英日翻訳タスクにおけるスワップモデルを通した seq2seq と Transformer の比較

根石 将人
東京大学大学院 情報理工学系研究科
neishi@tkl.iis.u-tokyo.ac.jp

吉永 直樹
東京大学 生産技術研究所
ynaga@iis.u-tokyo.ac.jp

1 はじめに

複雑な計算グラフの形で定義される深層学習モデルは多様であり，それらのモデルが扱える問題は集合的に包含関係になく，個々のモデルの観察のみでは適用タスクにおいてどのような問題が解けているか不明瞭である． 深層学習に基づく手法 (Neural Machine Translation; NMT) が主流となった 機械翻訳においても，異なる構造のモデルが多数提案されている [1, 2, 3]. NMT モデルの性能はテストセットにおける BLEU [4] 等の指標で定量的に評価されることが一般的に行われる．これに加え，モデルの構造の差異がもたらす影響についても定量的に評価・分析することは，今後のモデル改善における重要な課題である．

本稿では，NMT 初期から広く用いられている RNN に基づく NMT モデル（以降 seq2seq）[1] と，現在主流となりつつある自己注意機構に基づく NMT モデル (Transformer) [3] という 2 種のエンコーダ・デコーダモデルに注目し，これらの構造の差異による影響を明らかにする．具体的には，i) 各モデルの翻訳結果の類似度，また，Koehn ら [5] の統計的機械翻訳と NMT との比較研究を参考にして，ii) 学習データ量と精度の関係，iii) 翻訳原文の文長と精度の関係，iv) ドメイン外データセットでの精度を，実験により評価する．そのための方法論として，seq2seq と Transformer のエンコーダ及びデコーダを互いに交換する 2 つのスワップモデルを提案し，合計 4 種のモデルに対して翻訳タスクを通した比較分析を行う．

具体的に，ASPEC [6] を用いた英日翻訳タスクを通して得られた知見は以下の通りである．

- エンコーダとデコーダの組み合わせにより，モデルの機能（得意とする問題）は変わる．
- Transformer のエンコーダ・デコーダの影響力は強く，2 つのスワップモデルは共に，翻訳結果及び学習データ量に対する翻訳性能において，Seq2seq よりも Transformer に類似する．
- 学習データ量を減らしたとき，seq2seq は他の 3 つのモデルに比べ翻訳精度に顕著に劣る．
- 長文の翻訳では seq2seq が優れており，その理由として RNN エンコーダの貢献が大きい．

2 関連研究

NMT の登場以来，従来主流であった統計的機械翻訳 (Statistical Machine Translation; SMT) は長らく NMT との翻訳精度の比較対象となっている．特に NMT と SMT の違いを分析した研究としては，Bentivogli ら [7]，Toral ら [8]，Koehn ら [5] の研究が挙げられる．これらによって，大規模対訳データがある状況下では SMT に対する NMT の優位性が決定的なものと示されており，現在は NMT の研究が主流となっている．

NMT モデルの構造の多様化も進んでおり，RNN の置き換えによる高速化を狙った ConvS2S [2] や Transformer [3]，また，これまでの NMT を複合したモデル (RNMT+) [9] などが提案されているが，モデル間の比較は標準的な評価指標 (BLEU スコア) に基づく数値評価に留まっており，精度の向上幅も減少傾向にある．

そのような状況下で，Ding ら [10] は，seq2seq を対象として，可視化を通して NMT モデルの解釈を行っており，また，Lakew ら [11] は seq2seq と Transformer の 2 つのモデルについて，多言語翻訳タスクを中心に，詳細な誤り分析を行っている．

3 ニューラル機械翻訳モデル

本節ではエンコーダとデコーダから構成される seq2seq と Transformer について説明する．どちらの

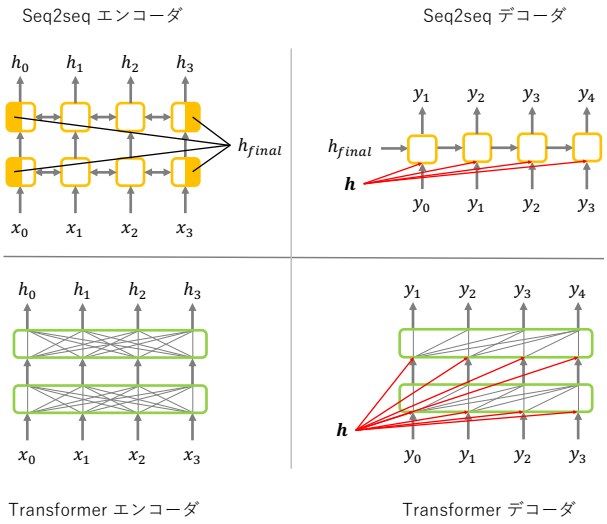

図 1: seq2seq と Transformer の概略図

エンコーダも原言語のトークン列 $\mathbf{x} = (x_1, ..., x_M)$ を入力とし，$\mathbf{h} = (h_1, ..., h_M)$ を出力するが，seq2seq のエンコーダについては，各 RNN の最終状態を連結した h_{final} も出力する（図 1）．また，どちらのデコーダもエンコーダの出力 $\mathbf{h}$ を入力とし，目的言語のトークン列 $\mathbf{y} = (y_1, ..., y_N)$ を出力するが，seq2seq のデコーダについては，h_{final} を用いて RNN の隠れ状態の初期化を行う．

3.1 構造の違い

seq2seq と Transformer の違いの一つは，入力系列に対する位置情報の扱いである．seq2seq は RNN を用いているため，系列データを逐次的に RNN に入力することにより，相対的な位置関係を扱っている．一方で，Transformer は位置情報の付加層を独立に設けており，絶対的な位置に基づいて定数を入力に加算している．この点において，相対位置を扱う seq2seq の方が，絶対位置を扱う Transformer に比べ，汎用的な性能を持つことが予想される．

注意機構の構造も大きく異なる．seq2seq は自己注意機構は持っておらず，系列データにおける前後の情報は，RNN の隠れ状態の遷移にのみ依存している．また，デコーダからエンコーダの出力への注意機構は，一時刻の処理に付き，一度きりかつ一つの文脈ベクトルのみを扱う．それに対して Transformer は，自己注意機構によりエンコーダについては入力全体，デコーダについて出力済系列へのアクセスが可能であり，さらに Multi-head 機構により複数の文脈ベクトルを扱う．また，デコーダからエンコーダの出力への注意機構は一時刻あたりデコーダの層数回の注意が張られ，かつ複数の文脈ベクトルを扱う．これらの点においては，seq2seq は扱える情報が局所的に縛られている一方で，Transformer は縛りのない大域的なアクセスが可能である優位性がある．

3.2 seq2seq-Transformer スワップモデル

本研究では，seq2seq と Transformer についてより詳細な分析を行うために，互いにエンコーダとデコーダを交換した 2 つのスワップモデルについても実験を行う．

seq2seq のエンコーダと Transformer のデコーダを組み合わせたモデルは，エンコーダの出力 $\mathbf{h}$ と RNN の最終隠れ状態 h_{final} のうち，前者のみをデコーダの入力とする．

また，Transformer のエンコーダと seq2seq のデコーダを組み合わせたモデルは，Transformer のエンコーダの出力 $\mathbf{h}$ をデコーダの入力の一つとするが，この場合，デコーダの RNN の隠れ状態の初期化を行う入力が存在しない．しかしながら，後の予備実験で示すように，seq2seq のデコーダは RNN の隠れ状態を全て 0 で初期化しても，エンコーダの最終隠れ状態で初期化を行った場合と同等の性能を発揮する．よって，このスワップモデルではデコーダの RNN の隠れ状態は全て 0 で初期化した．

4 実験

seq2seq，Transformer，及び互いにエンコーダとデコーダをスワップした 2 つのモデルの合計 4 つのモデルを比較する．（以降では便宜的に，エンコーダとデコーダの弁別に基づき，SS，ST，TS，TT と表記する．）

行う実験は次の 3 つである．i) モデル間の類似度を測るために，各モデルの翻訳結果同士の類似度を自動評価を用いて比較する．ii) 学習データ量と精度の関係を明らかにするべく，全てのモデルについて，学習データ量を変化させて学習を行う．iii) 翻訳文の長さと精度の関係を明らかにするべく，各モデルの翻訳結果を，原言語文の長さで分割して評価を行う．

実験設定　タスクは ASPEC [6] による英日翻訳を用いた．ASPEC は 1,783,817 文対の学習データ，1790 文対の開発データ，1812 文対のテストデータからなる．

表 1: 各モデルの BLEU スコアと RIBES スコア

	SS	ST	TS	TT	SS*
BLEU	*37.02*	**39.19**	37.74	38.65	36.94
RIBES	*82.01*	**83.33**	82.51	83.31	82.17

英語のデータは，Moses[1] (ver. 2.2.1) を用いてトークン化及び Truecasing を行い，日本語のデータは KyTea[2] (ver. 0.4.2) [12] を用いてトークン化を行った．以上の処理をしたデータに対して，さらに SentencePiece[3] を用いたトークン化を，両言語のデータを結合して行った．この時，分割アルゴリズムは unigram を選択し，語彙数は 16,000（両言語共有）とした．また，学習データは先頭から 1,500,000 文対までを用い，両言語文長が 50 以下の文対のみを使用した．

実装には PyTorch[4] (ver. 0.4.1) を用いた．Seq2seq のエンコーダには 3 層の双方向 LSMT を，デコーダには 1 層の単方向 LSTM を用い，Seq2seq の各 LSTM の隠れ状態は 512 次元とした．Transformer のエンコーダ，デコーダは共に 6 層とし，モデルサイズは 512 次元，フィードフォワード層は 2048 次元とした．

上記以外のパラメータは全て共通とし，埋め込み層は 512 次元，dropout の確率は 0.2 とした．最適化手法には初期学習率を 0.0001 とした Adam を用い，パラメータ更新時の勾配のノルムの最大値を 3.0 とした．ミニバッチサイズは 128 とし，400,000 ステップの学習を行った．以降の実験では，10,000 ステップ毎に開発データを用いてチューニングしたモデルを使用する．なお，Transformer について Vaswani ら [3] が採用している学習中の学習率の変更，及び正解データの調整は，条件を揃えるために本研究では使用していない．

自動評価による最高翻訳精度の比較　各モデルの翻訳精度を，自動評価手法である BLEU [4] と RIBES [13] を用いて評価した（図 1）．なお，予備実験として，SS モデルにおいて，デコーダの RNN の初期隠れ状態を 0 で初期化したモデル，SS*を追加した．SS と SS*はほぼ同等の性能を発揮していることが分かる．

翻訳結果に基づく比較　文単位の BLEU，文字単位の編集距離，トークン単位の編集距離，平均単語ベクトルによる文ベクトルの距離，単語ベクトルに基づく Word Mover's Distance [14] の 5 つの自動評価手法を用いて，各モデルが生成した翻訳文の相互類似度を評価する．なお，生のモデルの出力を評価すべく，SentencePiece によるトークンのまま比較を行った．また，単語ベクトルは，gensim (ver. 3.4.0) の word2vec をデフォルト設定で使用し，日本語の学習データを用いて学習を行った．5 つの評価手法全てにおいて，2 モデル間の類似度の順位が同一であったため，文単位の BLEU と文ベクトルの距離による評価結果を表 2 に示す．

全ての評価手法において，ST と TT が最も類似する結果となった．一方で，SS と TS が最も類似しない結果となった．これは，構造を一切共有しない SS と TT が最も類似しないだろうという直感から外れる結果である．また，最も類似している ST と TT，最も類似していない SS と TS が，どちらもデコーダを共有していることと合わせて，個別のエンコーダ・デコーダがモデルの出力の類似度に直接的には影響しないことが分かった．この結果から，エンコーダ及びデコーダは，それぞれ独立して機能する訳ではなく，組み合わせによりその機能を変えていると考えられる．

表 2: 翻訳文間の文単位 BLEU スコアの平均（左下），及び平均文ベクトル距離の平均（右上）

	REF	SS	ST	TS	TT
REF		2.93	2.85	2.93	2.85
SS	37.61		2.15	*2.24*	2.16
ST	39.87	57.86		2.12	**1.92**
TS	38.23	*56.48*	59.23		1.96
TT	39.2	57.17	**63.54**	62.22	

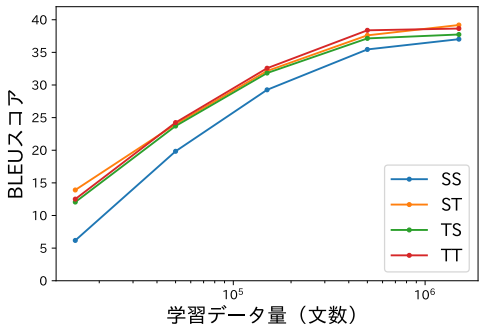

図 2: 学習データ量に対する各モデルの BLEU スコア

学習データ量に対する翻訳精度の比較　本実験では，学習に用いる学習データ量について，1,500,000 文対を全て用いる場合に，先頭から 15,000 文対，50,000 文

[1] http://www.statmt.org/moses/
[2] http://www.phontron.com/kytea/
[3] https://github.com/google/sentencepiece
[4] https://pytorch.org/

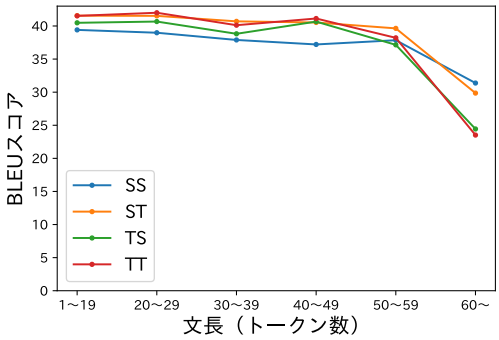

図 3: 原言語文の文長毎の BLEU スコア

対，150,000 文対，500,000 文対の場合を加えて，合計 5 つの場合に対して各モデルの学習を行い，その翻訳精度を比較する．翻訳精度の評価には自動評価手法である BLEU を用い，また，翻訳文は，SentencePiece によるトークンを平文に戻した後，KyTea にて再分割して評価を行った．結果を表 2 に示す．

全ての学習データ量の場合に渡って，SS が最も精度が低く，また，他の 3 つのモデルと異なる振る舞いをしていることが分かる．ST，TS が共に TT と類似した結果となっていることを考慮すると，この結果からも，エンコーダ及びデコーダは，組み合わせによりその機能を変えていると考えられる．

翻訳文の文長に基づく比較　テストデータの翻訳文を，翻訳前の原言語文の文長に基づいて分割し，別々に BLEU スコアを計算した．結果を図 3 に示す．

全てのモデルにおいて，学習時に除外した文長 50 以上の時点からスコアの下降が見られる．4 つのモデル間で下降の様子の比較をすると，SS と ST は緩やかで，TS と TT は急である．これらはそれぞれ共通のエンコーダを用いていることから，学習データ以上の文長をもつ文の翻訳については，seq2seq のエンコーダが，Transformer のエンコーダより優れていると考えられる．この 2 つの主な違いの一つは位置情報の扱いであり，学習データを超える範囲の位置情報については，RNN による相対的な位置の扱いが，絶対的な位置情報の付加手法に比べて優れていると推測される．

ドメイン外データセットでの精度　京都フリー翻訳タスク (KFTT)[5] のテストデータを用いて，ドメイン外での翻訳精度を比較した．しかし，全モデル中の最高 BLEU スコアでも 8.09 点と低く，翻訳結果にも未知語が非常に多かったため，有用な知見は得られなかった．

[5] http://www.phontron.com/kftt/

5　おわりに

本研究では，RNN に基づく NMT モデルである seq2seq と，自己注意機構に基づく NMT モデルである Transformer，そして，互いのエンコーダ・デコーダを交換した 2 つのスワップモデルの合計 4 つのモデルに対して翻訳タスクを通した比較分析を行った．比較にスワップモデルを含めたことにより，エンコーダ，デコーダが組み合わせにより機能を変えている可能性が示唆され，また，学習データ以上の長文については，相対的な位置情報を扱う RNN を用いたエンコーダの優位性が示された．

謝辞　本研究の一部は，情報通信研究機構の委託研究の成果です．

参考文献

[1] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[2] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N Dauphin. Convolutional Sequence to Sequence Learning. In *ICML*.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*.

[4] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[5] P. Koehn and R. Knowles. Six challenges for neural machine translation. In *NMT*, 2017.

[6] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian scientific paper excerpt corpus. In *LREC*, pp. 2204–2208, 2016.

[7] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico. Neural versus phrase-based machine translation quality: a case study. In *EMNLP*, 2016.

[8] A. Toral and Víctor M. Sánchez-Cartagena. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *EACL*, 2017.

[9] M. Xu Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, and M. Hughes. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*, 2018.

[10] Y. Ding, , Y. Liu, H. Luan, and M. Sun. Visualizing and understanding neural machine translation. In *ACL*, 2017.

[11] S. M. Lakew, M. Cettolo, and M. Federico. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. In *COLING*, 2018.

[12] G. Neubig, Y. Nakata, and S. Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *ACL-HLT, Short Papers*, pp. 529–533, 2011.

[13] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, and H. Tsukada. Automatic evaluation of translation quality for distant language pairs. In *EMNLP*.

[14] M. Kusner, Y. Sun, N.s Kolkin, and K. Weinberger. From word embeddings to document distances. In *ICML*.