人工データでの事前学習によるニューラル機械翻訳の性能向上

田村 弘人　平澤 寅庄　金 輝燦　岡 照晃　小町 守
東京都立大学
{tamura-hiroto, hirasawa-tosho, kim-hwichan}@ed.tmu.ac.jp
{teruaki-oka, komachi}@tmu.ac.jp

概要

転移学習において非言語データで**事前学習**することで，言語情報以外のどのような特性が転移されるかの研究が行われている [1, 2, 3, 4]．ニューラル機械翻訳 (NMT) においては Aji ら [5] が**人工データ**で事前学習した場合，事前学習なしのモデルよりも高い性能を示したと報告している．しかし彼らはそれぞれ 1 種類の人工データ，事前学習法でしか実験していない．本研究では様々な特性を持つ人工データを用意し，2 つの目的関数を用いて事前学習した場合の NMT での性能を比較調査する．また転移学習に対する ablation study を行い，各事前学習データ，事前学習法でモデルの各コンポーネントへの影響を調査した．結果として，トークンの頻度情報を持たせた人工データで事前学習したモデルは，実データで事前学習したものよりも高い性能を示した．

1　はじめに

転移学習は NMT の性能向上に効果的な手法であり，特に少資源状況において顕著な性能改善が見られる [6, 7, 8, 9, 10]．言語情報の知識転移が性能向上の主な貢献とされる一方，様々なタスクにおいて人工データで事前学習することで，言語情報以外のどのような特性が転移されるかが調査されている．

Chiang と Lee [3] は Transformer [11] ベースの言語モデルで，事前学習データ内の意味以外のどのような特性が下流タスクの性能に影響するかを調査した．人工データで事前学習し GLUE [12] タスクでの性能を評価した結果，事前学習なしの場合よりも高い性能を示した．Transformer ベースの系列変換モデルでは Krishna ら [4] が要約タスクにおいて，人工データで事前学習したモデルが実データで事前学習した場合に匹敵する性能を示したと報告している．

Aji ら [5] は NMT において，人工データを用いた**自己符号化**で事前学習することで事前学習なしのモデルよりも高い翻訳性能を示し，特に少資源言語対において顕著な性能向上が見られた．しかし彼らの実験はランダムな数字列で構成された人工データでのみ実験しており，他の特性を備えた人工データでの実験は未調査である．また事前学習法として自己符号化を採用しているが，他の学習法での翻訳性能に対する影響も未知である．

そこで本研究では，下流タスクを NMT として様々な種類の人工データを用いて事前学習することで，NMT でどのような特性を持った人工データが翻訳性能に効果的かを調査する．また事前学習法として，自己符号化と MAsked Sequence to Sequence pre-training (**MASS**)[1] [9] を選択し，事前学習法ごとの人工データの特性の影響も調査する．さらに転移学習に対して ablation study を行い，事前学習時のデータセットや学習法ごとにモデルのどのコンポーネントの転移が性能に貢献しているか，反対にどれをファインチューニングした方が良いかを調査する．本研究での主な貢献は以下の通りである：

1. 対訳データが少資源の場合，人工データでの事前学習により，事前学習なしのモデルよりも高い性能を示した[2]．
2. 事前学習法が自己符号化の場合，トークンの頻度情報を含めた人工データで事前学習すると，実データの場合よりも高い性能を示した．
3. 事前学習法が MASS の場合，人工データでの学習は不十分であり実データで訓練する必要があることを示した．
4. Ablation study により事前学習法が自己符号化の場合，頻度情報を学習した encoders が下流タスクの主な性能の要因であることを明らかにした．

1）　MASS は Song ら [9] によって提案された系列変換の masked language model であり，多量の単言語データで事前学習することで特に少資源 NMT で顕著な性能向上が確認されている．
2）　多資源状況では性能向上が見られなかったため，対訳データを 3 万文対，10 万文対として実験を行った．

2 事前学習データ

2.1 実データ

本研究では英語からドイツ語への翻訳タスクにおける人工データを用いた事前学習の影響を分析する．ファインチューニング時に使用する対訳データは WMT14 [13] の英独翻訳タスクから獲得している．そのため事前学習に用いる英語，ドイツ語データもドメインの観点から同じく WMT14 の対訳データから抽出し，各言語の単言語データとして扱った．

English (En) 全対訳データの英語側から，ファインチューニング時に使用する訓練データと重複しない文をランダムに抽出する．

German (De) “English” と同様にドイツ語データを抽出する．

English + German (En + De) “English”, “German” でのデータサイズが約 5,000 万トークンになるように，それぞれの言語データから文を抽出し，混合する．

2.2 人工データ

本研究で使用する人工データは全て数字列で構成される．具体的には，0 から（ファインチューニングに使用する対訳データの語彙サイズ − 1）の範囲の整数が人工データの語彙に含まれる[3]．系列長は全て 128 である．

Random 各数字は語彙の一様分布から独立に抽出され，系列が作成される．

Unigram ファインチューニングで使用する対訳データから unigram 分布を作成し，その分布に沿って数字を独立に抽出し系列を作成する．

Zipf Zipf 分布（式 1）から独立に各数字が抽出され，系列が作成される．

$$f(k;s,N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)} \tag{1}$$

ここで N はトークンの要素数，k はトークンの頻度順位，s は分布を特徴付ける指数であり，s が小さいほど滑らかな分布になる．本実験では $s = 1.0$ とした．“Unigram” では対訳データから分布を求めているが，この手法では一切実データを用いていない．

3） ファインチューニングに使用する対訳データの語彙サイズが 5,000 の場合，人工データの作成には 0~4,999 の範囲の整数が使用される．

表 1 異なる対訳データサイズにおける，事前学習法，事前学習データごとのテストセットでの BLEU スコア．“N/A” は事前学習なしのモデルを表す．“AE” は自己符号化を表す．

	3 万		10 万	
事前学習データ	**AE**	**MASS**	**AE**	**MASS**
N/A	3.1	3.1	**15.0**	**15.0**
En	8.3	8.8	13.5	14.3
De	8.2	8.5	13.3	14.7
En + De	8.2	**8.9**	13.3	14.3
Random	7.5	5.9	13.5	12.5
Unigram	**8.9**	5.0	13.5	14.2
Zipf	8.8	8.0	13.8	13.8

3 実験設定

事前学習データのデータサイズは約 1 億トークンである．ファインチューニングに使用する対訳データは全対訳データ約 450 万文対からランダムに 3 万文対と 10 万文対を抽出し，それぞれのデータサイズで NMT モデルの訓練を行った．開発セットは newstest2013，テストセットは newstest2014 を使用した．全ての実データは Moses[4] [14] で正規化，トークン化し，その後 BPE [15] でサブワード化を行った．BPE の語彙サイズは対訳データが 3 万文対，10 万文対の場合それぞれ 8,000，16,000 である．また事前学習時に使用する実データは上で学習した語彙を用いてサブワード化する．

全ての実験は Song ら [9] のコード[5]を用いて行った．モデルは Transformer (base) を使用し，事前学習済モデルの重みで初期化後，ファインチューニングを行った．初期化の際，語彙の割り当ては frequency assignment [5] を採用している．主なハイパーパラメータは付録 A に示す．事前学習の更新回数は 10 万ステップである．ファインチューニング時は early stopping を採用しており，開発セットでのロスが 10 エポック間減少しなかった場合，訓練を終了する．事前学習は各データセットで 1 回だけ行い，ファインチューニングと事前学習なしのモデルの訓練は異なるシードで 3 回行った．評価は SacreBLEU[6][7] [16] を用いて，case-sensitive BLEU を計算した．報告する全てのスコアは 3 つのシードでの平均値である．

4） https://github.com/moses-smt/mosesdecoder
5） https://github.com/microsoft/MASS/tree/master/MASS-supNMT
6） https://github.com/mjpost/sacrebleu
7） BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+test.{wmt13,wmt14}+tok.13a+version.1.5.1

4 結果

表 1 にファインチューニング時の各対訳データサイズにおける，事前学習法，事前学習データごとのテストセットでの BLEU スコアを示す．

対訳データサイズ：3 万 事前学習法が自己符号化の場合は，全ての事前学習データで，事前学習なしの場合よりも性能向上が見られる（貢献 1）．実データで事前学習したモデルは “Random” よりは良い性能であるが，下から 2 つの人工データでのモデルより低い性能を示している．“Unigram” と “Zipf” は頻度情報のみを考慮したデータセットであることから，頻度情報のみを学習することで十分な性能が発揮できることを示している（貢献 2）．事前学習法が MASS の場合は，人工データで事前学習したモデルは事前学習なしのものよりも高い性能ではあるが，実データで事前学習したモデルが人工データでの事前学習よりも明らかに高い性能を示している．実データは共起性や構造性など頻度以外の情報も含んでいる．よって MASS では頻度情報は性能に貢献するが，頻度以外の情報も学習することで更なる性能向上が達成されることを示している（貢献 3）．

対訳データサイズ：10 万 自己符号化の場合，全ての事前学習データで，事前学習なしの場合よりも性能が悪化しており，どのデータでも同程度のスコアが得られる．MASS の場合も全事前学習データで，事前学習なしの場合よりも低いスコアしか達成できないが，人工データよりも実データで事前学習した方が良いという傾向はデータサイズが 3 万の時と同じである．これらから対訳データサイズが 10 万の場合は，各事前学習法，事前学習データで性能向上が見られなかった．

以上の結果から 5 節では対訳データサイズを 3 万とし，各事前学習法，事前学習データによるモデルの各コンポーネントへの影響を分析する．

5 分析

4 節では各事前学習法により，実データで訓練した場合と，人工データでの場合とで異なる結果を確認した．これを受け，事前学習データが “En” と “Zipf” の場合の各事前学習法での挙動を確認する．具体的にはモデルを embeddings (emb), encoders (enc), cross-attentions (x-attn), decoders (dec)[8] の大きく 4 つのコンポーネントに分けて転移学習に対する ablation study を行う．

8） emb は encoder 側と decoder 側，両方の embeddings を対象としている．enc, dec は self-attentions と layer-norms, feed-forward-networks を含む．x-attn は layer-norms を含む．

表 2 各コンポーネントを転移した時の，事前学習法，事前学習データごとのテストセットにおける BLEU スコア．“✓” は該当するコンポーネントを転移していることを意味する．

	コンポーネント				AE		MASS	
行	emb	enc	x-attn	dec	En	Zipf	En	Zipf
1					3.1	3.1	3.1	3.1
2		✓			6.3	6.3	5.2	4.6
3			✓		1.0	0.9	1.1	3.7
4				✓	7.0	7.3	3.1	3.3
5		✓	✓		7.9	8.1	7.0	5.8
6		✓		✓	7.2	8.0	5.1	5.5
7			✓	✓	5.8	4.1	4.2	4.6
8		✓	✓	✓	8.6	9.1	7.7	7.1
9	✓				5.4	3.7	4.0	3.6
10	✓	✓			6.6	6.5	7.4	4.6
11	✓		✓		1.8	1.0	6.1	4.3
12	✓			✓	7.7	7.5	3.9	4.4
13	✓	✓	✓		8.7	8.7	9.1	6.5
14	✓	✓		✓	7.2	7.6	7.4	5.5
15	✓		✓	✓	4.9	4.5	6.6	5.7
16	✓	✓	✓	✓	8.3	8.8	8.8	8.0

表 2 に選択的に各コンポーネントを転移し，ファインチューニングした時の BLEU スコアを示す．また，表 3 に全てのコンポーネントを転移した後，ある部分のみをフリーズしてファインチューニングした場合の BLEU スコアを示す．

emb の転移の影響 表 2 から事前学習法が自己符号化の場合，emb のみを転移した場合を除いて emb の影響はわずかである．emb 以外を全て転移した場合では，全てのコンポーネントを転移した場合よりも高い性能を出している（8 行目）．一方で事前学習データが “En” の時，“Zipf” の場合に比べて emb は性能に貢献している（1, 9 行目）．表 3 から，emb をフリーズすると両事前学習データとも性能が悪化しているが，“Zipf” よりも “En” の方が悪影響を受けている．これは emb の重みを enc 側と dec 側とで共有しており，“En” の場合 emb は英語に最適化されているが，ドイツ語の emb としても使用されているためである．これに対し “Zipf” では言語横断的な頻度情報のみを emb は学習しているため性能悪化が小さいと考えられる．以上から，自己符号化では emb の情報を少しは転移しているが，ファインチューニングで重みが再学習されると考えられる．そして，主な性能貢献は emb 以外のコンポーネントの影響によるものであり，それはトークンの頻度の学習で達

成されると推測できる．

MASS の場合，表 2 より明らかに emb を転移すると性能向上している．特に事前学習データが “En” だと “Zipf” よりも emb の性能に対する貢献が強いと確認できる．表 3 から emb をフリーズした時，両事前学習データとも性能が悪化しているが，“En” よりも “Zipf” の方が悪影響を受けている．これは “En” では文脈的な emb が得られたが，“Zipf” では無理に文脈的な emb を獲得しようとして低品質なものが生成されてしまったからだと考えられる．

enc の転移の影響　自己符号化の場合，表 2 から事前学習データに依らず enc の転移が性能に貢献していることが確認できる（5~8 行目）．表 3 から両事前学習データともに enc をフリーズしても何もフリーズしない場合と同程度の性能を出している．よって enc の学習は頻度情報のみで十分であり，主な翻訳性能の貢献に繋がると考えられる（貢献 4）．

MASS の場合，表 2 より事前学習データが “En” では enc が性能に貢献しているが，“Zipf” では他のコンポーネントの場合と同程度のスコアである（10~12 行目）．表 3 からは enc をフリーズした場合，他のコンポーネントをフリーズした場合よりも低いスコアが見られる．これは MASS で enc が学習した表現は NMT で必要とされる表現とは異なり，NMT での再学習が必要であることを示している．また事前学習データが “Zipf” の時は大幅な性能低下が見られるため，頻度情報のみでは enc の学習は上手く行われない．よって MASS では，enc は共起性や構造性を持つ実データでの事前学習が必要であり，下流タスクで enc を再学習する必要がある．

x-attn の転移の影響　自己符号化の場合，表 2 から事前学習データによらず x-attn のみを転移すると大幅に性能が下がるため，他のコンポーネント（特に enc）と組み合わせる必要がある，これはトークンの対応がそのトークン自身とで取られており，NMT で求められるトークンの対応とかけ離れた表現が学習されているためと考えられる．また，表 3 でフリーズしても比較的性能が下がらないのは，事前学習済み enc と組み合わせているためと考えられる．

MASS の場合，表 2 から “En” ではかなり低いスコアが見られる（3 行目）．これは英語からドイツ語へのトークンの対応が学習されていないためと考えられる．一方で “Zipf” だと大きなスコアの変動は見れられない．“Zipf” データはトークンが独立に抽出されるため，MASS での学習であまり偏りのないトークン同士の対応分布が獲得できたためと考えられる．表 3 でのフリーズの結果は自己符号化と同様に，事前学習済み enc と組み合わせているため比較的性能が下がらないと考えられる．

表 3　全てのコンポーネントを転移した後，各部分をフリーズした時の，事前学習法，事前学習データごとのテストセットにおける BLEU スコア．“×” は該当するコンポーネントをフリーズしていることを意味する．

	コンポーネント				AE		MASS	
行	emb	enc	x-attn	dec	En	Zipf	En	Zipf
1					8.3	8.8	8.8	8.0
2	×				5.4	7.4	7.1	6.0
3		×			8.2	8.6	6.4	2.5
4			×		8.0	8.4	7.4	6.6
5				×	7.0	7.2	7.6	4.5

dec の転移の影響　自己符号化の場合，表 2 より両事前学習データともに同程度のスコアを出している（4, 12 行目）．表 3 より dec をフリーズした場合，enc, x-attn に比べ性能が下がるが，両事前学習データで同程度のスコアである．よって dec でも頻度の学習が行われているが，下流タスクで再学習する必要がある．

MASS の場合は表 2 からは両事前学習データともに同程度のスコアである（4, 12 行目）．表 3 より enc の場合と同様に，“Zipf” で訓練したものは “En” よりも性能が低いため実データで学習する必要がある．

6　おわりに

本研究では実データ，人工データを自己符号化と MASS の 2 つの事前学習法で訓練し NMT での性能を調査した．自己符号化の場合は，トークンの頻度を考慮したデータセットで学習したモデルが実データでのものよりも高い性能であったが，MASS では実データで学習する必要があることを示した．

また転移学習に対する ablation study を行い，モデルの各コンポーネントが各事前学習データ，事前学習法により受ける翻訳性能への影響を調査した．自己符号化の場合，事前学習データの種類に関わらず embeddings 以外のコンポーネントが主な性能に貢献しており，encoders がトークンの頻度情報を学習するだけで高い性能が発揮されることを示した．一方 MASS では実データで学習することで embeddings が性能に大きく貢献することを示した．

今後の展望として，MASS でも人工データでの事前学習で高い性能を出せるような，共起性，構造性を考慮した人工データの作成を目指している．

参考文献

[1] Isabel Papadimitriou and Dan Jurafsky. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In **EMNLP**, 2020.

[2] Cheng-Han Chiang and Hung yi Lee. Pre-training a language model without human language. *arXiv preprint arXiv:2012.11995*, 2020.

[3] Cheng-Han Chiang and Hung yi Lee. On the transferability of pre-trained language models: A study from artificial datasets. *arXiv preprint arXiv:2109.03537*, 2021.

[4] Kundan Krishna, Jeffrey Bigham, and Zachary C. Lipton. Does pretraining for summarization require knowledge transfer? In **Findings of EMNLP**, 2021.

[5] Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. In neural machine translation, what does transfer learning transfer? In **ACL**, 2020.

[6] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation. In **EMNLP**, 2016.

[7] Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. An empirical study of language relatedness for transfer learning in neural machine translation. In **PACLIC**, 2017.

[8] Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? In **NAACL**, 2018.

[9] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MASS: Masked sequence to sequence pre-training for language generation. In **ICML**, 2019.

[10] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **ACL**, 2020.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **NeurIPS**, 2017.

[12] 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. In **ICLR**, 2019.

[13] Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In **WMT**, 2014.

[14] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In **ACL**, 2007.

[15] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In **ACL**, 2016.

[16] Matt Post. A call for clarity in reporting BLEU scores. In **WMT**, 2018.

A　ハイパーパラメータ

表 4 に本実験で設定した主なハイパーパラメータを示す．word-mask 率は MASS で事前学習を行う際，入力系列をどれだけマスクするかの割合である．

表 4　各事前学習法ごとの事前学習時 (PT) とファインチューニング時 (FT) のハイパーパラメータ．

		AE		MASS	
ハイパーパラメータ	**N/A**	**PT**	**FT**	**PT**	**FT**
学習率	5e-4	5e-4	5e-4	5e-5	5e-5
dropout 率	0.3	0.1	0.3	0.1	0.3
word-mask 率	N/A	N/A	N/A	0.5	N/A
バッチサイズ	4,096 × 8 トークン				
warmup ステップ数	4,000				
ビーム幅	4				