エンコーダ・デコーダの学習に効果的な摂動の調査

高瀬 翔
東京工業大学
sho.takase@nlp.c.titech.ac.jp

清野 舜
理化学研究所 / 東北大学
shun.kiyono@riken.jp

1 はじめに

機械翻訳や要約，文法誤り訂正などの系列変換タスクにおいて，エンコーダ・デコーダモデルは目覚ましい性能を達成している [1, 2, 3]．ニューラルモデルの表現力は極めて高いため，過学習を防ぐために，ドロップアウトを始めとして，様々な正則化手法が提案されてきた [4]．

ニューラルモデルの正則化のために，元々の入力に対して，摂動と呼ばれる微小な変化を加えた事例の使用が提案されている [5, 6]．学習に摂動を用いた際は，摂動を加えた事例と加えていない事例のいずれに対しても，正確なラベルを予測するよう，モデルを学習する．系列変換タスクにおいては，入力文内のトークンの一部を別のトークンに置換したものと，埋め込み空間において微小なノイズを加えたものという，2 種類の摂動事例が用いられている [7, 8]．例えば Bengio ら [9] はスケジュールドサンプリングという，デコーダの出力確率分布を元にトークンをサンプリングし，これを摂動としてデコーダの入力に用いる手法を提案した．Sato ら [8] は，モデルの損失を大きくする敵対的摂動をエンコーダ・デコーダの埋め込み空間に適用している．

これらの研究は提案手法によって，頑健なエンコーダ・デコーダモデルが構築できたと報告している．一方で，これらの手法は摂動を計算するために，最低一回は前向き計算を行わなければならないため，摂動を用いず，単純にエンコーダ・デコーダを学習する場合と比べて，学習に多大な時間を要する．実際，スケジュールドサンプリングにおいては，用いる摂動の個数と同じ回数デコーダを計算しなければならない．また，敵対的事例を用いる場合には，勾配を用いて摂動を計算するため，前向き計算に加えて，誤差逆伝播を行う必要がある [9, 8]．

学習に多大な時間を要する性質は，学習に要する費用の増大につながる．例えば，Transformer (big) [10] モデルについて，広く使われている，WMT 英-独の訓練データを用い [11]，AWS のインスタンス上で，敵対的摂動 [8] を用いて学習する場合，約 18 万円必要である．性能向上のためには，ハイパーパラメータの探索はもちろん，複数モデルを用いたアンサンブルを行うことも多々あり [12]，上記のような摂動を用いた学習では莫大な費用がかかってしまう．Strubell ら [13] や Schwartz ら [14] は近年のニューラルモデルの学習は計算量が増大しており，計算量に比して効率的な手法を探る必要を示唆した．例えば，Li ら [15] は定められた時間の中で，もっとも良いモデルを得る戦略を調査している．しかしながら，摂動を用いた学習については，注意が払われていない．

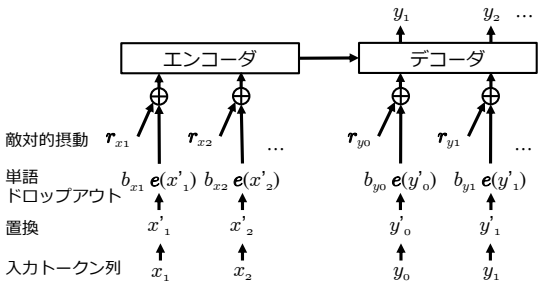


図 1 本研究で用いる摂動の概要．各摂動は独立なので，本図のように組み合わせることが可能．

そこで本研究では，摂動を用いたエンコーダ・デコーダの学習において，時間的に効率の良い手法を探る．単語ドロップアウト [16] やランダムに入力トークンを置換して摂動事例とする手法のように，計算量の小さな手法も含めて比較を行う．このような計算量の小さな手法は，既存研究においてはベースラインとして用いられていることもあるが [9]，多大な計算時間を要する手法 [9, 8] と比較して，短い時間で同等の性能を達成できる，すなわち，時間的効率が良いことが実験を通して明らかになった．

2 エンコーダ・デコーダ

本研究ではエンコーダ・デコーダを用いて，機械翻訳のような系列変換タスクに取り組む．本節では

エンコーダ・デコーダの定義を概説する．

系列変換タスクでは，エンコーダ・デコーダは入力系列に対応した系列を生成する．長さ I の入力トークン列および長さ J の出力トークン列をそれぞれ $\boldsymbol{x}_{1:I}$，$\boldsymbol{y}_{1:J}$ とすると，エンコーダ・デコーダは次の条件付き確率を計算する：

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{j=1}^{J+1} p(y_j|\boldsymbol{y}_{0:j-1}, \boldsymbol{X}), \tag{1}$$

ここで，y_0 および y_{J+1} はそれぞれ文頭，文末を示す特殊なトークンとし，$\boldsymbol{X} = \boldsymbol{x}_{1:I}$，$\boldsymbol{Y} = \boldsymbol{y}_{1:J+1}$ とする．

学習においては，訓練データにおける負の対数尤度を最小化するパラメータ $\boldsymbol{\theta}$ を探す．$\boldsymbol{X}_n$ と $\boldsymbol{Y}_n$ という，対応する系列のペアを含む訓練データを $\mathcal{D}$ とする，すなわち，$\mathcal{D} = \{(\boldsymbol{X}_n, \boldsymbol{Y}_n)\}_{n=1}^{|\mathcal{D}|}$ とすると，次の損失関数を最小化するように学習を行う．

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{D}} \log p(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\theta}). \tag{2}$$

3 摂動

本節では，本研究で用いる，置換，単語ドロップアウト，敵対的摂動の 3 種類の摂動を概説する．図 1 に示したように，これらの摂動は独立な手法なので，組み合わせることも可能であり，実際に，実験では置換と単語ドロップアウトの組み合わせも用いる．

3.1 置換

スケジュールドサンプリング [9] のように，本来の入力トークンをサンプリングしたトークンに置き換える手法を，本研究では置換と呼ぶ．置換では，入力系列 $\boldsymbol{X}$ 内のトークン x_i について，ある分布 Q_{x_i} からサンプリングしたトークン $\hat{x}_i$ と確率 α で置き換えることで，新たな入力系列 $\boldsymbol{X}'$ を構築する．

$$\hat{x}_i \sim Q_{x_i}, \tag{3}$$

$$x'_i = \begin{cases} x_i & \text{確率 } \alpha \\ \hat{x}_i & \text{確率 } 1-\alpha. \end{cases} \tag{4}$$

同様にして，$\boldsymbol{Y}$ から $\boldsymbol{Y}'$ を構築する．

Bengio ら [9] はカリキュラム学習の発想で，学習が進むにつれ，徐々に α が減少するような関数を用いた．この関数により，学習初期では正しい入力トークンを頻繁に用い，学習後期ではサンプリングしたトークンを頻繁に用いるようになる．本研究でも同様な関数を用いて α の計算を行う．

$$\alpha_t = \max\left(q, \frac{k}{k + \exp(\frac{t}{k})}\right) \tag{5}$$

ここで，q と k はハイパーパラメータである．上記の式により，α_t は 1 から q まで，学習ステップ t に依存して減少していく．この α_t を各ステップにおける α として用いる．

分布 Q_{x_i} については，条件付き確率，一様分布，類似度の 3 種類を用いる．

条件付き確率：Rep(SS)　Bengio ら [9] は訓練時と推論時における差異に対処するため，スケジュールドサンプリングを提案した．スケジュールドサンプリングは次の条件付き確率を Q_{y_i} として用いる．

$$p(\hat{y}_i|\boldsymbol{y}'_{0:i-1}, \boldsymbol{X}). \tag{6}$$

スケジュールドサンプリングはデコーダ側の摂動を計算する手法であり，Q_{x_i} に対応する関数はなく，エンコーダ側の入力については手を加えない．

元来のスケジュールドサンプリングはデコーダ側のトークン数分デコーダの前向き計算を行う必要があり，デコーダ側の入力系列長に比例した計算時間を要する．これに対処するため，Duckworth ら [17] はデコーダ側の全入力に対応する条件付き確率を 1 度に計算してしまう，並列化したスケジュールドサンプリングを提案した．この並列化したスケジュールドサンプリングは時間的効率が良いと報告されているため，本研究ではこれを用いる．

一様分布：Rep(Uni)　並列化したスケジュールドサンプリングを用いたとしても，式（6）を計算するために，最低 1 回はデコーダの前向き計算を行わなければならない．より時間的効率の良い摂動を探るために，本研究では一様分布と類似度という 2 種類の高速な摂動の計算手法との比較を行う．一様分布（Rep(Uni)）はその名の通り，各語彙における一様分布を Q_{x_i} や Q_{y_i} として用いる．例えばエンコーダ側の摂動事例を得るために，エンコーダ側の語彙からランダムに 1 つサンプリングし，式（4）の $\hat{x}_i$ として用いる．本手法は既存研究においてベースラインとしても用いられている手法である [9]．

類似度：Rep(Sim)　式（6）の条件付き確率は，元々の入力トークンに似たトークンに対して高い確率を付与していると考えられる．このため，一様分布よりも摂動として適したトークンを得る手法として，似たトークンに高い確率を付与している分布を考える．エンコーダ側の語彙を $\mathcal{V}_x$，エンコーダ側の埋

め込み表現（次元数は d_x）を $\boldsymbol{E}_x \in \mathbb{R}^{|\mathcal{V}_x| \times d_x}$，トークン x_i に対応する埋め込み表現を $\boldsymbol{e}(x_i)$ と表すとしたとき，次の確率分布を Q_{x_i} として用いる．

$$\mathrm{softmax}(\boldsymbol{E}_x \boldsymbol{e}(x_i)), \tag{7}$$

ここで，softmax(.) はソフトマックス関数である．すなわち，式（7）は $\boldsymbol{e}(x_i)$ に似た埋め込み表現に高い確率を付与する．言い換えれば，式（7）は文脈を考慮していない場合の，x_i の類似トークンに高い確率を付与する．デコーダ側についても，$\boldsymbol{e}(y_i)$ を用いて同様に計算する．

3.2 単語ドロップアウト：WDrop

単語ドロップアウトは入力トークン x_i について，ランダムに，埋め込み表現 $\boldsymbol{e}(x_i)$ の代わりに，ゼロベクトルを摂動として用いる [16]．

$$b_{x_i} \sim \mathrm{Bernoulli}(\beta), \tag{8}$$

$$\mathrm{WDrop}(x_i, b_{x_i}) = b_{x_i} \boldsymbol{e}(x_i), \tag{9}$$

ここで，Bernoulli(β) は確率 β で 1 を返し，それ以外に 0 を返す関数である．すなわち，$\mathrm{WDrop}(x_i, b_{x_i})$ は β で $\boldsymbol{e}(x_i)$ を，それ以外にゼロベクトルを返す関数である．式（9）を入力系列の各トークンに適用し，得られた系列を摂動事例として用いる．

3.3 敵対的摂動：Adv

Miyato ら [6] は埋め込み空間において敵対的摂動を計算する手法を提案した．この手法では，元々の入力トークンを別のものと置き換えるのではなく，入力トークンの埋め込み表現に敵対的摂動を加える．Sato ら [8] は本手法をエンコーダ・デコーダに適用し，性能向上を報告しており，本研究はこれにならって敵対的摂動を計算する．

敵対的摂動，すなわち，損失の値に深刻な悪影響を及ぼす摂動は損失関数 $\mathcal{L}(\boldsymbol{\theta})$ の勾配を元に計算する．入力トークン x_i についての敵対的摂動を $\boldsymbol{r}_{x_i} \in \mathbb{R}^{d_x}$ としたとき，敵対的摂動を適用した埋め込み表現 $\boldsymbol{e}'(x_i)$ は次式で計算する．

$$\boldsymbol{e}'(x_i) = \boldsymbol{e}(x_i) + \boldsymbol{r}_{x_i}, \tag{10}$$

$$\boldsymbol{r}_{x_i} = \epsilon \frac{\boldsymbol{c}_{x_i}}{||\boldsymbol{c}_{x_i}||}, \tag{11}$$

$$\boldsymbol{c}_{x_i} = \nabla_{\boldsymbol{e}(x_i)} \mathcal{L}(\boldsymbol{\theta}), \tag{12}$$

ここで，ϵ は敵対的摂動のノルムを調整するハイパーパラメータである．この式を入力系列のすべてのトークンに適用することで，摂動事例を得る．

3.4 学習

置換や単語ドロップアウトを用いた場合の学習としては，摂動事例から正しい系列を予測するよう学習する．例えば，置換を用いた際は，次の負の対数尤度を最小化する．

$$\begin{aligned}\mathcal{L}'(\boldsymbol{\theta}) &= -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \log p(\boldsymbol{Y}|\boldsymbol{X}', \boldsymbol{Y}'; \boldsymbol{\theta}), \\ &= -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \sum_{j=1}^{J+1} \log p(y_j|\boldsymbol{y}'_{0:j-1}, \boldsymbol{X}'; \boldsymbol{\theta}).\end{aligned} \tag{13}$$

敵対的摂動を用いた場合には，既存研究にならい [18, 8]，一度行った前向き計算を利用し，元の損失関数と敵対的摂動を用いて計算した損失 $\mathcal{A}(\boldsymbol{\theta})$ の合計を最小化する．

$$\mathcal{J}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \lambda \mathcal{A}(\boldsymbol{\theta}), \tag{14}$$

ここで，λ は 2 つの損失関数を釣り合わせるためのハイパーパラメータである．$\mathcal{A}(\boldsymbol{\theta})$ としては，元の入力に対する出力確率分布を正例とする．すなわち，

$$\mathcal{A}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \mathrm{KL}\left(p(\cdot|\boldsymbol{X}; \boldsymbol{\theta}) || p(\cdot|\boldsymbol{X}, \boldsymbol{r}; \boldsymbol{\theta})\right), \tag{15}$$

ここで，$\boldsymbol{r}$ は各入力トークンに対する敵対的摂動をつなぎ合わせたベクトルとし，KL(·||·) はカルバック・ライブラー情報量とする．

4 機械翻訳における実験

系列変換における代表的なタスクとして，翻訳タスクでの実験を行う[1]．広く使用されているデータセットとして，WMT 英-独データセットを用いる．具体的には，Ott ら [11] と同じ前処理を行った，450 万文対を訓練データとして用い，newstest2010-2016 で性能を評価する．性能評価には，SacreBLEU [19] を用いて BLEU を計算する．

エンコーダ・デコーダとしては Transformer [10] の big 設定を用いる．実装は fairseq を元に行った[2]．各ハイパーパラメータについては，$q = 0.9$，$k = 1000$，$\beta = 0.9$ とし，Adv については，既存研究 [8] と同じ値を使用した．学習ステップ数は全ての手法で 50,000 とした．

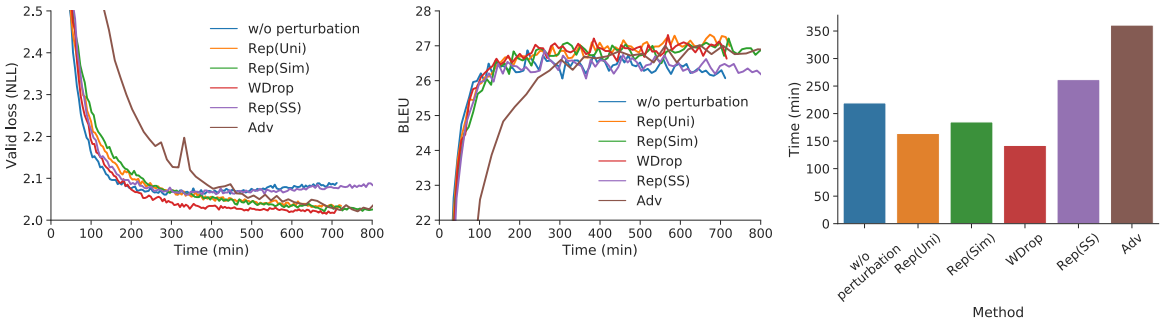
4.1 結果

摂動なしの Transformer（w/o perturbation）および各摂動を用いた際の BLEU 値を表 1 に示す．なお，

1） 要約，文法誤り訂正での実験については，付録に記した．
2） https://github.com/pytorch/fairseq

表 1　newstest2010-2016 での BLEU 値，それらの平均値，摂動なしの Transformer を基準とした際の各手法での計算速度．

手法	摂動位置	2010	2011	2012	2013	2014	2015	2016	平均	速度
摂動なし（w/o perturbation）	-	24.22	22.11	22.69	26.60	28.46	30.50	33.58	26.88	×1.00
Rep(Uni)	両方	24.75	22.68	23.32	27.01	28.89	**31.38**	34.94	**27.57**	×1.00
Rep(Sim)	両方	24.77	22.50	23.10	26.91	**28.98**	31.03	34.29	27.37	×0.92
WDrop	両方	**24.92**	22.71	**23.40**	27.11	28.73	30.99	34.80	27.52	×1.00
Rep(Uni)+WDrop	両方	24.82	22.82	23.38	**27.30**	28.56	30.65	35.02	27.51	×1.00
Rep(Sim)+WDrop	両方	24.83	**22.95**	**23.40**	27.23	28.65	30.88	**35.05**	**27.57**	×0.92
Rep(SS)	デコーダ	24.44	21.97	22.74	26.77	28.44	30.83	33.71	26.99	×0.87
Adv	両方	24.71	22.60	23.23	26.98	28.97	30.49	34.40	27.34	×0.33



(a) NLL　　(b) BLEU　　(c) 摂動なしの Transformer の BLEU値に到達するまでの時間

図 2　newstest 2013 での負の対数尤度（NLL），BLEU 値，摂動なしの Transformer の BLEU 値に到達するまでの時間．

表 1 に記したように，デコーダ側の摂動計算手法である Rep(SS) 以外はエンコーダ側，デコーダ側の両方に摂動を適用した．BLEU 値に加えて，表 1 は摂動なしの Transformer を基準とし，各摂動を用いた際の計算速度を示している．また，高速な手法である WDrop，Rep(Uni)，Rep(Sim) については，組み合わせた際の結果も記した．

まず，摂動なしの Transformer は newstest 2014 において，28.46 と既存研究 [10] と同程度の BLEU 値を達成しており，強いベースラインとみなして良いと考えられる．BLEU の平均値において，摂動を用いた全ての手法が摂動なしの場合よりも高い値を達成しており，摂動を用いることで，エンコーダ・デコーダの性能が向上することが分かる．加えて，素朴な手法である WDrop，Rep(Uni)，Rep(Sim)，およびその組み合わせは，既存研究で優れているとされていた Adv や Rep(SS) に対し [9, 8]，計算速度も速く，BLEU 値でも上回っている．

図 2 は訓練時間に対する，各手法での newstest 2013 における負の対数尤度（a），BLEU 値（b），および，摂動なしの Transformer の学習修了時の BLEU 値に到達するまでの時間（c）を示している．図（a）（b）より，学習の早い段階で，WDrop，Rep(Uni)，Rep(Sim) は摂動なしの Transformer よりも良い負の対数尤度と BLEU 値に到達していることが分かる．さらに，図の（c）から，これらの手法は摂動なしの Transformer 最終的な BLEU 値を，Rep(SS) や Adv，摂動なしの場合よりも高速に達成していることが分かり，特に，Adv よりも 2 倍高速である．これらの結果から，WDrop，Rep(Uni)，Rep(Sim) およびその組み合わせは，時間的効率の良い摂動と言える．

5　おわりに

本研究では，エンコーダ・デコーダに様々な摂動を適用し，系列変換タスクにおいて，時間的効率の良い摂動を調査した．既存研究では，スケジュールドサンプリングや敵対的摂動の有用性が報告されていたが [9, 20, 8]，本研究の設定においては，単語ドロップアウト [16] のような単純な手法がこれらと同等の性能を高速に達成可能であることを示した．この結果から，エンコーダ・デコーダに摂動を適用する際は，まずは単純な手法を試すことを勧めたい．実際に頑健なモデルを構築するため，摂動を選択する際に，また，これからの摂動に関する研究の実験的な比較において，本研究が一助となれば幸いである．

謝辞　本研究は JSPS 科研費 JP18K18119 の助成を受けたものです.

参考文献

[1]Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pp. 3104–3112, 2014.

[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 (EMNLP)*, pp. 379–389, 2015.

[3]Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 753–762, 2017.

[4]Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, Vol. 15, No. 56, pp. 1929–1958, 2014.

[5]Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[6]Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[7]Yong Cheng, Lu Jiang, and Wolfgang Macherey. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4324–4333, 2019.

[8]Motoki Sato, Jun Suzuki, and Shun Kiyono. Effective adversarial regularization for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 204–210, 2019.

[9]Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems 28 (NIPS)*, pp. 1171–1179. 2015.

[10]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 (NIPS)*, pp. 5998–6008. 2017.

[11]Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*, pp. 1–9, 2018.

[12]Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (WMT)*, pp. 1–61, 2019.

[13]Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3645–3650, 2019.

[14]Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *CoRR*, Vol. abs/1907.10597, , 2019.

[15]Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E. Gonzalez. Train large, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 11432–11442, 2020.

[16]Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. 1019–1027. 2016.

[17]Daniel Duckworth, Arvind Neelakantan, Ben Goodrich, Lukasz Kaiser, and Samy Bengio. Parallel scheduled sampling. *CoRR*, 2019.

[18]Takeru Miyato, Shin ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

[19]Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation (WMT)*, pp. 186–191, 2018.

[20]Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4334–4343, 2019.

[21]Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui. Massive exploration of pseudo data for grammatical error correction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp. 2134–2145, 2020.

[22]Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 52–75, 2019.

[23]Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL)*, pp. 1–14, 2014.

[24]Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pp. 825–835, 2016.

[25]Daniel Dahlmeier and Hwee Tou Ng. Better Evaluation for Grammatical Error Correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 568–572, 2012.

[26]Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13063–13075, 2019.

[27]Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning (ICML)*, pp. 5926–5936, 2019.

[28]Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[29]Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2401–2410.

[30]Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pp. 95–100, 2012.

[31]Sho Takase and Naoaki Okazaki. Positional encoding to control output sequence length. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 3999–4004, 2019.

[32]Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9054–9065, 2019.

表 2　各手法の $F_{0.5}$．また，摂動なしと同一の設定として，Kiyono ら [21] の報告値を掲載した．

手法	摂動位置	開発	テスト	CoNLL
摂動なし	-	47.25	64.74	61.62
Rep(Uni)	両方	47.77	64.67	62.22
Rep(Sim)	両方	47.58	64.51	62.29
WDrop	両方	48.53	65.47	62.22
Rep(Uni)+WDrop	両方	48.58	65.94	**62.33**
Rep(Sim)+WDrop	両方	**48.72**	**65.97**	62.29
Rep(SS)	デコーダ	47.84	65.18	62.30
Adv	両方	48.17	65.90	62.23
Kiyono ら [21]	-	-	65.0	62.2

A　文法誤り訂正における実験

系列変換タスクとして，文法誤り訂正と生成的要約タスクでの実験を行う．文法誤り訂正の評価データとしては，広く用いられている，BEA データセットの開発データとテストデータ [22]，および，CoNLL-2014 のテストデータ（CoNLL） [23] を用いる．BEA については ERRANT [24] で計算した $F_{0.5}$，CoNLL については M^2 [25] を評価に用いる．

既存研究 [21] にならい，疑似パラレルコーパスでエンコーダ・デコーダ（Transformer）を学習した後，摂動を用いたファインチューニングを BEA の訓練データ上で行う．各ハイパーパラメータについては，4 節と同様の値を用いた．

A.1　結果

表 2 に各手法での結果を示す．これらのスコアは，5 つの異なるランダムシードを用いてファインチューニングを行い，その平均値である．加えて，本表には摂動なしと同一の設定として，Kiyono ら [21] の報告値を記している[3)]．

表 2 より，BEA テストデータ（テスト）における Rep(Uni) と Rep(Sim) を除き，摂動を用いる全ての手法が摂動なしのスコアを上回っていることが分かる．機械翻訳における実験と同様に，WDrop，およびこれと Rep(Uni)，Rep(Sim) の組み合わせが Adv と同等の性能を達成している．これらの手法は Rep(SS) や Adv よりも計算が高速であることから，文法誤り訂正タスクにおいても，時間的な効率の良い手法であると言える．

B　生成的要約における実験

生成的要約の評価データとして，広く用いられている，Annotated English Gigaword から抽出された，1951 の文と要約（正確には見出し文）のペアからなるテストデータ [30, 2] を用いる．

3)　表 2 の摂動なしの結果は Kiyono ら [21] と比べて僅かに低いが，これは学習におけるランダム性によるものと考えられる．

表 3　Annotated English Gigaword テストデータにおける，各手法の ROUGE-1，2，L の F_1 値（R-1，R-2，R-L）．本表の下の部分は近年の研究における報告値である．

手法	摂動位置	R-1	R-2	R-L
摂動なし	-	39.20	19.84	36.21
Rep(Uni)	両方	**39.81**	20.40	**36.93**
Rep(Sim)	両方	39.70	20.14	36.77
WDrop	両方	39.66	**20.45**	36.59
Rep(Uni)+WDrop	両方	39.36	20.13	36.62
Rep(Sim)+WDrop	両方	39.56	20.14	36.66
Rep(SS)	デコーダ	39.20	20.04	36.27
Dong ら [26]	-	38.45	19.45	35.75
Song ら [27]	-	38.73	19.71	35.96
Zhang ら [28]	-	39.12	19.86	36.24
Qi ら [29]	-	39.51	20.42	36.69

訓練データとしては，既存研究 [2, 31] にならい，Annotated English Gigaword から抽出した 380 万の文と要約のペアを用いる．また，近年の生成的要約タスクでは，大規模コーパスで事前学習したエンコーダ・デコーダを用いることも多く [26, 27, 28, 29]，これらの研究にならい，REALNEWS [32] と News Crawl [12] から文と要約のペアを抽出し，訓練データとした．合計で，1700 万の文と要約ペアを訓練データに用いた．機械翻訳，文法誤り訂正と同様，エンコーダ・デコーダには Transformer [10] を用い，ハイパーパラメータについては 4 節と同じ値を用いた．なお，敵対的摂動については，勾配が爆発してしまい，学習を安定させることができなかった．ランダムシードを変更して何度か学習を行ったが全て発散してしまい，また，ハイパーパラメータの探索を行うことも予算の都合上難しかったため，本実験においては，敵対的摂動は除くこととする．

B.1　結果

表 3 に Annotated English Gigaword テストデータにおける，各手法の ROUGE-1，2，L の F_1 値，および，近年の研究における報告値を示した [26, 27, 28, 29]．表 3 より，機械翻訳や文法誤り訂正での結果と同じく，摂動を用いた全ての手法が摂動なしのスコアを上回っていることが分かる．WDrop，Rep(Uni)，Rep(Sim)，およびその組み合わせは Rep(SS) よりも高い ROUGE 値を達成している．これらは計算速度も高速であることから，生成的要約タスクにおいても，時間的効率の良い手法であることが伺える．加えて，Rep(Uni) と WDrop はそれぞれ，ROUGE-1 と L，ROUGE-2 において，現状のトップスコア [29] よりも高い値を達成している．