複数の参照訳を考慮したニューラル機械翻訳モデルの学習手法

宮崎桂輔　徳永健伸
東京工業大学　情報理工学院
{miyazaki.k.am@m, take@c}.titech.ac.jp

概要

文の翻訳では訳者や文脈などの条件により異なる訳文が期待され，機械翻訳タスクのベンチマークに用いられる対訳コーパス内にも一つの原文に複数の参照訳が対応している例が多く存在する．一方で既存の機械翻訳モデルの学習手法ではこうしたコーパス内の構造は明示的には用いられない．そこで本研究では，コーパス内の原文が複数の参照訳を持っているという情報を，機械翻訳モデルの学習に明示的に用いる手法を提案する．評価実験において，提案手法を用いて学習したモデルは旧来の最尤誤差損失のみを用いたモデルに劣る結果となったが，提案手法が文レベル最適輸送損失を用いて学習するモデルの性能向上には寄与することが示された．

1　はじめに

文の翻訳において，訳者や文脈などの条件によって目的言語の訳文は異なるものが期待されうる．実際，機械翻訳タスクのベンチマークによく用いられる対訳コーパスにおいて，原文に対して複数の参照訳を持つ文対が共存することがある．日英対訳コーパス JESC[1] および英独対訳コーパス WMT14 En-De[2] の学習データセットにおいて，複数の参照訳を持つ原文の数および原文を共有する参照訳ののべ数を表 1 に示す．また，一つの原文が持つ参照訳の個数の分布を図 1 に示す．機械翻訳タスクによく用いられる対訳コーパスに，一つの原文が複数の参照訳を持つ例が無視できない割合で存在することがわかる．

一方で，既存の機械翻訳モデルの学習時に，対訳コーパス内に前述のような例が存在するという情報は明示的には用いられない．そこで，本研究では一つの原文が複数の参照訳を持っているという情報を明示的にモデリングした機械翻訳モデルの学習損失を提案する．

提案手法により，対訳コーパス内に存在する，原文が複数の参照訳を持っているという情報を機械翻訳モデルの学習に明示的に使用することができ，機械翻訳モデルの性能向上が見込まれる．なお，本研究では機械翻訳タスクのみを扱うが，提案手法は機械翻訳に限らず自己回帰生成を用いた文章生成全体に適用可能である．

表 1　学習データセットの統計量

	JESC	WMT14 En-De
文対数	2,797,388	3,896,364
複数参照訳を持つ原文の数	153,609	37,529
原文を共有する参照訳の数	574,519	118,003

2　関連研究

本研究では文レベル最適輸送損失 [3] を用いる．文レベル最適輸送損失は，モデルの生成文と参照訳を埋め込み空間上の点の集合とみなして得られる二文間の最適輸送コストであり，自動微分可能な最適輸送アルゴリズムである IPOT[4] を用いて計算することでモデルの損失として学習に用いるものである．旧来の最尤誤差損失に加えて文レベル最適輸送損失を学習に用いることにより LSTM ベースの GNMT モデル [5] の性能を向上させることが，検証により示されている [3]．

各時刻における次単語予測の最尤誤差の総和として計算される旧来の最尤誤差損失よりも，生成文と参照訳の間の最適輸送コストである文レベル最適輸送損失を活用する方が，複数の参照訳が存在するという情報をモデリングするのに適切であると考え，この既存研究に注目した．

3　提案手法

3.1　損失の定義

提案手法では文レベル最適輸送損失 [3] を用いて，一つの原文が複数の参照訳を持っているという情報を明示的に学習に利用する．文レベル最適

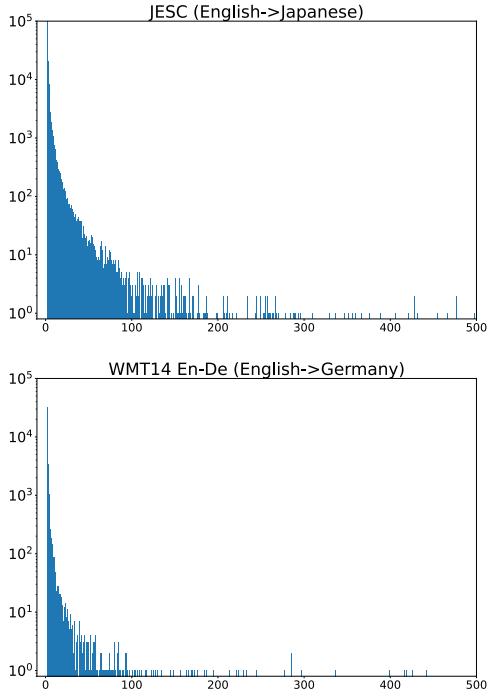

図 1 原文が持つ参照訳の数 x(横軸) と，参照訳を x 個持つ原文の数 (縦軸) の関係

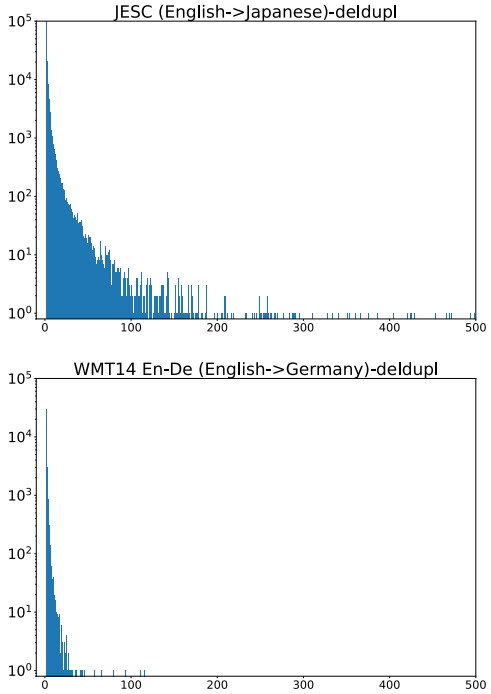

図 2 文対の重複を削除したデータセットにおける，原文が持つ参照訳の数 x(横軸) と，参照訳を x 個持つ原文の数 (縦軸) の関係

輸送損失は，モデルが生成した訳文 g，参照訳 r に対して，最適輸送アルゴリズム IPOT を用いて $\mathcal{L}_{\text{seq}}(g,r) = \text{IPOT}(g,r)$ と計算される．

データセット内の原文 s に対して，モデルが生成した訳文を g，参照訳の集合を $\mathcal{R} = (r_1,\ldots,r_n)$ としたとき，原文 s に対する複数の参照訳を考慮した文レベル最適輸送損失 $\mathcal{L}^*_{\text{seq}}(g,\mathcal{R})$ は，生成した訳文と各参照訳から計算される文レベル最適輸送損失の平均または総和とする．すなわち，複数の参照訳を考慮した文レベル最適輸送損失を

$$\mathcal{L}^*_{\text{seq}}(g,\mathcal{R}) = \frac{1}{n}\sum_{r\in\mathcal{R}}\mathcal{L}_{\text{seq}}(g,r) = \frac{1}{n}\sum_{r\in\mathcal{R}}\text{IPOT}(g,r) \quad (1)$$

または

$$\mathcal{L}^*_{\text{seq}}(g,\mathcal{R}) = \sum_{r\in\mathcal{R}}\mathcal{L}_{\text{seq}}(g,r) = \sum_{r\in\mathcal{R}}\text{IPOT}(g,r) \quad (2)$$

と定める．

3.2 損失の実装

提案手法はミニバッチ作成の工夫による実現が可能である．具体的には，同一の原文を持つ対訳が同じミニバッチに入るような制約を課してミニバッチを作成すればよい．なお，本研究では fairseq[6] で実装されているミニバッチ作成のアルゴリズムを利用し，以下の三通りのミニバッチ作成アルゴリズムを用いる．

- 参照訳の文長が近い文対を集めてミニバッチを作成する方法 (ベースラインの手法．以下，”base”と記載)
- 同一の原文を持つ対訳が同じミニバッチに入る制約を設け，参照訳の最大文長が近い原文を集めてミニバッチを作成する方法 (以下，”samebatch”と記載)
- 同一の原文を持つ対訳が同じミニバッチに入り，かつミニバッチ内に参照訳を複数持つ原文と参照訳を一つのみ持つ原文が共存しない，という制約を設けて，参照訳の最大文長が近い原文を集めてミニバッチを作成する方法 (以下，”divided”と記載)

なお，同一の原文を持つ対訳の数が多すぎて一つのミニバッチに入りきらない例が存在した場合は，制約を緩和し，複数のミニバッチに分けたうえで，連続する学習ステップで学習に用いられるような処理を施すこととした．

また，文レベル最適輸送損失の平均をとる式 (1) と総和をとる式 (2) の二種類の計算方法を提案した．これらは，ミニバッチ内で損失の和を計算する際に単純に総和をとる方法（ベースラインの手法．以下 “base” と記載）と，参照訳を n 個持つ原文に関する損失を $1/n$ するような重みつき和をとる方法（以

下，“decloss” と記載）により実現可能である．

さらに，本研究で扱うコーパスにおいて，原文・参照訳ともに一致する対訳が散見された．そこで，コーパス内から原文・参照訳ともに一致するような対訳の重複を削減する措置（以下，“deldupl” と表記）をとった．対訳の重複を削減した後のデータセットにおける，一種類の原文が持つ参照訳の個数の分布を表 2 に示す．なお，この処理で削除された文対の数は，JESC が 1,502 文対，WMT14 En-De が 36,274 文対であった．この処理を行わない手法は以下 “base” と表記する．

以上より，実装におけるオプションの種類は

1. ミニバッチ作成の手法：base，samebatch，divided
2. ミニバッチ内の損失の集約に関する手法：base，decloss
3. 重複する対訳の処理に関する手法：base，deldupl

のとおりである．以下ではオプションの選び方に応じて，手法の表記を “[ミニバッチ作成の手法]+[ミニバッチ内の損失の集約に関する手法]+[重複する対訳の処理に関する手法]” のように記す．提案手法は samebatch+base+deldupl，または samebatch+decloss+deldupl と表せる．

4 実験設定

4.1 データセット

本研究では，JESC[1] を用いた英日翻訳と，WMT14 En-De[2] を用いた英独翻訳を行った．

JESC は，MeCab[7] を用いて形態素単位の分かち書きを行ってから BPE によるサブワード分割を行った．BPE のマージ回数は，日本語，英語ともに 32,000 から文字ベースの語彙数を引いた値とした．最終的な英語の語彙数は 31,932，日本語の語彙数は 31,604 となった．

WMT14 En-De は，fairseq で公開されている前処理スクリプト[1)]を用いてデータの前処理および BPE[8] によるサブワード分割を行った．なお，BPE のマージ回数は 32,000 とした．最終的な英語の語彙数は 33,616，ドイツ語の語彙数は 34,888 となった．

4.2 モデル

モデルは Transformer の base モデル [9] を用いた．ベースラインモデルとして，学習に最尤誤差損失のみを用いるモデルを作成した．さらに，最尤誤差損失と文レベル最適輸送損失の和を用いるモデルを作成した．このモデルは，提案手法のオプションが base+base+base である場合に相当する．

提案手法として，最尤誤差損失と文レベル最適輸送損失の和を用い，さらにオプションとして samebatch+base+deldupl および samebatch+decloss+deldupl を用いるモデルを作成した．

さらに，アブレーション実験として，3.2 節に示した提案手法のオプションの種類を組み合わせた全 12 種類のオプションについてもモデルを作成した．

4.3 学習設定

最適化アルゴリズムとして Adam[10] を用いた．学習率は固定の値とせず，ウォームアップステップ中の学習率は初期学習率の 10^{-7} から最大学習率に線形に推移するようにし，ウォームアップステップ以降はステップ数の平方根に反比例するように学習率を減衰させた．学習率の最大値は，$\{0.0001, 0.0002, 0.0005, 0.000666, 0.001\}$ から探索を行った．ウォームアップステップ数は，JESC の場合 6,000，WMT14 En-De の場合 27,000 とした．

ドロップアウト率は 0.3 とし，最尤誤差損失の計算時に平滑化値 0.1 のラベル平滑化 [11] を行った．

文レベル最適輸送損失では Soft-copying mechanism は適用しないものとし，微分可能な文章生成のために argmax の代わりに温度 $\tau = 0.1$ の Soft-argmax を用いた．最尤誤差損失に足し合わせる際の重みパラメータ γ は 0.1 とした．

ミニバッチサイズは 1GPU あたり 3,584 トークンとし，NVIDIA Tesla P100 GPU を 4 つ用いて学習を行った．

最大学習ステップ数は 150,000 とした．1 エポック毎に検証データセットに対するモデルの損失を計算し，損失が一番低いモデルを選択した．

4.4 評価方法

各条件においてモデルを 1 回のみ学習し，評価を行った．評価指標は BLEU とし，脱トークン化した生成文に対して sacreBLEU[12] により算出した．ただし，JESC を用いた評価時には，目的言語側である日本語のテストセット内の文を MeCab を用いて形態素単位に分かち書きしてから評価に用いた．

生成時のビームサーチの探索幅については，JESC の場合 1 から 10 までの間で 1 刻みに探索を行っ

1) https://github.com/facebookresearch/fairseq/blob/main/examples/translation/prepare-wmt14en2de.sh

た．WMT14 En-De の場合は，4 に固定した．ビームサーチに付随する文長に対する罰則は，JESC，WMT14 En-De ともに 0.5 から 1.5 までの間で 0.1 刻みに探索を行った．

4.5 追加実験

4.3 節に示した学習設定で実験を行ったところ，WMT14 En-De における BLEU スコアが先行研究 [9] で報告されている値よりも著しく低いものとなった．そこで，追加実験として，WMT14 En-De を用い，学習パラメータを変えて再度実験を行った．前述した 1 回目の実験と異なる点は，ミニバッチサイズを 1GPU あたり 4,096 トークンとし，update_freq を 2 とすることで実質ミニバッチサイズを 1GPU あたり 8,192 とした点，学習率を 0.0007 とした点，ウォームアップステップ数を 4,000 とした点，ドロップアウト率を 0.1 とした点である．

5 実験結果

実験結果を表 2 に示す．なお，ベースラインモデルの一つである base+base+base よりも高い BLEU スコアは太字で，各データセットにおける最高スコアは太字＋下線で示した．

提案手法である samebatch+base+deldupl は，文レベル最適輸送損失を用いたベースラインモデルである base+base+base よりも一貫して高い性能を示した．この結果から，文レベル最適輸送損失を用いて学習を行う際に samebatch+base+deldupl の提案手法を追加することの有効性は示された．

一方で，JESC および WMT14 En-De の一回目の実験において，文レベル最適輸送損失を用いたすべてのモデルが旧来の最尤誤差損失のみを用いるモデルに劣る結果となった．WMT14 En-De の二回目の実験においても，文レベル最適輸送損失を用いたベースラインモデルをはじめとする複数のモデルが旧来の最尤誤差損失のみを用いるモデルに劣る結果を示した．この原因として，本実験におけるパラメータのチューニングが不足しており文レベル最適輸送損失の効果を引き出しきれていないこと，または文レベル最適輸送損失の有効性が限定的であることが考えられる．

また，アブレーション実験を見ると，複数の参照訳を持つ原文の割合が比較的高かった JESC において，バッチ作成アルゴリズムに divided を用いた際の結果が base+base+base よりも 0.4 ポイント以上低くなっている．さらに，バッチ作成アルゴリズムに samebatch を用い，deldupl を適用しなかった場合についても base+base+base より劣る結果となっている．このことから，複数の参照訳を持つ原文の割合が高く，一つの原文に対して多数の参照訳が紐づく例が存在するデータセットにおいては，参照訳の数が極端に多い原文に影響を受けて，バッチ作成アルゴリズムを divided にするとバッチ作成時の制約が強すぎてモデルの学習がうまくいかないこと，そして samebatch を用いる場合も deldupl を適用しないと有効なバッチの作成が行えないことが考えられる．

表 2 テストデータセットに対する BLEU スコア．dec. は decloss，del. は deldupl の適用の有無を示す．

モデル batching	dec.	del.	JESC	WMT14 En-De 1 回目	WMT14 En-De 2 回目
Transformer base	-	×	**13.91**	**24.85**	**25.89**
Transformer-OT base	×	×	13.45	24.47	25.87
samebatch	×	○	**13.58**	**24.75**	**26.03**
samebatch	○	○	**13.70**	**24.62**	25.64
base	○	×	**13.71**	**24.63**	**26.14**
base	×	○	13.40	**24.64**	**26.03**
base	○	○	13.28	**24.60**	25.71
samebatch	×	×	13.19	**24.65**	**26.14**
samebatch	○	×	13.10	**24.54**	**26.15**
divided	×	×	12.53	**24.65**	25.81
divided	○	×	13.00	**24.62**	25.81
divided	×	○	12.47	24.38	**25.92**
divided	○	○	13.04	24.33	25.85

6 おわりに

本研究では，コーパス内の原文が複数の参照訳を持っているという情報を，機械翻訳モデルの学習に明示的に用いる手法を提案した．実験において提案手法は旧来の最尤誤差損失のみを用いる学習手法に劣る結果となったが，文レベル最適輸送損失を用いるモデルに限れば提案手法が既存手法よりも良い性能を示した．

今後取り組むべき課題として，モデル学習時のパラメータの探索を行うことが挙げられる．

今後の展望としては，データセット内の複数参照訳を持つ原文の割合が提案手法の性能に与える影響を調査することを検討している．また，近年機械翻訳を含む幅広いタスクで活用されている近傍事例を用いて，提案手法を拡張することを検討している．

謝辞

研究について貴重なコメントをくださった山中光さん，飯島慧さん，論文のレビューをくださった石渡太智さん，伊藤光一さん，鈴木偉士さんに感謝の意を表す．

参考文献

[1] Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. JESC: Japanese-English subtitle corpus. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[2] Ondřej 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 **Proceedings of the Ninth Workshop on Statistical Machine Translation**, pp. 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[3] Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. Improving sequence-to-sequence learning via optimal transport. In **International Conference on Learning Representations**, 2019.

[4] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In Ryan P. Adams and Vibhav Gogate, editors, **Proceedings of The 35th Uncertainty in Artificial Intelligence Conference**, Vol. 115 of **Proceedings of Machine Learning Research**, pp. 433–453. PMLR, 22–25 Jul 2020.

[5] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

[6] 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.

[7] MeCab: Yet Another Part-of-Speech and Morphological Analyzer. https://taku910.github.io/mecab/.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30, pp. 5998–6008. Curran Associates, Inc., 2017.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**, 2015.

[11] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions, 2017.

[12] Matt Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation: Research Papers**, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics.