無関連パラレルコーパスを用いた平易文生成

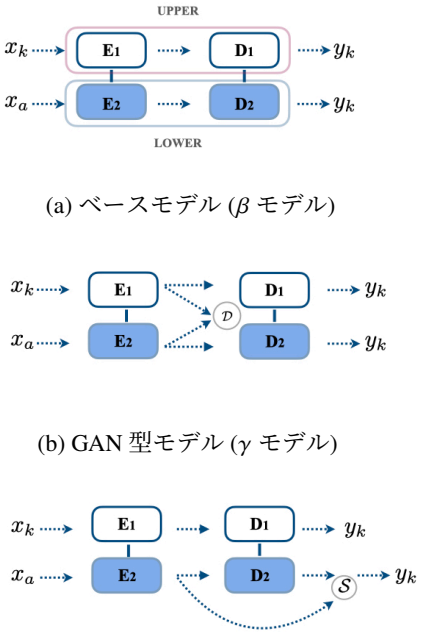
野本忠司
国文学研究資料館
nomoto@acm.org

1 はじめに

本稿では，二つの互いに意味的関連性のないコーパスを用いて，平易文を生成することを考える．文の平易化について大きな問題となるのが大量の教師データの確保である．人手によるアノテーションを避けるため，これまで字幕，ウィキペディアなどの既存オンラインデータを駆使して学習データを「鋳造」する動きが主流であったが，近年ターゲットコーパスを用いることなく平易化を目指すアプローチに徐々にも注目が集まるようになってきた．本研究もこの流れに沿うものであるが特に本稿に直接関連するものとしては [1] がある．この研究では平易化学習に用いるパラレルコーパスのうちターゲットコーパスをソースコーパスから機械的な手続き（類似度判定）により「合成」しても十分な有効性があることを示した．当時平易化用ターゲットコーパスはそれと明示的に意図して作成されたものなくてはならないという暗黙のコンセンサスがあったため，この結果は研究者に驚きをもって迎えられた．しかし [1] のアプローチが成立するには，原文とターゲットの間に十分な類似性が確保されていることが条件となる．この条件が崩れると同手法は有効性を失う．さらに関連した問題として計算コストの問題ある．N 文から成るソースコーパスから仮想ターゲットを作るためには $(N^2 - N)/2$ 回の比較計算が必要となる．多くのコーパスのサイズは数十万〜数百万文に達しておりターゲットコーパス合成の労力は非常に大きいものとなる．このような背景のもと本稿ではこのような類似性に関する条件を導入せず，平易文生成が可能かどうか検討する．

2 双対ネットワーク

本稿では二つの同型のエンコーダ・デコーダーネットワークを重ね合わせた双対モデルを導入する．エンコーダー，デコーダーはいずれも LSTM とする．本モデルの目標は平易文の属性を抽出して複雑な文に転写し後者を平易化することにある．特に本稿では図 1 に示す三つのアプローチを検討する．x_k は平易文，x_a は複雑文とする．y_k はターゲットを表す．ただし $x_k = y_k$ とする．また x_a, y_k の類似度（後述）はゼロ近くに保つ．双対ネットワークの上位部分はオートエンコーダー（図 1a の ‘UPPER’）下位部分 (図中 ‘LOWER’で示した箇所) は翻訳モデルとして機能する．上位，下位ネットワークはパラメーターをすべて共有する．モデルはすべて Fairseq ライブラリーを用いて構築した．[1)]



(a) ベースモデル (β モデル)

(b) GAN 型モデル (γ モデル)

(c) 類似度参照型モデル (σ モデル)

図 1: 双対ネットワーク

さらに平易文から複雑文への属性の転写を促すために二つの手法を導入する．ひとつは GAN を用いた方法もうひとつは類似度ロスを用いた方法である．特に GAN については Jensen-Shannon 法と Wasserstein 法について検討する．まず GAN について説明する．GAN は一般に弁別器を用いて目標か

1) https://github.com/pytorch/fairseq.git

表 1: JS-GAN と W-GAN. $E_1(x)$ はエンコーダー E_1 の出力．$E_2(x)$ も同様． G_p は Gradient Penalty[2]． λ はハイパーパラメーター．

	弁別ロス	生成ロス
JS-GAN	$\mathbb{E}[\log D(E_1(x))] + \mathbb{E}[\log(1 - D(E_2(x))]$	$\mathbb{E}[\log D(E_2(x))]$
W-GAN	$\mathbb{E}[f(E_1(x))] - \mathbb{E}[f(E_2(x))] + \lambda G_p$	$\mathbb{E}[f(E_2(x))]$

らのズレの情報を獲得し生成器に流すことで出力を目標に近づける．Jensen-Shannon 法 (JS-GAN) と Wasserstein 法 (W-GAN) の大きな違いは目標とのズレを測る方法の違いに由来する．JS-GAN ではシグモイド関数を用い W-GAN は Earth Mover's Distance と呼ばれる距離を用いる．GAN では通常二種類のロス関数を用いる．ひとつは弁別ロスであり，もうひとつは生成ロスである．それぞれ弁別器，生成器の訓練に用いられる．表 1 に JS-GAN と W-GAN の弁別ロス，生成ロスをまとめる．本稿では f, D は以下で定義されるものとする．

$$D(H(x)) = \textbf{sigmoid}(f(H(x))), \tag{1}$$

$$f(H(x)) = w^\top \textbf{relu}(f_c(H(x))) \tag{2}$$

f は全結合層，f_c は畳み込み層を表す．$H(x)$ は x を入力とする任意のネットワーク．本稿では特に GAN をエンコーダーの出力に適用する．これは複雑文の出力表現を平易文のそれに近づけることで，後者の属性が前者に転写されるのを狙っているためである．JS-GAN は W-GAN に比べて逆伝播される情報の粒度が荒くなる．本件ではこの違いが平易化に影響するのか否か注目する．

類似度参照型モデルは図 1 のベースモデルに原文とターゲットの意味を近づけるよう下位ネットワークに類似度ロスを備える．これは以下のように与える．$E_2(x), D_2(x)$ は E_2,D_2 の出力（図 1 参照）．

$$\mathcal{S} = 1 - \cos(E_2(x_a), D_2(E_2(x_a))). \tag{3}$$

最後に，図 1 のすべてのモデルについて以下のロスを付加する．

$$\begin{aligned}\mathcal{L}_{rec} = &\mathbb{E}_{x\sim P_k}[-\log P_{D_1}(y \mid E_1(x))] \\ &+ \mathbb{E}_{x\sim P_a}[-\log P_{D_2}(y \mid E_2(x))].\end{aligned} \tag{4}$$

なお，表 2 に各モデルで用いるロス関数をまとめた．

3 入出力交替型オートエンコーダー

一般に平易文とその原文は意味的，構文的にかなり類似している．本稿ではこの性質をさらに強くモデルに反映させるため，下位ネットワーク (LOWER) において，ターゲットを入力とランダムに入れ替えることを考える．具体的には下位ネットワークの y_k と x_a をある確率で交換する．ターゲットが x_a に置き換わったモデルは通常のオートエンコーダーと同じ振る舞いをする．このようなモデルを「入出力交替型オートエンコーダー」ないしは FFA (Flip-Flop Auto-Encoder) と呼ぶことにする．

表 2: 双対ネットワークにおけるロス関数

β モデル	$\mathcal{L}_{rec}$
γ モデル (JS-GAN)	$\mathcal{L}_{rec} + \mathbb{E}[\log D(E_2(x))]$
γ モデル (W-GAN)	$\mathcal{L}_{rec} + \mathbb{E}[f(E_2(x))]$
σ モデル	$\mathcal{L}_{rec} + \mathbb{E}[1 - \cos(E_2(x), D_2(E_2(x))]$

4 実験

実験では [3] が公開している原文・ターゲット無対応のコーパス (TSDATA) をベースにする（Wikipedia 由来）．[2] TSDATA は Flesch-Kincaid や Flesch Reading Ease 等一般的な平易化指標において原文がターゲットに対して難度が高くなるよう設計されている．今回このデータの前半 100 万対を取り出し学習用データとする．これを OOA-TRAIN と呼ぶ．テストデータとしては平易化研究でスタンダードでなりつつある， [4] がクラウドソースして構成したテストデータを用いる（以降 TURK-TEST）．ベースは Simple Wikipedia で 359 の原文・ターゲット対でそれぞれ人手による別解が 8 つ付随する．また参考のため [4] と [1] が用いた有対応学習データ (WikiLarge, SSCORPUS) のプロファイルを表 4 に載せる．SSCORPUS[3] と WikiLarge[4] は前者が類似度指標 (MAS) を用いて Wikipedia から学習データを自動構築しているのに対して後者は Wikipedia と Simple Wikipedia を対応づけ人手でデータ構築を行っている点で大きく異なる．

表 4 から以下のような指摘が可能である．まず TSDATA, OOA-TRAIN は類似性がゼロに近く FGL, FRE も大差なくほぼ同じと見なせる．一方

2) https://github.com/subramanyamdvss/UnsupNTS.git
3) https://github.com/tmu-nlp/sscorpus.git
4) https://github.com/XingxingZhang/dress.git

表 3: 本稿で用いる無対応の学習データの例

原文	ターゲット
six centuries later the king of kotte , veera parakramabahu viii (14771496) , had a network of canals constructed connecting outlying villages with colombo and negombo lagoon so that produce such as arecanuts , cloves , cardamom , pepper and cinnamon , could be more easily transported to the kingdoms main seaport at negombo .	at the top of the head there were five small holes , through which food would be ingested and waste products discharged .
in 1874 , he was nominated by the liberal-conservative convention as local candidate for the county of cumberland .	he scored 107 goals in 429 league games in a 17-year career in the football league and scottish football league .

表 4: 実験で用いるデータセット． TOK: 一文あたり平均語数. FRE: Flesch Reading Ease. FGL: Flesch-Kincaid Grade Level . SIM: 原文・ターゲットのコサイン類似度. SIZE: 原文・ターゲット対の総数． ALG は対応の有無を示す.

	TSDATA		OOA-TRAIN		TURK-TEST		WIKILARGE		SSCORPUS	
	原文	ターゲット	原文	ターゲット	原文	ターゲット	原文	ターゲット	原文	ターゲット
TOK	34.30	18.89	34.39	18.44	22.61	22.15	25.11	18.46	25.26	17.95
FRE	32.91	83.95	32.15	85.43	67.55	77.22	63.79	74.79	62.26	74.52
FGL	17.37	6.41	17.65	6.09	9.51	8.03	10.82	7.71	10.87	7.35
SIM	0.0032		0.0030		0.7767		0.5731		0.6679	
SIZE	2,000,000		1,000,000		359		296,402		492,993	
ALG	NO		NO		YES		YES		YES	

SSCORPUS は WikiLarge に極めて近い．原文・ターゲットのコサイン値は WikiLarge のそれを超える．また FGL/FRE もほぼ同じと言ってよい．このことから SSCORPUS と WikiLarge は作り方の違いはあるもののトークン数，FGL/FRE, 原文とターゲットの意味的近さ (SIM) において同一と見なせる．TURK-TEST は WikiLarge をベースにしているためプロファイルは後者のものを引き継いでいる．なおコサイン値 (SIM) はストップワードを排し SciPy を用いて計算した．[5] また本稿では開発データとして WikiLarge パッケージに同包されている Dev データを利用した（992 ペア）．つまり本稿の β, γ, σ 各モデルは学習データとして OOA-TRAIN, 開発データとして WikiLarge，テストデータとして TURK-TEST を用いていることになる．

結果を表 6 に示す．FFA による入出力交替確率は 0.2 とした．ここで DIFF とは入力と出力の単語数の差の平均を示す．例えば DIFF=2 は出力が入力に比べて文の長さが 2 単語分短いことを意味する．SARI は [5] が発案した出力の多様性（入力といかに違うか）を示す指標．上向き矢印↑は当該指標の値が高いと良く，下向き↓は小さいと良いことを示す．γ モデル (J) は JS-GAN を，γ モデル (W) は W-GAN を導入した γ 型モデルを表す．比較のため現在 SOTA として知られているモデル (UNTS) [3] の（現実験環境下で再現した）結果を併記した．UNTS は RNN 型の多言語同時翻訳モデルをベースにしている．本稿アプローチ，特に γ モデル (J) が主要指標である BLEU，SARI において UNTS を凌いでいることが確認できる．

さらに本稿とは異なったアプローチで教師なし平易文生成を実現している 2 つモデルとの比較を行った（表 7 ）．いずれもテストは TURK-TEST を用いた．ひとつは MAS と呼ばれる類似度指標を用いて原文・ターゲットのアライメントを行いフレーズ統計翻訳システム (MOSES)[6] で学習させるアプローチ [1]（表では PBSMT），もうひとつは LIGHTLS と呼ばれる GloVe で学習済みの単語埋め込みから近い単語を見つけ原文の単語と置き換え平易化する一種の辞書引き法である [6]（表中では LLS）．[7] いずれも γ モデル (J) を SARI において凌いでいるものの BLEU，DIFF において劣っている．

これが具体的どのような出力の違いとして現れる

5） https://www.scipy.org/.

6） http://www.statmt.org/moses/

7） https://github.com/codogogo/lightls.git

表 5: 生成例

(1)	原文	in architectural decoration small pieces of colored and iridescent shell have been used to create mosaics and inlays , which have been used to decorate walls , furniture and boxes .
	γ モデル (J)	in architectural decoration small pieces of colored and iridescent shell have been used to create mosaics and inlays .
	PBSMT	in architectural decoration , small pieces of colored and iridescent shell have been used to create various and imitating , which have been used to decorate walls , furniture and boxes .
	ターゲット	small pieces of colored shell and iridescent shell have been used to create mosaics and inlays which have been used to decorate larger items such as boxes and furniture .
(2)	原文	aside from this , cameron has often worked in christian-themed productions , among them the post-rapture films left behind : the movie , left behind ii : tribulation force , and left behind : world at war , in which he plays cameron " buck " williams .
	γ モデル (J)	aside from this , cameron has often worked in christian-themed productions .
	PBSMT	cameron has often worked in christianity-related movies , among them the post-rapture movies left behind : the movie , left behind ii : establish force , and left behind : world at war , in which he plays cameron “ buck " " williams .
	ターゲット	cameron has also often worked in christianity-related movies , among them the post-rapture movies left behind : the movie , left behind ii : tribulation force , and left behind : world at war , in which he plays cameron " buck " williams .
(3)	原文	in return , rollo swore fealty to charles , converted to christianity , and undertook to defend the northern region of france against the incursions of other viking groups .
	γ モデル (J)	in return , rollo swore fealty to charles , converted to christianity .
	PBSMT	in return , rollo swore fealty to charles , converted to christianity , and undertook to defend the northern region of france against raids by other viking groups .
	ターゲット	in return , rollo swore fealty to charles , converted to christianity , and swore to defend the northern region of france against raids by other viking groups .

表 6: 実験結果

	BLEU↑	SARI↑	FGL↓	FRE↑	DIFF↑
β モデル	**0.970**	0.292	13.32	56.62	0.31
γ モデル (J)	0.880	**0.344**	**8.11**	70.91	**4.75**
γ モデル (W)	0.919	0.316	11.26	63.08	1.89
σ モデル	0.955	0.305	12.32	59.56	0.77
UNTS	0.356	0.307	7.54	84.09	9.26

表 7: コーパスに厳密な意味的対応を必要としない既存モデル

	BLEU↑	SARI↑	FGL↓	FRE↑	DIFF↑
PBSMT	0.896	0.352	9.15	70.06	0.11
LLS	0.738	0.349	8.93	71.62	0.19
γ モデル (J)	0.880	0.344	8.11	70.91	4.75

のか，いくつかの生成例を用いて説明する（表 5）．なお今回紙面の都合上 LLS は割愛する．表 5 が示すように PBSMT の出力は基本的に単語レベルの置き換えが中心になっている（(2)（3）の ‘movies’, ‘raids’など）． 一方 γ モデル (J) では構文上の大きな切り詰めが起きている．このように評価上の数字では見えないが，両者の出力スタイルに大きな違いが存在する．理由としては γ モデル (J) は学習データにおいて原文・ターゲットに意味的対応関係が存在せず，単語置き換えのパターンの発見には至らなかったのではないかと想像される．

5 おわりに

以上，本件では二つの意味的関連性がないコーパスを用いた平易文生成について検討した．結果，本件が提案する JS GAN を導入した双対ネットワークが既知手法と比し優れていることを確認した．教師を用いない平易化はモデルを工夫するか，データを工夫するかのいずれかになるが，本件では前者のアプローチを提案した．データを工夫するやり方は [1] が代表的であるが実現には大規模な文同士の比較を要しコストが大きい．本稿ではこのような手続きを経ず平易化ができるのかという問題を設定し，一応の見通しを得た．本件では紙面の都合上 FFA の重要性について触れなかったが単に GAN を使えば済むというものではないことを実験的に確認していることを付記しておく．

参考文献

[1]Tomoyuki Kajiwara. *Text Simplification without Simplified Corpora*. PhD thesis, Tokyo Metropolitan University, March 2018.

[2]Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, Vol. abs/1704.00028, , 2017.

[3]Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2058–2068, Florence, Italy, July 2019. Association for Computational Linguistics.

[4]Xingxing Zhang and Mirella Lapata. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 584–594, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[5]Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 401–415, 2016.

[6]Goran Glavaš and Sanja Štajner. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 63–68, Beijing, China, July 2015. Association for Computational Linguistics.