Vol.2017-NL-233 No.1
2017/10/24

ニューラル機械翻訳における埋め込み層の教師なし事前学習

根石 将人[1,a)]　佐久間 仁[1,b)]　遠田 哲史[1,c)]　石渡 祥之佑[1,d)]　吉永 直樹[2,e)]　豊田 正史[2,f)]

概要：大規模なニューラルネットワークの最適化では，広大な探索範囲とその非凸性により，得られる局所最適解の質とその収束速度は各パラメタの初期値に強く依存する．本研究では，Encoder-Decoder モデルを利用した機械翻訳において，より良い局所最適解への収束と学習の高速化を目的とし，モデルの中で単語を単語埋め込みに変換する埋め込み層の低コストな事前学習方法を提案する．Encoder-Decoder モデルの事前学習の対象としては，埋め込み層以外にも隠れ層や出力層が考えられるが，埋め込み層には単純なニューラル言語モデルを用いて教師無しで高速に学習可能であるという利点が存在する．そこで本論文で提案する方法では，既存の言語モデルにより翻訳タスクの対訳コーパスのみで低コストに事前学習した単語埋め込みを用いて Encoder-Decoder モデルの埋め込み層の初期化を行う．実験では，ASPEC 英日翻訳タスクの評価データを用いて，事前学習する単語埋め込みの学習データの種類（一般ドメイン，翻訳タスクの学習データ），学習手法（CBOW, Skip-gram, SI-Skip-gram, GloVe），初期化する対象の埋め込み層（Encoder, Decoder），初期化後の更新の有無などを変え，モデルの学習速度と翻訳性能の観点で初期化の効果を検証する．

1. はじめに

機械翻訳の分野では，ニューラルネットワークを用いたニューラル機械翻訳（NMT）が，単純なモデル構造と翻訳性能の高さから非常に注目を集めている．中でも Encoder-Decoder モデル [3] や Sequence-to-Sequence モデル [16] は，後に提案された Attention 機構 [2] と合わせて，旧来の句に基づく統計的機械翻訳を超える翻訳性能を達成するに至った．

NMT ではニューラルネットワーク自体の構造だけでなく，ネットワーク内の各パラメタの事前学習 [14] や最適化手法 [1]，バッチの構成 [8] など，周辺的な変更がモデルの性能に大きく影響することが知られている [4]．その中でも事前学習は工夫の余地が大きく，影響も大きいことが最近 Ramachandran らにより指摘されている [14]．彼らの手法では Encoder-Decoder モデルを対象として，そのネットワーク構造に類似する言語モデルを用いて各層の事前学習を行うが，英独機械翻訳タスクにおいて BLEU スコア 2.7 ポイントの向上と高い効果を上げる一方で，大規模な単言語コーパスと多大な計算コストを要する．

そこで本論文では，NMT モデルにおける埋め込み層のみを，翻訳タスクの対訳コーパスだけで事前学習した単語埋め込みを用いて初期化する方法を提案する．初期化対象を埋め込み層に限ることで，事前学習に高速な教師なし学習を用いることを可能とし，かつ外部コーパスを使用しないため，非常に低コストな方法である．また，埋め込み層を有するあらゆるニューラルネットワークモデルへの適用が可能であり，一度事前学習したパラメタの値をそのまま異なるタスクのモデルへ応用することも考えられる．本論文では事前学習する単語埋め込みの学習データの種類や，学習手法，また初期化する対象の埋め込み層や初期化後の更新の有無などについての比較検討も行う．

日英対訳コーパスを用いた翻訳実験では，従来のランダム初期化と比較し，CBOW を用いた事前学習による初期化により BLEU スコア 1.79 ポイントの向上を達成し，同時に学習の高速化も実現した．

以降の本論文の構成は次の通りである．続く 2 節では関連研究について述べる．3 節で提案した事前学習方法を基礎技術と合わせて説明する．4 節において実験と結果を述べ，その考察を行う．最後に 5 節で本論文のまとめを行う．

2. 関連研究

本節では，本論文の手法である事前学習についての説明をし，本研究と関連する 3 つの研究について詳細を述べる，

[1] 東京大学大学院情報理工学系研究科
[2] 東京大学生産技術研究所
[a)] neishi@tkl.iis.u-tokyo.ac.jp
[b)] jsakuma@tkl.iis.u-tokyo.ac.jp
[c)] tohda@tkl.iis.u-tokyo.ac.jp
[d)] ishiwatari@tkl.iis.u-tokyo.ac.jp
[e)] ynaga@iis.u-tokyo.ac.jp
[f)] toyoda@iis.u-tokyo.ac.jp

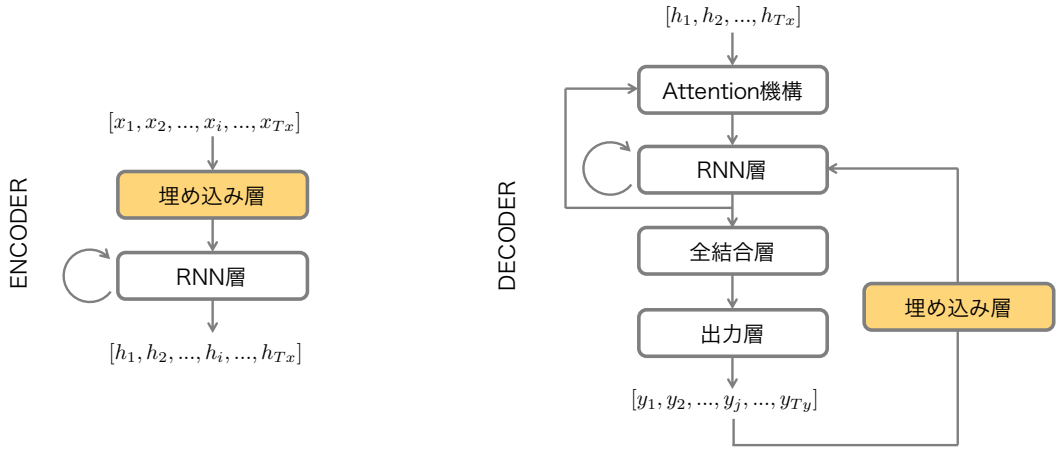

図 1 Attention 機構付き Encoder-Decoder モデルの概形

最後にそれらの研究と本論文の手法との違いを述べる．

事前学習は，ネットワークの各層のパラメタの値を目標タスクのデータ以外のデータで学習することで，ニューラルネットワークの性能を上げる手法である．この手法では，目的タスクに対して十分な量の教師データが用意できない場合に，まず同様のタスクでドメインが異なる十分な量のデータセットを用いて汎化的な学習を行う．そしてその後に少量の教師データを用いて目的タスクに特化するよう調整を行う．

また，これらの手法は，パラメタの最適解探索という観点では次の様に捉えることもできる．一般に大規模なニューラルネットワークの最適化は，その非凸性により大域的最適解にたどり着くことは難しく，探索空間が広大であるため短時間での収束は難しい．そこで，各パラメタの初期値を比較的良い最適解の近くに設定することで，最適解の探索を容易とし，収束解の質と学習速度の 2 つを向上させることが期待できる．

NMT に限らず，言語を扱う Encoder-Decoder モデルのパラメタの事前学習の研究は，言語モデルの自然な文を生成する能力を NMT モデルに取り入れることを目的とする．Gulcehre らは，事前学習ではないが，対訳コーパスで Attention 機構付き Encoder-Decoder モデルを，大規模単言語コーパスで言語モデルをそれぞれ別に学習し，それらを統合する 2 通りのモデルを提案した [5]．

Venugopalan らは，機械翻訳ではなく動画からのキャプション生成タスクを対象としているが，Gulcehre ら [5] を踏まえた異なる統合モデルに加え，Attention 機構付き Encoder-Decoder モデルの Decoder の一部である埋め込み層と RNN 層を，大規模単言語コーパスを用いて事前学習した言語モデルのパラメタを用いて初期化する手法を提案した [17]．また，さらに GloVe[13] を用いて大規模単言語コーパスから単語埋め込みを学習し，Decoder の埋め込み層の初期化や，モデルの損失関数に利用する手法も提案した．

Ramachandran らは原言語と目的言語の 2 つの言語モデルを大規模な単言語コーパスでそれぞれ事前学習し，その埋め込み層と RNN 層の一部および出力層を用いて，Attention 機構付きの Encoder-Decoder モデルの Encoder と Decoder の対応する各層を初期化する手法を提案した [14]．これにより BLEU スコアで 2.7 ポイント（En-De）の向上を達成したが，過学習の防止の為に，事前学習で初期化した層に翻訳タスクの学習と元の言語モデルの学習のマルチタスク学習を行っており，全体のモデルが非常に複雑になっている．

これら言語モデルを取り入れる手法に対し，本論文の方法はパラメタの最適解探索の促進を目的とした．事前学習で初期化する範囲と使用するコーパスを制限し，教師なしの高速な学習手法を用いることで，先行研究と比較して格段に低コストな方法でありながら，日英翻訳実験において BLEU スコア 1.79 ポイントの向上を達成した．

3. Attention 機構付き Encoder-Decoder モデルにおける埋め込み層の事前学習

本研究では，Attention 機構付きの Encoder-Decoder モデルにおける埋め込み層の事前学習による初期化を提案する．本節では，まず Attention 機構 [2] 付きの Encoder-Decoder モデル [16] の基本的な説明をし，その後に本論文で提案する埋め込み層の教師なし事前学習手法について説明する．

3.1 Attention 機構付き Encoder-Decoder モデル

Sutskever らによって提案された Encoder-Decoder モデル [16] は，Encoder と Decoder と呼ばれる 2 つの再帰型ニューラルネットワーク（RNN）から構成される．Encoder は原言語の入力文を中間表現である固定長の連続値ベクトルの集合へとエンコードし，Decoder はそのベクトルの集合を目標言語の出力文にデコードするという二段階の仕組みになっている．Bahdanau らは Encoder-Decoder モデル

の Decoder に Attention 機構と呼ぶ機構を組み込み，ベクトルの集合から目標言語の文をデコードする際に，原言語文の各単語の寄与分を考慮するよう改良した [2]．全体の構造を図 1 に示す．

Encoder は，まず入力文を単語（またはサブワード [15]）に分割した単語列 $[x_1, x_2, ..., x_{Tx}]$ を入力とし，各単語を埋め込み層と呼ばれる層においてそれぞれ固定長のベクトル表現である単語埋め込みに変換する．そして得られた単語埋め込みを語順に従って一つずつ RNN 層に入力することで，前後関係を考慮したベクトル列 $[h_1, h_2, ..., h_{Tx}]$ へとエンコードする．

Decoder は，Encoder の出力 $[h_1, h_2, ..., h_{Tx}]$ を入力とし，ステップ毎に 1 つずつ単語をデコードする．Attention 機構は Encoder で生成されたベクトル列に対する重み付けを，自身の状態と Decoder の前状態に基づいて行う．続く RNN 層では，埋め込み層から受け取る直前の出力単語 y_{j-1} の情報も用いて，入力のベクトル列の重み付き線形和を出力単語の情報を持つベクトルへと変換する．全結合層でこのベクトルを語彙数次元のベクトルに写像した後，出力層においてソフトマックス関数を用いて各単語の生成確率へとデコードする．出力された単語の情報を RNN 層に返すために，Decoder 内の埋め込み層が出力単語 y_j を連続値ベクトルに変換した後，RNN 層へと入力を行う．最終的な出力の単語列 $[y_1, y_2, ..., y_{Ty}]$ は，貪欲法やビーム探索などのアルゴリズムにより各単語の生成確率に基づいて決定する．

3.2 埋め込み層の教師なし事前学習

3.1 節で説明した Attention 機構付きの Encoder-Decoder モデルの各層のパラメタは，通常はランダムに，あるいは正規分布を用いて初期化されるが，我々は Encoder と Decoder の埋め込み層を事前学習した単語埋め込みにより初期化することを試みる．先行研究 [17] でも用いられているように，単語埋め込みには GloVe などの教師無しで高速に大規模生コーパスから学習する手法が存在するため，埋め込み層の初期値については極めて低コストで得ることが可能である．また，埋め込み層はタスクに依らず，単語を連続値ベクトルとして扱う全てのモデルに存在するため，この初期化は機械翻訳以外のタスクに対しても適用可能である．

単語埋め込みの事前学習には，理想的には翻訳対象と同一ドメインの大規模データがあることが望ましいが，そのようなコーパスは必ずしも容易に利用できるとは限らないため，翻訳タスクの対訳コーパスのみを用いた場合の効果を検証する．対訳コーパスの規模にもよるが，1GB 程度のテキストデータであれば，高速な単語埋め込み学習手法と合わせて，CPU （Intel Xeon CPU E5-2680 v4 @ 2.40GHz）のみを用いて 10 分程度で事前学習が終わり，非常に低コストに適用することができる．

	学習データ		開発データ		評価データ	
	英	日	英	日	英	日
文数	1,783,817		1790		1812	
平均トークン数	31.08	33.13	31.06	34.58	30.69	34.03

表 1 前処理後のデータセット

4. 評価実験

本節では，提案した事前学習方法の評価のための実験を行う．まず 4.1 節で実験設定について説明をする，4.2 節では，提案する事前学習による初期化とランダム初期化との比較に加えて，事前学習して得られた単語埋め込みを用いて様々な初期化を試みる実験を行う．4.3 節では実験結果から考察を行う．

4.1 実験設定

本論文では ASPEC (Asian Scientific Paper Excerpt Corpus)[9] 日英対訳コーパスを用い，英日翻訳タスクでの実験を行なった．本節ではまずコーパスの前処理について言及した後，実際の学習に使用した Encoder-Decoder モデル，埋め込み層の初期化，翻訳結果の評価方法の各設定について説明をする．以降の実験では，特に言及がないものについては，以降に続く 4 節内で述べる基本設定をそのまま使用する．なお，この基本設定は WAT 2017 参加システム[*1][10] において開発データを用いてチューニングしたものである．

4.1.1 前処理

前処理は基本的に WAT 2017 で推奨されている前処理[*2]に従い行なった．英語については Moses[*3] (ver. 2.2.1)[7] のスクリプトを用いてトークン化及び Truecasing を行い，日本語については KyTea[*4] (ver. 0.4.2)[11] を用いて単語分割を行なった．また対訳コーパスの学習データについては文の単語数の上限を 50 とし，それ以上のデータは除外した．

この基本的な前処理に加え，未知語問題の軽減のために本実験ではさらに SentencePiece[*5]を用いてトークン化を行った．これは文を文字列として扱い，部分文字列の頻度に基づきトークン境界を学習した後，切り分ける処理である．この処理では，単語分割された文も半角スペースを特殊記号に置き換えた上で純粋な文字列として扱い，再度トークン化を行う．SentencePiece の設定として，トークン境界を決める言語モデルには初期設定値のユニグラ

[*1] https://github.com/nem6ishi/wat17
[*2] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2017/baseline/dataPreparationJE.html
[*3] http://www.statmt.org/moses/
[*4] http://www.phontron.com/kytea/
[*5] https://github.com/google/sentencepiece

ムを用い，ASPEC の学習データを使って学習した．また ASPEC データセットの特性上，数字などの日英に共通のトークンがあることから，両言語を混ぜた学習データを用いて語彙数を 16000 個に設定した．これらの設定で学習したモデルを用い、データセット全てのトークン化を行なった．以上の前処理によって得られたデータセットの統計量を表 1 に示す．

4.2 節では，初期化に用いる単語埋め込みの学習に，ASPEC の学習データに加えて Wikipedia コーパスを追加する実験を行う．Wikipedia コーパスは，日英共に 2017 年 09 月 01 日の本文全文の dump データを使用した．まず WikiExtractor[*6]を用いて本文を抽出した後，不要なタグを除去したものに対して，ASPEC と同様の前処理を行った．SentencePiece によるトークン化についても，ASPEC の学習データで学習したモデルを用いて行った．

4.1.2 Encoder-Decoder モデル

本論文の Encoder-Decoder モデルの実装は，Google によるオープンソース実装[*7]を元に改良を加えたものを使用した．RNN 層としては，Encoder には 2 層の双方向 LSTM を，Decoder には 4 層の LSTM を用い，共にドロップアウト率は 0.8 とした．隠れ層の次元数は全て 512 に統一し，学習の最適化には初期学習率を 0.0001（元実装の初期設定値は 0.001）とした Adam [6] を用い，ミニバッチ学習におけるバッチサイズは 256 とした．

4.1.3 単語埋め込みの学習

埋め込み層の初期値とする単語埋め込みの学習には，word2vec (ver. 1.0)[*8] に実装されている CBOW，Skip-gram，fastText (ver. 1.0)[*9]に実装されている Subword Information Skip-gram (SI-Skip-gram)，そして GloVe (ver. 1.2)[*10] を様々な窓幅で比較し，最も有効であった窓幅を 5 とした CBOW を基本設定とした．これらの学習手法の比較については 4.2.1 節で詳しく述べる．

学習データは当該タスクコーパスである ASPEC の学習データのみに限り，SentencePiece によって語彙は共有化されているため，この学習においても両言語のデータを連結したデータを用いた．単語埋め込みの次元数は Encoder-Decoder モデルの隠れ層の次元数に合わせ 512 とし，それ以外のパラメタは各実装の初期設定値に倣った．

本実験のモデルでは通常の語彙に加えて，“SEQUENCE_START”（文頭），“SEQUENCE_END”（文末），“UNK”（未知語）の 3 つの特殊トークンが使用される．そこで，単語埋め込みの学習でも “SEQUENCE_START”，“SEQUENCE_END” については学習の前に学習データの各文の先頭と最後に加えることで通常の語彙と同様に学習を行った．“UNK” については，単語埋め込みの学習が終わった段階で，16000 個の語彙に含まれない全ての単語の単語埋め込みの平均値を使用した．

[*6] https://github.com/attardi/wikiextractor
[*7] https://google.github.io/seq2seq/
[*8] https://github.com/svn2github/word2vec
[*9] https://github.com/facebookresearch/fastText
[*10] https://github.com/stanfordnlp/GloVe

初期化方法	最高スコア	ステップ
ランダム初期化	33.71	160,000
事前学習による初期化（ASPEC のみ）	**35.50**	138,000
事前学習による初期化（ASPEC + Wikipedia コーパス）	34.68	180,000

表 2　埋め込み層初期化実験の各最高スコアとそのステップ数

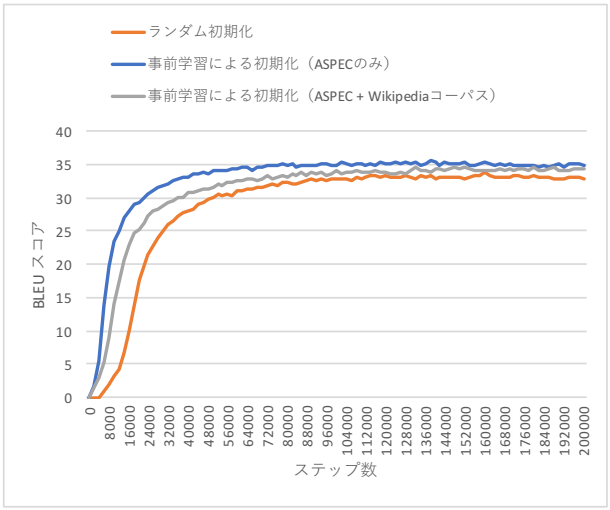

図 2　埋め込み層初期化実験の学習曲線

4.1.4 評価方法

翻訳結果の評価には BLEU [12] を用いた．本実験では全て，モデルの選定のために使用する開発データを用いて評価を行なった．実装の仕様により，出力のトークン列を連結し，特殊記号に置き換えた半角スペースを元に戻したものを翻訳結果として出力する．これを一度半角スペースを取り除き文字列に戻した上で，再度 KyTea を用いてトークン化を行い，BLEU のスコアを算出した．評価を高速に行うため，デコードには全てビーム探索ではなく貪欲法を用いた．ただし，予備実験によりデコードのアルゴリズムを変更してもモデルの優劣は入れ替わらないことを確認している．モデルは更新ステップ数 2000 毎に評価を行い，最も高いスコアをそのモデルのスコアとした．

4.2 実験結果

実験では，本論文で提案する事前学習による初期化と従来のランダム初期化との比較に加えて，事前学習して得られた単語埋め込みを用いて様々な初期化を試み，その効果について詳しく考察を行う．具体的には，ランダム初期化との比較による有用性の検証実験の後に，初期化に用いる単語埋め込みの学習手法の比較実験（4.2.1 節），事前学習による初期化をした場合の Encoder と Decoder の埋め込

初期化手法	窓幅	BLEU スコア	スコア差
ランダム初期化	-	33.71	0
CBOW	2	34.97	+1.26
	5	**35.50**	**+1.79**
	10	35.25	+1.54
Skip-gram	2	34.17	+0.46
	5	34.44	+0.73
	10	34.38	+0.67
SI-Skip-gram	2	34.04	+0.33
	5	34.44	+0.73
	10	34.33	+0.62
GloVe	2	34.50	+0.69
	5	34.58	+0.77
	10	33.98	+0.27
	15	34.35	+0.64
学習済みモデル（ランダム初期化）	-	33.81	+0.10
学習済みモデル（CBOW）	(5)	35.14	+1.43

表 3 初期化に用いる単語埋め込みの学習手法による翻訳性能の違い

み層の影響の違いを比較する実験（4.2.2 節），初期化した埋め込み層を学習させずに固定する実験（4.2.3 節），学習率の影響を確認する実験（4.2.4 節）を行った．本節では実験結果についてのみ報告し，詳細な考察は 4.3 節で行う．

まず埋め込み層の初期化について，(1) ランダム初期化，(2)ASPEC の学習データのみを用いて CBOW で事前学習した単語埋め込みによる初期化，さらに (3) 事前学習に用いるコーパスとして，ASPEC に加えて Wikipedia も追加した場合の計 3 通りについて実験を行なった．

表 2 に実験結果を，図 2 に学習曲線をそれぞれ示す．この結果より，事前学習による初期化は従来のランダム初期化に対して翻訳性能と学習速度の両方が向上していることが確認できる．特に ASPEC のみを用いた事前学習による初期化では，ランダム初期化に比べて BLEU スコアで 1.79 ポイントと大幅な改善が得られた．また学習曲線から，事前学習による初期化のどちらの場合でも，ランダム初期化に比べて立ち上がりが早く，全体の学習が高速化されていることがわかる．ただし，Wikipedia コーパスを加えた場合はスコアは向上しているものの，収束までのステップ数も増大している．

4.2.1 単語埋め込みの学習手法の影響

前節では窓幅を 5 とした CBOW を基本設定としたが，本節では窓幅及び単語埋め込みの学習手法の比較実験を行う．比較する単語埋め込み学習手法には CBOW，Skip-gram，SI-Skip-gram，GloVe を用いた．窓幅については 2，5，10 の 3 種類に加え，GloVe のみに対しては GloVe の初期設定である 15 の場合の実験も行なった．

また，異なるタスクでの事前学習と同一タスクでの事前学習を比較するために，既に一度学習済みのモデルから埋め込み層だけを抜き出し，これを初期値として設定する場合の実験も行った．学習済みの埋め込み層を取り出すモデルとしては，ランダム初期化での学習済みモデルと，基本設定とした窓幅 5 の CBOW での学習済みモデルの 2 つを扱った．

	初期化方法			
	Encoder	**Decoder**	最高スコア	ステップ
(1)	ランダム	ランダム	33.71	160,000
(2)	CBOW	ランダム	34.93	172,000
(3)	ランダム	CBOW	34.07	154,000
(4)	CBOW	CBOW	35.50	138,000

表 4 事前学習による初期化の対象を変化させた場合の各最高スコアとそのステップ数

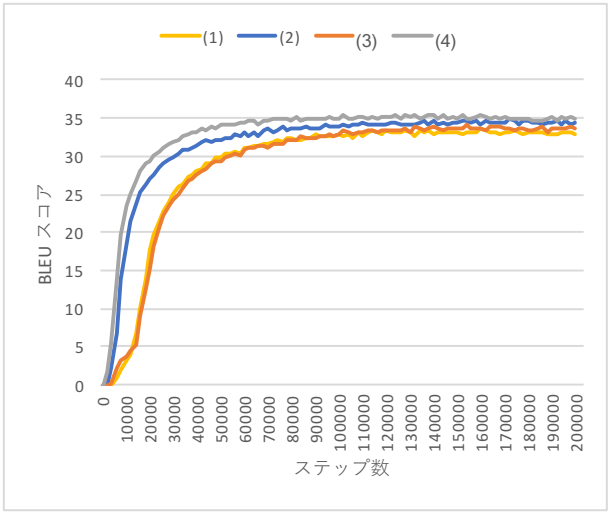

図 3 事前学習による初期化の対象を変化させた場合の学習曲線（各番号は表 4 と対応する．）

実験の結果を表 3 に示す．どの手法でも基本的にはランダム初期化に勝るという結果になった．その中でも唯一 CBOW だけは従来手法からの BLEU スコアの上がり幅が 1 ポイント以上に達し，事前学習に最も適した単語埋め込み学習手法であることが確認できる．また窓幅については，全ての手法に渡って中間値である 5 が最も良い結果となった．なお，この結果により次節以降の実験では全て窓幅を 5 とした CBOW を用いている．

翻訳タスクでの学習済みの埋め込み層の値を初期値として用いたモデルでは，元の学習済みモデルと比較して，ランダム初期化はわずかに BLEU スコアが上がり，CBOW はわずかに下がる結果となった．どちらにおいても，元の学習済みモデルからの大きな向上は認められなかった．

4.2.2 Encoder と Decoder の初期化による影響の違い

これまでの実験では常に Encoder と Decoder の両方の埋め込み層に事前学習を適用していたが，この実験では手法の適用範囲を変化させた場合のモデルを学習する．これにより Encoder と Decoder，それぞれの埋め込み層の翻訳

固定する埋め込み層	最高スコア	ステップ
両方	35.32	152,000
Encoder 側のみ	34.98	148,000
Decoder 側のみ	35.24	122,000
固定なし	35.50	138,000

表 5 初期化した埋め込み層を学習させず固定した場合の各最高スコアとそのステップ数

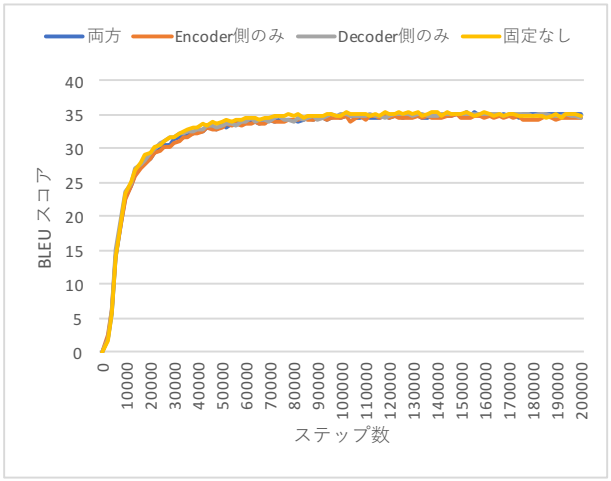

図 4 初期化した埋め込み層を学習させず固定した場合の学習曲線

性能への影響の違いを比較する．

表 4 に実験結果を，図 3 に学習曲線をそれぞれ示す．学習曲線の図から，Encoder 側のみの場合は両方を事前学習した場合に近く，Decoder 側のみの場合は従来のランダム初期化に近い結果を示すことがわかる．最高 BLEU スコアの点でも Encoder 側のみの場合は Decoder 側のみの場合に比べて 0.86 ポイント高い結果となった．

4.2.3 埋め込み層の固定

これまでの実験では，事前学習した各単語埋め込みのパラメタを埋め込み層の初期値として扱い，初期化後には Encoder-Decoder モデルの他の層と同様に学習をし，最適化を行った．しかしながら，事前学習した単語埋め込みのパラメタがその時点で翻訳タスクでの単語埋め込みとして十分に有用であった場合，それ以上の学習は必ずしも必要でないと考えられる．この仮説を検証するため，直前の 4.2.2 節の実験のように，初期化を Encoder の埋め込み層，Decoder の埋め込み層，両方と 3 つの場合についてそれぞれの値を固定したまま学習を行い，その比較を行った．

表 5 に実験結果を，図 4 に学習曲線をそれぞれ示す．最高 BLEU スコアだけを見ると，固定しない場合が僅かに良い結果となっているが，一方で学習曲線にはほとんど差異は見られなかった．

4.2.4 学習率の影響

最後に学習率を変化させる実験を行う．4.2.3 節で述べたように，もしも事前学習した値が単語埋め込みとして既に十分に有用であった場合は，それ以上の学習は不要である．4.2.3 節ではこれを埋め込み層のみを固定することで検証したが，一方で学習率を十分に小さくすることでも事前学習の値をあまり変えずにモデルの学習を行うことができる．そこでこの実験では学習率を 0.01 から 0.00001 まで変化させて，その影響を確かめた．

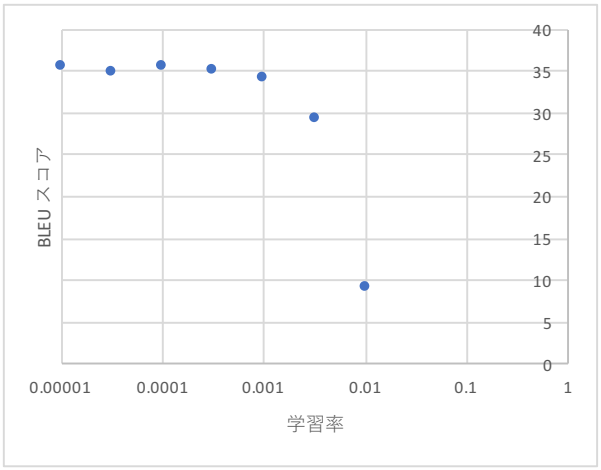

図 5 事前学習した単語埋め込みで初期化した場合の学習率と BLEU スコアの関係

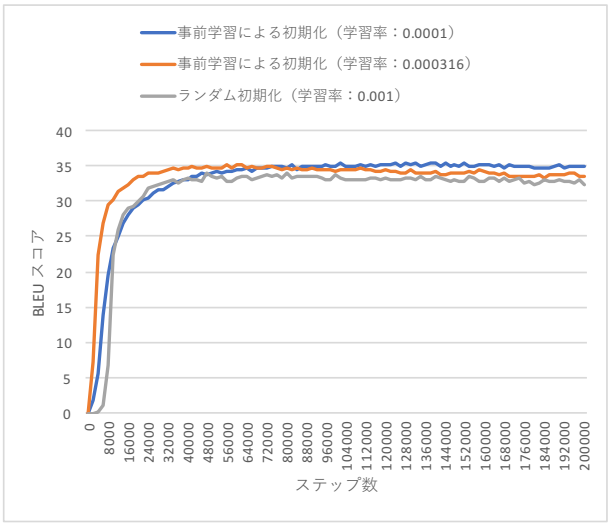

図 6 学習率と学習曲線

結果を図 5 に示す．学習率が 0.000316 から BLEU スコアの収束が見られるため，本実験の実装では学習率をそれ以下にすることが妥当である．改良を加えた元実装の学習率の初期設定値が 0.001 であることを考えると，元実装に比べて適当な学習率が小さくなっている．これはすなわち，事前学習による初期化では学習が遅くなることを意味する．

そこで追加実験として，学習率を 0.0001 と 0.000316 とした事前学習による初期化と，学習率を 0.001 としたランダム初期化との比較を行った．学習曲線を図 6 に示す．学習率を 0.000316 とした事前学習による初期化は全てのステップにおいてランダム初期化の BLEU スコアを常に上

回った．学習率を 0.0001 とした場合でも，ランダム初期化と学習率に 10 倍の違いがあるにも関わらず，最初期の立ち上がりでは事前学習による初期化が優れている．またランダム初期化が 48,000 ステップで最高 BLEU スコアに達しているのに対して，学習率を 0.0001 とした事前学習による初期化は直後の 52,000 ステップでそのスコアを追い越した．

4.3 考察

4.2 節最初の実験では教師なし学習による埋め込み層の初期化が翻訳性能と学習速度の向上に貢献することが示された．ASPEC の学習データに Wikipedia コーパスを追加した場合は，ランダム初期化は上回ったものの，ASPEC の学習データのみの場合に比べて翻訳性能でも学習速度でも劣る結果となった．この原因としては，異なるドメインのデータによる影響や，SentencePiece によるトークン化の影響などが考えられる．特に後者については，ASPEC の学習データを用いて学習したモデルを使用して分割したため，Wikipedia コーパスのトークン化としては不適当であった可能性が考えられる．

4.2.1 節では，今回比較した中では窓幅を 5 とした CBOW が初期化用の単語埋め込みの学習手法として最適であることが示された．窓枠内の文脈単語と目的単語の一対一の狭い関係を扱う Skip-gram と SI-Skip-gram や，広くコーパス全体の単語のバランスを重視する GloVe に対して，CBOW は窓幅内の文脈単語の集合と目的単語を扱う手法である．トークン毎に翻訳文を生成する Encoder-Decoder モデルでは，CBOW の適度な範囲での単語（トークン）間関係を扱う手法が適していると考えられる．窓幅についても，全ての手法に渡って中間値である 5 が最も良い結果になっており，これについても適度な範囲が有効であることが示唆されている．

また同じく 4.2.1 節では，同一タスクでの学習済み埋め込み層を用いた初期化はほとんど BLEU スコアに影響しないことが示された．これにより，学習済みであるかどうかに比べ，単語埋め込みの学習手法自体が BLEU スコアの上昇に影響していることがわかった．同じ機械翻訳タスクの事前学習よりも別タスクである CBOW での事前学習が優れていると言う結果は非常に興味深い．

4.2.2 節では，教師なし学習による埋め込み層の初期化において，Decoder 側の埋め込み層よりも Encoder 側の埋め込み層が翻訳性能と学習速度の両方の向上に影響していることが示された．ニューラルネットワークの各層は前の層の出力を入力として受け取るため，前の層の学習が不十分である場合は不適切な値を入力として受け取ることになる．そのため，ネットワークのほぼ最後尾である Decoder 側の埋め込み層は，ネットワークの入力部である Encoder 側の埋め込み層と比較して不適切な学習をしやすいと言える．事前学習したパラメタの値が不適切に更新され，事前学習の意味が失われると考えると妥当な結果である．

4.2.3 節では，事前学習による埋め込み層の初期化後の学習の有無が，モデル全体の学習にほとんど影響しないことが示された．埋め込み層以外の層はランダムに初期化されており，学習において揺らぎがあることを考えると，全ての場合に渡ってほぼ差はないと考えられる．これは言い換えると，CBOW で学習した単語埋め込みがその時点で機械翻訳タスクでも有効であると言うことである．これは非常に興味深い結果である．CBOW と機械翻訳は全く異なるタスクであるので，CBOW の学習結果が機械翻訳でも使えるのであれば，タスク横断的な単語の埋め込み表現の存在可能性が考えられる．機械翻訳タスクの結果が CBOW において有用であるかの検証を始め，様々に研究されているタスク特化型の単語埋め込みの横断的な比較検討を今後進めたい．

4.2.4 節では，事前学習による埋め込み層の初期化は，学習率を小さくすることによる学習の低速化はあるものの，それを上回る学習の高速化を実現することが示された．この結果により，モデルの一部でしかない埋め込み層の初期化が，モデル全体の学習を高速化していることがわかる．また，4.2.2 節の結果と照らし合わせると，ニューラルネットワークにおける学習の高速化には入力に近い Encoder 側の埋め込み層の事前学習が非常に効果的であることがわかる．

以上の結果を以下にまとめる．

- 教師なしの事前学習による埋め込み層の初期化は，従来のランダム初期化に比べて，翻訳性能の向上と学習の高速化の点で効果的である．
- 今回比較した中では，初期化用の単語埋め込みの学習手法は，窓幅を 5 とした CBOW が最も適している．
- 初期化に用いるパラメタは，学習済みであるかどうかよりも，単語埋め込みの学習手法が BLEU スコアの上昇に影響する．
- 事前学習による埋め込み層の初期化では，2 つある埋め込み層のうち，Encoder 側の埋め込み層の方が影響力が大きい．
- 事前学習した単語埋め込みは，その時点で翻訳タスクにおいて十分有用であり，学習の必要性は少ない．
- 事前学習による埋め込み層の初期化を適用する場合は，使わない場合に比べて学習率をやや小さく設定する必要があるが，学習の高速化は達成される．

5. おわりに

本論文では，言語モデルにより翻訳タスクの対訳コーパスのみを用いて教師なしで事前学習した単語埋め込みを用いた，ニューラル機械翻訳の埋め込み層の初期化方法を提案した．埋め込み層を初期化対象としており，また外部

コーパスを使用せず，高速な単語埋め込みの学習手法を用いるため，機械翻訳以外のタスクにも容易に適用することが可能である．評価実験の結果，教師なし学習による埋め込み層の初期化が翻訳性能の向上と学習の高速化の点で有用であるということだけでなく，機械翻訳のニューラルネットワークにおける各埋め込み層の影響力の違いや，タスク横断的な単語の埋め込み表現の可能性が示された．

今後の課題としては，提案した事前学習方法を異なる構造のニューラル機械翻訳に用いることを始めとし，対話などの機械翻訳以外のタスクへの応用，検証が考えられる．

謝辞 本研究は JSPS 科研費 JP16K16109, JP16H02905 の助成を受けたものです．

参考文献

[1] Bahar, P., Alkhouli, T., Peter, J.-T., Brix, C. J.-S. and Ney, H.: Empirical Investigation of Optimization Algorithms in Neural Machine Translation, *The Prague Bulletin of Mathematical Linguistics*, Vol. 108, No. 1, pp. 13–25 (2017).

[2] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *Proceedings of the Third International Conference on Learning Representations (ICLR)* (2015).

[3] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734 (2014).

[4] Denkowski, M. and Neubig, G.: Stronger Baselines for Trustable Results in Neural Machine Translation, *Proceedings of the First Workshop on Neural Machine Translation*, Association for Computational Linguistics, pp. 18–27 (2017).

[5] Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H. and Bengio, Y.: On using monolingual corpora in neural machine translation, *arXiv preprint arXiv:1503.03535* (2015).

[6] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *Proceedings of the third International Conference on Learning Representations (ICLR)* (2015).

[7] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), Demo and Poster Sessions*, pp. 177–180 (2007).

[8] Morishita, M., Oda, Y., Neubig, G., Yoshino, K., Sudoh, K. and Nakamura, S.: An Empirical Study of Mini-Batch Creation Strategies for Neural Machine Translation, pp. 61–68 (2017).

[9] Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S. and Isahara, H.: ASPEC: Asian Scientific Paper Excerpt Corpus, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pp. 2204–2208 (2016).

[10] Neishi, M., Sakuma, J., Tohda, S., Ishiwatari, S., Yoshinaga, N. and Toyoda, M.: A Bag of Useful Tricks for Practical Neural Machine Translation: Embedding Layer Initialization and Large Batch Size, *Proceedings of the 4rd Workshop on Asian Translation (WAT2017)* (2017 (to appear)).

[11] Neubig, G., Nakata, Y. and Mori, S.: Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), Short Papers*, pp. 529–533 (2011).

[12] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation, *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318 (2002).

[13] Pennington, J., Socher, R. and Manning, C. D.: GloVe: Global Vectors for Word Representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014).

[14] Ramachandran, P., Liu, P. and Le, Q.: Unsupervised Pretraining for Sequence to Sequence Learning, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 383–391 (2017).

[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 (ACL)*, pp. 1715–1725 (2016).

[16] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems (NIPS) 27*, pp. 3104–3112 (2014).

[17] Venugopalan, S., Hendricks, L. A., Mooney, R. and Saenko, K.: Improving LSTM-based Video Description with Linguistic Knowledge Mined from Text, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1961–1966 (2016).