抽出型要約と言い換えによる生成型要約の訓練データ拡張

Loem Mengsay　高瀬 翔　金子 正弘　岡崎 直観
東京工業大学情報理工学院
{mengsay.loem,sho.takase,masahiro.kaneko}@nlp.c.titech.ac.jp
okazaki@c.titech.ac.jp

概要

大量の訓練データを用いたニューラルモデルは生成型要約タスクにおいて高い性能を達成している．しかしながら，大規模な並列コーパスの構築はコストの観点から容易ではない．これを解決するため，本研究では生成型要約タスクの疑似訓練データを低コストで効果的に構築する手法を提案し，訓練データを拡張する．提案手法は抽出型要約と言い換えの2つのステップで疑似訓練データを構築する．抽出型要約では入力テキストの主要部分を抽出し，言い換えではその多様な表現を得る．実験を通して，提案手法は生成型要約タスクの性能を向上させ，逆翻訳や自己学習などの既存の訓練データ拡張手法を上回ることを確認する．

1　はじめに

ニューラルエンコーダ・デコーダは，機械翻訳や自動要約などの様々な系列変換タスクにおいて顕著な性能を達成している [1, 2]．近年の研究においてニューラル手法の性能は訓練データ量に対数比例することが指摘されており [3]，系列変換タスクにおいて高い性能を達成するためには，大規模な並列コーパスが必要である [4]．本稿では，生成型要約タスクにおけるニューラルエンコーダ・デコーダの性能を向上させるために，訓練データを拡張することに取り組む．

人手による並列コーパスの構築はコストが高いため，既存研究では疑似訓練データを自動的に構築する方法が検討されている．疑似訓練データを構築する方法としては，逆翻訳 [5] が広く用いられている．翻訳タスクにおける逆翻訳では，翻訳先の言語の文から翻訳元の言語の文を生成するモデルを学習し，得られたモデルを翻訳先の言語のコーパスに適用し，翻訳元の言語の疑似コーパスを生成する．この逆翻訳手法を要約に適用した場合，モデルは要約から原文書を生成する必要があるため，要約タスクにおける逆翻訳は本質的に非現実的である（付録 A）．

He ら [6] は，自己学習が機械翻訳や要約タスクの性能を向上させることを示した．自己学習では教師モデルを学習し，そのモデルを入力側のコーパスに適応し，出力側の疑似コーパスを生成する．逆翻訳が非現実的な処理であったのに対し，自己学習による疑似要約の生成は合理的である．しかしながら，自己学習を適用した場合，多様な要約の生成が困難であると指摘されている [7]．これらの問題に加え，自己学習や逆翻訳において高品質な疑似データを得るためには，大量の訓練データで教師モデルや逆翻訳モデルを学習しておく必要があるため，データ構築のコストが高い．

そこで，本研究では生成型要約の疑似訓練データを構築する新たなアプローチとして，抽出型要約と言い換えの組み合わせによる手法を提案する．提案手法は原文の統語構造を基に，ヒューリスティックな手法で重要な部分を要約として抽出する．このため，逆翻訳や自己学習と異なり，疑似訓練データ作成のためだけにニューラルモデルを構築する必要がない．抽出された要約に対し，既存のモデルを活用した言い換え手法を適用し，多様な疑似要約を得る．

本研究では見出し生成タスクと文書要約タスクで実験を行う．実験を通して，提案手法による疑似訓練データは両タスクにおいて性能を向上させることを確認した．具体的には，疑似データを用いることにより両タスクの ROUGE F1 スコアが 0.50 以上向上した．また，生成された疑似データの性質を分析し，提案手法による疑似データの生成が従来手法より効率的であることを示した．提案手法は真の訓練データが少ない低リソースの設定においても頑健であることが確認された（付録 B）．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

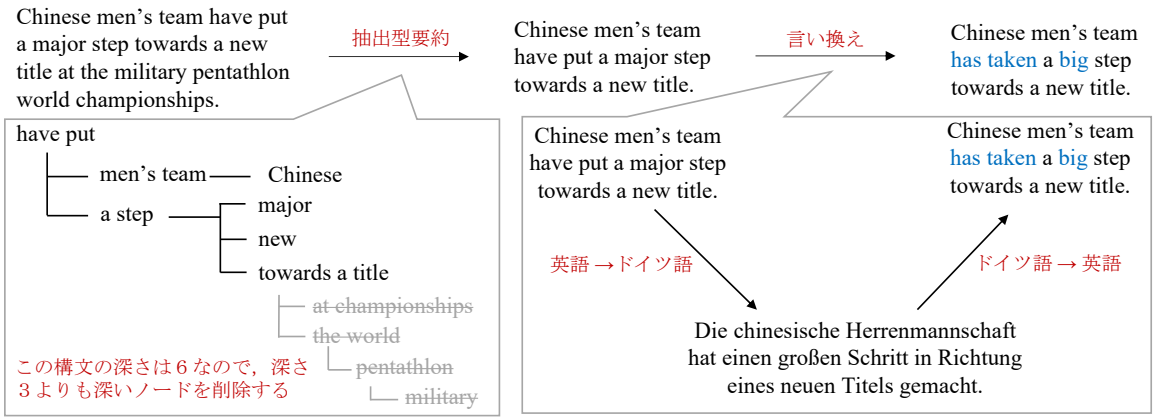

図 1　提案手法による疑似データの生成過程の例.

2　提案手法

1 節で述べたように，提案手法は抽出型要約と言い換えから構成される．図 1 に提案手法の概要を示す．提案手法は文を入力とし，その文に対応する疑似要約文を生成する．このため，文書から疑似要約を構築する場合には，文書内に含まれる複数の文に対して独立に提案手法を適用し，各文の要約文の集合を文書の要約とする．

2.1　ステップ 1：抽出型要約

抽出型要約のステップでは，原文の中から重要な部分を抽出する，すなわち，入力文の文圧縮を行う．先行研究ではルールベースの手法 [8]，構文木から重要部分を検出するアプローチ [9]，ニューラルベースの手法 [10, 11] などの圧縮方法が提案されている．本研究では，コストの少ない手法，すなわち，新たにモデルを構築する必要のない手法として，与えられた文の構文木を基に文圧縮を行う．

本研究では，文の重要な部分は与えられた文の構文木の根付き部分木であると定義する．まず，与えられた文を係り受け解析し，その係り受け木を得る．次に，係り受け木の深いノード，すなわち，末端に近いノードを刈り，元の構文木よりも深さの小さい根付き部分木を得る．本手法では部分木の深さによって，出力要約の長さ（単語数）を調整することができる．本研究では，係り受け木の深さの半分より深いノードを刈る．図 1 の下段（左）に本ステップの例を示す．

2.2　ステップ 2：言い換え

ステップ 1 により抽出された要約は原文に含まれる単語のみで構成されている．要約の多様性を高めるために要約に言い換えの手法を適用する．高品質なニューラル機械翻訳モデルが公開されているため，言い換えには機械翻訳モデルを用いたアプローチ [12] を採用する．具体的には，文を別の言語に翻訳し，翻訳された文を元の言語に翻訳する折り返し翻訳を行い，言い換えを得る．図 1 の下段（右）に本ステップの例を示す．

3　実験

提案手法の効果を調べるため，見出し生成と文書要約の 2 つの要約タスクで実験を行った．見出し生成タスクでは Gigaword データセット [2] を用いた．このデータセットには，英語 Gigaword コーパスから抽出された約 380 万の文書の 1 文目と見出し文の対が訓練データとして含まれる．文書要約タスクでは，単一の文書要約タスクに広く利用されている CNN/DailyMail データセット [13] を用いた．このデータセットには，CNN と DailyMail のウェブサイトから抽出されたニュース記事と要約の 28 万対が訓練データとして含まれている．ニューラルエンコーダ・デコーダモデルとして fairseq[1] に実装されている Transformer[14] を用いた．

3.1　比較手法

既存研究で提案されてきた代表的な訓練データ拡張手法との比較を行う．疑似データの構築は各データセットの訓練データをもとに行う．なお，Caswell

1）https://github.com/pytorch/fairseq

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1　見出し生成タスクと文書要約タスクにおける ROUGE F1 スコア．丸括弧の中は疑似訓練データ数を示す．

手法	見出し生成				文書要約			
	訓練事例数	R-1	R-2	R-L	訓練事例数	R-1	R-2	R-L
拡張なし	380 万	37.95	18.80	35.05	28 万	39.76	17.55	36.75
オーバーサンプリング	760 万	38.26	19.14	35.41	56 万	40.14	17.86	37.05
逆翻訳	760 万（380 万）	38.49	19.24	35.63	56 万（28 万）	39.93	17.74	36.85
自己学習	760 万（380 万）	38.32	19.06	35.37	56 万（28 万）	40.19	17.87	37.21
提案手法	760 万（380 万）	**38.51**	**19.52**	**35.72**	56 万（28 万）	**40.57**	**18.22**	**37.51**

ら [15] に従い，全ての疑似訓練データについての入力の先頭に <Pseudo> の特殊トークンを付加する．

オーバーサンプリング　訓練データから原文書と要約の対をサンプリングし，訓練データに追加する．すなわち，この手法で構築される訓練データには，真の訓練データのみが含まれる．

逆翻訳　訓練データを用いて，要約から原文を生成するニューラルエンコーダ・デコーダの学習を行う．次に，このモデルを訓練データ内の要約に適用し，対応する原文書を生成する．生成された原文書と真の要約の対を疑似訓練データとして使用する．

自己学習　各訓練データを用いて，原文書から要約を生成するニューラルエンコーダ・デコーダを学習する．次に，訓練データの原文をニューラルエンコーダ・デコーダに入力し，対応する要約を生成する．真の原文書と生成された要約の対を疑似訓練データとして使用する．

提案手法　各訓練データに対して提案手法を適用する．2 節で説明したように，提案手法は文単位で疑似要約を生成する．ニューラルエンコーダ・デコーダでは先頭の数文を入力とすることが多いことにならい，文書要約タスクでは，原文書に含まれる冒頭の 3 文を提案手法への入力とし，各文の要約を元の順序で連結して原文書の要約とする．抽出型要約ステップの係り受け解析には，spaCy[2] を使用する．言い換えステップでは，Ng ら [16] が構築した英独・独英翻訳モデル[3] を使用する．

3.2　結果

表 1 に真の訓練データのみを用いた場合と，各データ拡張手法を用いた手法の ROUGE スコアを示す．オーバーサンプリングは拡張を行わない場合と比較して高いスコアを達成している．この結果は，重複した学習事例であっても訓練データ量が増えるほど，ニューラルエンコーダ・デコーダの性能が向上することを示唆している．逆翻訳と自己学習は拡張を行わない場合よりも性能は高いが，オーバーサンプリングと同程度のスコアである．これらの結果は，性能改善は両者が生成する疑似データの品質ではなく，学習データの増加によってもたらされたことを示唆している．逆翻訳と自己学習は疑似データの生成に別のモデルを学習する必要があるため，そのコストを考慮するとオーバーサンプリングが優れている．これらに対し，提案手法は他のデータ拡張手法よりも高い性能を達成した．特に，提案手法による疑似訓練データは見出し生成における ROUGE-2 スコアを大幅に向上させた．文書要約では，拡張を行わない場合と比較して提案手法によるデータ拡張は全ての ROUGE スコアを有意に向上させた[4]．これらの結果から，提案手法はオーバーサンプリング，逆翻訳，自己学習を含む既存のデータ拡張手法よりも，生成型要約タスクのための疑似データを構築するのに有効であると言える．

表 2　生成された疑似要約の真の要約に対する BLEU と F1 BERTScores.

タスク	手法	BLEU	BERTScore
見出し生成	自己学習	28.64	92.44
	提案手法	1.51	86.19
文書要約	自己学習	19.91	90.02
	提案手法	5.89	87.33

4　分析

4.1　疑似データの多様性

1 節で述べたように，提案手法は自己学習より多様な要約を生成するために言い換えを行なっている．この効果を検証するために，自己学習と提案手法により生成された疑似要約を比較する．表 2 は各訓練データにおける真の要約と生成された疑似

2）https://spacy.io/
3）https://github.com/pytorch/fairseq/tree/main/examples/translation
4）スチューデントの t 検定で $p < 0.05$.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 3 疑似データ生成するのに要する時間と費用.

タスク	手法	訓練	生成	費用
見出し生成	逆翻訳	256 H	7 H	333 USD
	自己学習	256 H	4 H	328 USD
	提案手法	–	7 H	12 USD
文書要約	逆翻訳	384 H	16 H	511 USD
	自己学習	320 H	8 H	417 USD
	提案手法	–	15 H	26 USD

要約の BLEU スコアを示している．また，意味的類似性の指標として F1 ベースの BERTScore[17] も示している．この表から，自己学習と提案手法の BERTScore は高いことがわかる．この結果は生成された要約が真の要約と意味的に類似していることを意味する．このことから，いずれの方法で生成された要約も疑似データとして意味的に適していると言える．

一方，提案手法の BLEU スコアは自己学習のスコアよりも低い．この結果は，自己学習と比較して提案手法が真の要約と異なるフレーズを多く含んだ疑似要約を生成していることを示している．すなわち，提案手法で構築した訓練データは自己学習よりも多様な要約を含むことが分かる．

4.2 疑似データ生成の効率

提案手法は公開されている翻訳モデルをそのまま用いることができるため，逆翻訳や自己学習のように翻訳・生成モデルを学習する必要はない．そのため，既存手法と比較すると低コストで疑似データを構築できる．表 3 は各疑似データの構築手法の所要時間[5)]を示す．また，クラウドコンピューティングサービスである Amazon EC2 を用いて疑似データを構築した場合に必要な費用も示している．表 3 から，逆翻訳と自己学習はモデルの学習に多くの時間を要することがわかる．一方，提案手法では学習を必要としないため，他の手法と比較して 1/10 以下の金額で疑似データを構築できる．

5 関連研究

逆翻訳と自己学習は，系列変換タスクのデータ拡張手法として広く使われている [5, 18, 6]．逆翻訳は機械翻訳において有効なアプローチであるが，1 節で述べたように，要約タスクに適用するのは非現実的である．また，自己学習は機械翻訳や要約タスクにおいて有効な手法と報告されている [19, 6] が，多様な疑似データを生成することは困難である [7] ため，性能の向上に限界がある．

訓練データとの差異が小さい摂動を用いて性能改善を行う手法は，データ拡張とみなすことができる [20]．Takase ら [21] は，単語ドロップアウトや単語置換のような単純な手法は敵対的摂動よりも効率的に性能を向上させられることを示した．これらの摂動は本研究の提案手法と独立しているため，組み合わせることでさらに性能向上が期待できる．

提案手法の抽出型要約のステップでは文圧縮の手法を用いている．Dorr ら [8] は言語学的に動機づけられたヒューリスティックを用いたルールベースの文圧縮手法を提案した．Filippova ら [9] は訓練データから重要な部分木を学習する教師あり文圧縮法を提案した．また，近年ではニューラルネットを用いた文圧縮の研究も行われている [10, 11]．本研究では教師ありモデルや学習コーパスを必要としない，文の構文木に基づくルールベースの手法を採用し文の圧縮を行った．

提案手法は抽出された要約に対して言い換えを行い，疑似データの多様性を高めている．Bolshakov ら [22] は辞書に基づき単語を同義語に置き換えることで，言い換えを行う手法を提案した．最近の研究は，言い換えタスクを系列変換タスクとして定式化することにより，ニューラルベースの言い換え生成を行っている [23]．機械翻訳モデルを用いた折り返し翻訳による言い換え文生成も提案されている [12]．近年は高性能な機械翻訳モデルは公開されているため，本研究では言い換え生成の手法として折り返し翻訳を採用した．

6 おわりに

本研究では，生成型要約タスクのための疑似データを生成する新しい手法を提案した．提案手法は抽出型要約と言い換えの 2 つのステップから構成される．抽出型要約により入力の重要な部分を見つけ，言い換えにより多様な表現を獲得する．実験の結果，提案手法は逆翻訳や自己学習の疑似データ生成手法と比較して，より効果的であることが示された．また，提案手法は疑似データ生成において追加でモデルを学習する必要がないため，他の手法に対して，コストの面でも優れていることを示した．

謝辞 本研究は JSPS 科研費 19H01118 および JP21K17800 の助成を受けたものです．

5) 消費時間は 1GPU の場合で計算している

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In **3rd International Conference on Learning Representations**, 2015.

[2] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 379–389, 2015.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In **Advances in Neural Information Processing Systems 33**, pp. 1877–1901, 2020.

[4] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In **Proceedings of the First Workshop on Neural Machine Translation**, pp. 28–39, 2017.

[5] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics**, pp. 86–96, 2016.

[6] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In **International Conference on Learning Representations**, 2020.

[7] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In **International Conference on Learning Representations**, 2018.

[8] Bonnie Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: A parse-and-trim approach to headline generation. In **Proceedings of the HLT-NAACL 03 Text Summarization Workshop**, pp. 1–8, 2003.

[9] Katja Filippova and Yasemin Altun. Overcoming the lack of parallel data in sentence compression. In **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing**, pp. 1481–1491, 2013.

[10] Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. Sentence compression by deletion with LSTMs. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 360–368, 2015.

[11] Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. Higher-order syntactic attention network for longer sentence compression. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1716–1726, 2018.

[12] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Paraphrasing revisited with neural machine translation. In **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 881–893, 2017.

[13] 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**, pp. 1715–1725, August 2016.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in Neural Information Processing Systems 30**, pp. 5998–6008, 2017.

[15] Isaac Caswell, Ciprian Chelba, and David Grangier. Tagged back-translation. In **Proceedings of the Fourth Conference on Machine Translation**, pp. 53–63, 2019.

[16] Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. Facebook FAIR's WMT19 news translation task submission. In **Proceedings of the Fourth Conference on Machine Translation**, pp. 314–319, 2019.

[17] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. **CoRR**, Vol. abs/1904.09675, , 2019.

[18] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. An empirical study of incorporating pseudo data into grammatical error correction. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing**, pp. 1236–1242, 2019.

[19] Jiajun Zhang and Chengqing Zong. Exploiting source-side monolingual data in neural machine translation. In **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 1535–1545, 2016.

[20] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 452–457, 2018.

[21] Sho Takase and Shun Kiyono. Rethinking perturbations in encoder-decoders for fast training. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 5767–5780, 2021.

[22] Igor A. Bolshakov and Alexander Gelbukh. Synonymous paraphrasing using wordnet and internet. In **Natural Language Processing and Information Systems**, pp. 312–323, 2004.

[23] Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual LSTM networks. In **Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers**, pp. 2923–2934, 2016.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 4　逆翻訳により生成された原文と真の原文との ROUGE F1 スコア．比率と差 は，生成された原文と真の原文に含まれるトークン数の比較である．

タスク	比率	差	R-1	R-2	R-L
見出し生成	0.86	-5	35.14	15.13	28.59
文書要約	0.81	-297	13.76	1.09	13.07

A　逆翻訳による疑似データ

1 節で述べたように，生成型要約タスクにおける逆翻訳アプローチは，要約から原文を復元する必要があるため，本質的に不可能である．そこで，逆翻訳により生成された原文の性質を調べた．

表 4 は，逆翻訳により生成された原文の長さ（トークン数）の比率と差とを示す．生成された原文は元の原文より短いことが分かる．この結果は，逆翻訳が真の原文の情報を完全に復元できていないことを示している．つまり，要約から原文を生成することが難しいことを意味している．

また，逆翻訳により生成された原文が真のデータに対応しているかを調べるために，真の原文を正解と見なした場合の ROUGE スコアを表 4 に示す．文書要約の場合，ROUGE スコアは極めて低くなっている．この結果からも，逆翻訳は原文の生成に失敗していることがわかる．

一方，見出し生成の ROUGE スコアは文書要約の ROUGE スコアと比較して高い．この結果は，逆翻訳は要約から原文の核となる部分を復元できる可能性を示している．見出し生成タスクは，与えられた文章から見出しを生成するタスクであるため，要約（見出し）には原文の主要な部分が含まれていることが多い．このような性質が高い ROUGE スコアの要因になっていると考えられる．

B　低リソース設定

2 節で述べたように，提案手法は訓練データの量が少ない場合でも，頑健であると予想される．この仮説を調べるために，低リソースな訓練データの設定で実験を行なった．

見出し生成タスクと文書要約タスクの各訓練セットから，1 千件の事例を抽出する．抽出された事例は，真の訓練データとし，残りの事例を疑似データの生成に利用する．比較手法と実験設定は 3 と同様である．

表 5 と 6 に，見出し生成タスクと文書要約タスクにおける，各手法の ROUGE F1 スコアを示す．両タスクにおいて，オーバーサンプリングは拡張なしを上回る性能を得ることがわかる．このように，重複した学習データはニューラルエンコードデコーダモデルの性能を向上させる．この結果は，3.2 の結果と整合性がある．

表 5　低リソース設定における見出し生成タスクの結果．

手法	R-1	R-2	R-L
拡張なし	4.84	0.58	4.66
オーバーサンプリング	9.89	1.39	9.30
逆翻訳	12.19	2.43	11.31
自己学習	7.27	1.07	6.98
提案手法	**23.58**	**6.56**	**21.12**

表 6　低リソース設定における文書要約タスクの結果．

提案手法	R-1	R-2	R-L
拡張なし	2.48	0.29	2.45
オーバーサンプリング	13.63	0.89	12.63
逆翻訳	9.73	0.50	8.92
自己学習	14.37	1.52	13.36
提案手法	**34.47**	**12.91**	**31.36**

表 5 により，見出し生成では，逆翻訳がオーバーサンプリングより高いスコアを達成している．また，表 6 により，文書要約では自己学習がオーバーサンプリングより優れていることが分かる．この結果は，適切なタスクに適用すれば，既存のアプローチがオーバーサンプリングよりも効果的である可能性を示している．

一方，提案手法は両タスクにおいて他の手法より有意に高い性能を達成している．よって，提案手法は真の訓練データの量が少ない場合にも有効であることがわかる．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).