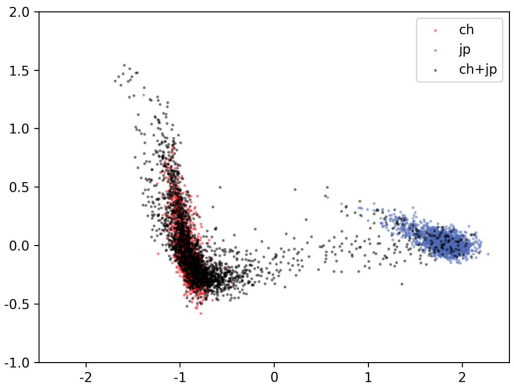
本研究では，単語埋め込み共有が翻訳結果に負の影響を与えていないか調べるための予備実験を行った．翻訳結果の文中には源言語の単語が存在する場合があった．

また，PCA アルゴリズムを用いて単語埋め込みを可視化し，中→日および英→日翻訳モデルにおける単語埋め込みを比較した．予備実験の結果と PCA 可視化のグラフに基づいて言語的な観点から，Transformer の単語埋め込みの共有が翻訳結果に悪影響を及ぼしていることが予想された．その緩和のために，Transformer-base モデルに最大 2048 個の訓練パラメータを追加する 3 種類の手法を提案し，評価実験を行なった．実験の結果，パラメータの付与の方法と翻訳対象となる言語，翻訳方向にもよるが，概ね翻訳精度が向上する傾向が見られた．

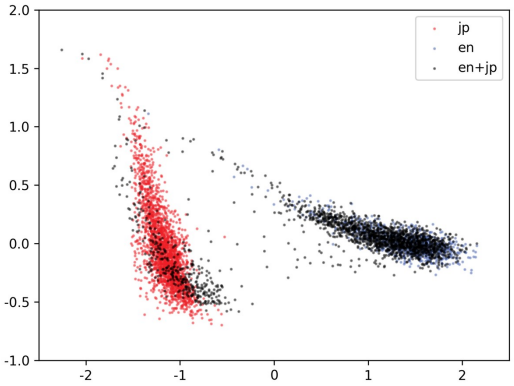
2　予備実験

max-tokens は，1 回のイテレーションで使用される源言語および目的言語の token の最大数（バッチサイズ）である．Popel ら [4] の実験結果から，max-tokens が Transformer の翻訳精度に大きな影響を与えていることが分かった．この影響は，イテレーション回数の増加によって大きく変化することはない．まず，共通単語の多い言語ペアである中日と，比較のため英日に対して，max-tokens を 1024 に設定して実験[1)] を行った．翻訳結果に存在した源言語の単語の数を表 1 に示す．中日翻訳と比較すると，英日翻訳では間違った単語が大量に出現することはなかった．

1）イテレーションは 20 万，他のパラメーターは 4.2 節と同じである．1024 より大きな値に設定した場合は，このような翻訳エラーは発生しない．



中→日翻訳



英→日翻訳

図 1　単語埋め込み分布の比較

Mikolov ら [5] によると，単語埋め込みを共有しない場合でも，異なる言語の単語はベクトル空間において類似した位置に分布することが知られている．我々は，前述の中日・英日翻訳実験での最適モデルの単語埋め込みを抽出し，頻度上位 5000 単語を PCA アルゴリズムを用いて次元削減した後，2 次元グラフにプロットした．図 1[2] から明らかなように，同一言語に固有単語が分布しているのに対して，共通単語は両言語間に分布しており，英日翻訳と比較して，中日翻訳ではより多くの共通単語が両言語間に分布している．

3　言語埋め込み

本研究は，図 1 の中日翻訳と英日翻訳における単語埋め込みの分布，及び BERT[6] におけるセグメント埋め込みの示唆に基づく言語埋め込み（Language Embedding，LE）を提案する．エンコーダーとデコーダーにそれぞれ異なる言語埋め込みを加え，同一の単語がエンコーダー側とデコーダー側で異なる表現を持つようにする．提案手法では，コーパス中の単語を 3 つのクラス[3] に分類し，言語埋め込みをどのように付与するかによって 3 つの手法を提案する．

表 1　max-token=1024 の時，翻訳実験結果

	訳文の長さ	間違い単語数
中→日	79801	16
日→中	63623	40
英→日	33994	1
日→英	48202	1

提案手法 1　エンコーダー側では，源言語固有単語と，源言語・目的言語の共通単語を 2 つに分類し，源言語固有単語の言語埋め込みを 0（更新なし）とする．デコーダー側では，目的言語固有単語と，源言語・目的言語の共通単語を 2 つに分類し，目的言語固有単語の言語埋め込みを 0（更新なし）とする．LE はそれぞれエンコーダーとデコーダーの端で計算され，式 (1) のようになる．この時，モデルにおいて増加するパラメータ数は $2 \times 2 \times 512$[4] $= 2048$ になり，計算すべきパラメータ数は 1024 になる．

$$\begin{aligned} Word_{LE}^{Encoder} &= Word_{\epsilon\,[\text{src=0,src+tgt}]} \times LE_{2\times 512}^{Encoder} \\ Word_{LE}^{Decoder} &= Word_{\epsilon\,[\text{tgt=0,src+tgt}]} \times LE_{2\times 512}^{Decoder} \end{aligned} \tag{1}$$

提案手法 2　エンコーダー側では，源言語固有，源言語・目的言語の共通単語を 1 つに分類する．すなわち，エンコーダーへの入力単語は 1 種類である．デコーダー側では，目的言語固有，源言語・目的言語の共通単語を 1 つに分類する．すなわち，デコーダーへの入力単語は 1 種類である．LE はそれぞれエンコーダーとデコーダーの端で計算され，式 (2) のようになる．このとき，モデルが増加するパラメータ数は $2 \times 512 = 1024$ になり，計算すべきパラメータ数は 1024 になる．

$$\begin{aligned} Word_{LE}^{Encoder} &= Word_{\epsilon\,[(\text{src,src+tgt})]} \times LE_{1\times 512}^{Encoder} \\ Word_{LE}^{Decoder} &= Word_{\epsilon\,[(\text{src,src+tgt})]} \times LE_{1\times 512}^{Decoder} \end{aligned} \tag{2}$$

提案手法 3　エンコーダー側では，源言語固有単語，源言語・目的言語の共通単語を 2 つに分類する．

2）　ch：中国語固有単語．jp：日本語固有単語．
en：英語固有単語．en+jp：英語と日本語の共通単語．
ch+jp：中国語と日本語の共通単語．

3）　クラス 1：源言語に固有の単語（src）
クラス 2：目的言語に固有の単語（tgt）
クラス 3：源言語と目的言語に共通の単語（src+tgt）

4）　transformer-base の単語埋め込みは 512 次元．

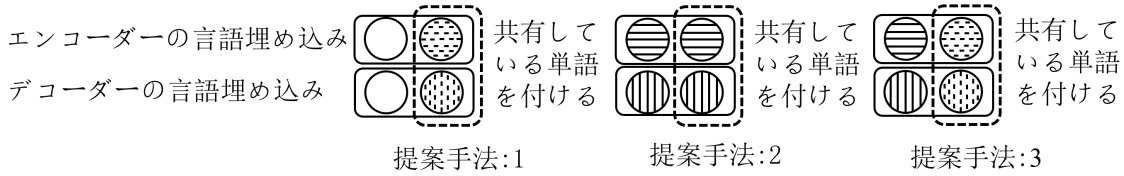


図 2 提案手法．上の長方形はエンコーダーの言語埋め込み，下の長方形はデコーダーの言語埋め込みである．長方形左側の円を固有単語の言語埋め込み，長方形右側の円を共通単語の言語埋め込みとする．提案手法ごとに，円に紋様がない場合はその言語埋め込みが 0 であり，更新しないことを表す．同一長方形中の円の紋様が同じ場合は，固有単語と共通単語が同じ言語埋め込みを持つことを表す．各円で表される言語埋め込みのサイズは，単語埋め込みのサイズと同じである．

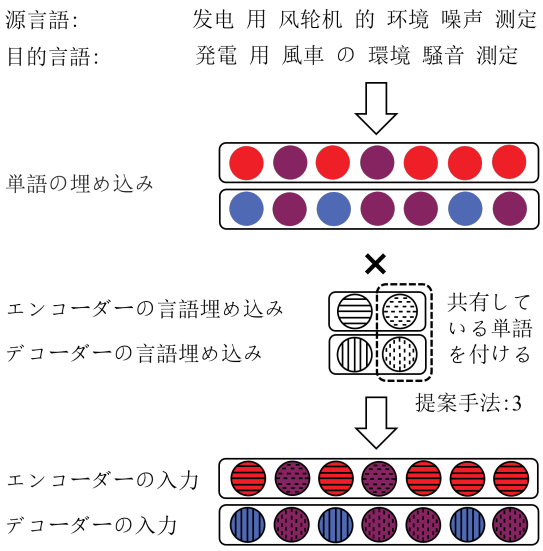


図 3 モデルへの入力過程．赤い円は源言語の固有単語埋め込み，青い円は目的言語の固有単語埋め込み，紫の円を源言語と目的言語の共通単語埋め込みを表す．実験に用いたコーパスにおいて「環境」,「測定」などは共通単語となった．

デコーダー側では，目的言語固有単語，源言語・目的言語の共通単語を 2 つに分類する．LE はそれぞれエンコーダーとデコーダーの端で計算され，式 (3) のようになる．このとき，モデルが増加するパラメータ数は $2 \times 2 \times 512 = 2048$ になり，計算すべきパラメータ数は 2048 になる．

$$\begin{aligned} Word_{LE}^{Encoder} &= Word_{\epsilon\,[\mathrm{src,src+tgt}]} \times LE_{2\times 512}^{Encoder} \\ Word_{LE}^{Decoder} &= Word_{\epsilon\,[\mathrm{tgt,src+tgt}]} \times LE_{2\times 512}^{Decoder} \end{aligned} \tag{3}$$

以上の 3 つの提案手法を図 2 に示す．
LE と単語埋め込みの融合方式を式 (4) に示す．

$$\begin{aligned} Encoder_{input} &= Word_{embedding} + Word_{LE}^{Encoder} \\ Decoder_{input} &= Word_{embedding} + Word_{LE}^{Decoder} \end{aligned} \tag{4}$$

提案手法の詳しい過程を図示すると図 3 のようになる．エンコーダーとデコーダーの入力単語を単語埋め込みに変換し，それに各単語の LE を加えて，モデルに入力する．

4 翻訳実験

実験では，コーパス中の全角記号を半角記号に変換した後，単語分割を行った．中国語文は LTP[7] で単語分割を行い，英語文は Stanza[8] で，日本語文は Ginza[9] で単語分割を行なった．データクリーニングなど前処理は行っていない．train データを用いてモデルの訓練を行い，中日，日中，英日，日英の 4 つ方向の翻訳実験を行った．

4.1 コーパス説明

コーパスには ASPEC-JC および ASPEC-JE[10] を用いた．ASPEC-JC は，日本の科学論文を手作業で中国語に翻訳して構築されている．ASPEC-JE は日本語と英語の科学論文の要約から構成されており，大量の英文名詞と数字を含む．語彙は train データの単語分割結果に基づいて作成した．使用データ数と語彙サイズを表 2 に示す．

表 2 用いた ASPEC コーパスデータ数

		ASPEC-JC	ASPEC-JE
文数	train	67,2315	1,000,000
	test	2,107	1,812
	dev	2,090	1,790
語彙サイズ	日本語	174,960	199,760
	中国/英語	303,456	374,232
	共有しない	478,416	573,992
	共有する	416,936	500,296

4.2 実験設定

ベースラインとして使用するモデルの構造は Transformer-base[1] と同じであり，実装には Fairseq[11] を用いた．学習率は 0.0007 とし，Adam を用いて学習を行なった．max-tokens は 2048 とした．SEED は 1 に固定し，訓練は 15 万イテレーショ

表 3　BLEU および PPL による評価結果

	中→日		日→中		英→日		日→英	
評判方法	BLEU(↑)	PPL(↓)	BLEU(↑)	PPL(↓)	BLEU(↑)	PPL(↓)	BLEU(↑)	PPL(↓)
ベースライン	36.67	5.38	27.32	9.25	33.75	6.91	23.05	10.67
提案手法 1	36.95	5.25	26.84	9.23	33.49	6.79	**24.06**	10.65
提案手法 2	37.36	**5.21**	**27.38**	**9.14**	**34.21**	**6.61**	23.24	10.84
提案手法 3	**37.66**	5.25	27.33	9.31	33.73	6.88	23.21	**10.64**

ン実行した．

提案手法の実験では，3 つの提案手法を使い，パラメーターはベースライン実験と同じ設定にした．

dev データにおいて valid loss が最小のモデルを保存した．これを最適モデルと呼ぶ．dev データでの Perplexity と test データに対する BLEU スコア（翻訳精度）により最適モデルを評価した．

4.3　実験結果

表 3 に BLEU スコアと Perplexity（PPL）を示す．

実験の結果，BLEU スコアは，手法 1 では日英翻訳で 1.01 ポイント向上したが，日中では 0.48 ポイント低下した（平均 0.14 ポイント向上）．手法 2 では中日翻訳で 0.69 向上したほか，すべての翻訳で向上した（平均 0.35 ポイント向上）．手法 3 は中日翻訳で 0.99 ポイント向上したが，英日ではわずかに低下した（平均 0.29 ポイント向上）．

中日翻訳におけるベースラインの最適モデルと提案手法 3 の最適モデル中の単語埋め込みを抽出し，PCA アルゴリズムを用いて次元削減して 2D グラフにプロットしたが，言語埋め込みがエンコーダやデコーダの入力に与える影響がグラフからははっきりと見えなかった．そこで，単語埋め込みを 512 次元から 3 次元に削減して 3D グラフにプロットした（図 4）．

この図において，baseline はベースライン実験の単語埋め込み，word_embedding は提案手法 3 の単語埋め込み，encoder_input は提案手法 3 のエンコーダー入力，decoder_input は提案手法 3 のデコーダー入力を表す．ベースライン実験での max-tokens=2048 のときの単語埋め込みの空間分布は，1024 のとき（2 節）とほぼ同じである．

図 4 を見ると，ベースラインと提案手法 3 には明らかな違いはない．エンコーダー入力を見てもベースラインの入力は単語埋め込みであるのに対し，提案手法 3 の入力は明らかに変化し，3 次元空間で線形分離可能な状態となった．この変化から，エンコーダーに入力される共通単語に言語埋め込みを付加すると，共通単語は，単語埋め込み中においてエンコーダーの入力言語（中国語）の方にシフトすると考えられる．同様に，デコーダーに入力される共通単語は，単語埋め込み中ではデコーダーの入力言語（日本語）に傾いている．共通単語は，エンコーダーとデコーダーにおける単語埋め込みが異なる表現を持つことを示している．

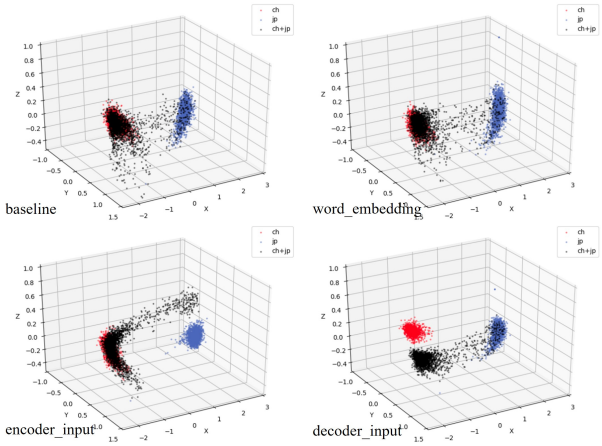


図 4　ベースラインと提案手法 3 を用いた場合の中日翻訳の単語埋め込み分布の比較

5　終わりに

Transformer において単語埋め込みを共有した場合に，源言語と目的言語の単語埋め込みの分布から，両言語の共通単語が翻訳精度に悪影響を及ぼしているのではないかと予想した．本研究では両言語を区別するための少量のパラメータ（言語埋め込み）を単語埋め込みと組み合わせて使用すること提案し，評価実験を行なった．その結果，翻訳対象となる言語と翻訳方法，言語埋め込み付与の仕方などの条件により違いはあるが，概ね翻訳精度が向上する傾向が見られた．

今後は，提案手法による実験結果のばらつきに対して，さらに頑健な解決策を提案するとともに，文字レベルとサブワードレベルの翻訳実験を行う予定である．

参考文献

[1] 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 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA**, pp. 5998–6008, 2017.

[2] Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. The best of both worlds: Combining recent advances in neural machine translation. **arXiv preprint arXiv:1804.09849**, 2018.

[3] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers**, pp. 157–163. Association for Computational Linguistics, 2017.

[4] Martin Popel and Ondrej Bojar. Training tips for the transformer model. **Prague Bull. Math. Linguistics**, Vol. 110, pp. 43–70, 2018.

[5] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In **1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings**, 2013.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. **CoRR**, Vol. abs/1810.04805, , 2018.

[7] Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. N-ltp: A open-source neural chinese language technology platform with pretrained models. **arXiv preprint arXiv:2009.11616**, 2020.

[8] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations**, 2020.

[9] 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. **arXiv e-prints**, 2019.

[10] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. Aspec: Asian scientific paper excerpt corpus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, **Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)**, pp. 2204–2208, Portorož, Slovenia, may 2016. European Language Resources Association (ELRA).

[11] 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 NAACL-HLT 2019: Demonstrations**, 2019.