

# TULIP: Web 小説を学習に用いた三段階 LSTM による台本形式小説 (SS) 生成

TULIP: Generation of Screenplay-style Novel (SS) Using Three-step LSTM Trained on Web Novels

鈴木 惇<sup>\*1</sup>  
Atsushi Suzuki

<sup>\*1</sup>東京大学 大学院情報理工学系研究科

Graduate School of Information Science and Technology, The University of Tokyo

This paper proposes a new method to generate screenplay-style novels (SS: side story or short story), mainly written on the web. The key idea is to incorporate each talker's utterance-style into existing neural conversation model. Our model, three-step unified LSTM interlocation producer (TULIP), consists of unified three LSTMs: utterance-encoding LSTM, context-updating LSTM, and utterance-decoding LSTM. We trained our model on screenplay-style novels on the web, and generated new novels from scratch, in other words, not by direct processing of existing novels. We evaluated novels generated by our model both in quality and quantity, and confirmed that our LSTM based method generated a natural Japanese sentence sequence. We also indicate the limitation of existing quantitative evaluation methods.

## 1. 序論

本研究は、主に Web 上で投稿されている、台本形式小説 (SS: ショートストーリーもしくはサイドストーリーの略とされる) を自動生成することを目的とする。台本形式小説とは、登場人物の発話のみにより構成され、かつ各々の発話の発話者が明示された小説である。図 1 に一例<sup>\*1</sup>を示す。台本形式小説は、匿名掲示板に多く投稿され、その一部は編集されて複数の「まとめサイト」に複製保管される。まとめサイトは一般的に広告収入をもとに運営されており [Kashiwabara 12]、台本形式の小説の自動生成は、それ自体が娯楽の提供として意義があるだけでなく、広告収入源としての需要も考えられる。台本形式小説生成は、ゲームのセリフ制作の補助も可能であり、その点からも需要が考えられる。

一般的な小説を自動生成する試みは、従来、元となる小説や文をベースに一部を加工することで新たな小説を生成するという方法で行われてきた [Takagi 14] [Jain 17]。しかしながらこの方法では、新たに生成される小説の枠組みが元となる小説に大幅に制約される。また、一般的な小説を扱う場合、元となる小説の各発話の発話者を特定する必要があり、その性能に出力の質が依存する。さらに、出力した小説の扱い方によっては、元となる小説の著作権を侵害する可能性が非常に高い。このような事情から、小説を生成する際は、特定の小説をベースとするのではなく、一から行われることが望ましい。

他方、台本形式小説は、登場人物の対話の連続ととらえることができる。従って、対話自動生成の手法を台本形式小説生成に応用することが考えられる。対話の自動生成の技術は、近年、深層学習の発展に伴い、急速に向上している [Serban 16] [Ouchi 16] [Li 16]。これらは必ずしも長い文脈や登場人物の口調の一貫性を考慮したものではないが、これらを拡張することで、質の高い台本形式小説を一から自動生成することが可能になると考えられる。

このような背景の下、本研究では、深層学習を用いた対話自動

生成を、登場人物の口調を考慮したモデルに拡張し、台本形式小説生成用のモデル three-step unified LSTM interlocation producer (TULIP) を提案する。本研究のモデルは、投稿された台本形式小説を学習した後、生成段階では学習データを直接用いず、学習されたパラメータのみを用いて新たな小説を生成する。従って、本研究は著作権を侵害しない小説生成の可能性を開く。

本研究はさらに、生成された小説の質を、出力文の精査による定性的評価のほか、翻訳で用いられる定量評価指標で評価を行い、性能を示す。さらに評価指標の問題点にも触れる。

```

早苗「楓ちゃん〜ん？」
楓「はい」
早苗「言いたいことはそれだけかしら？」
楓「そうですね……まるで洗いたて、干したてのタオルケットを触っているようでした」
早苗「どうも反省してないみたいね」
あい「早苗さん、落着こうか、振り上げたその一升瓶を一度下ろして」
楓「私の命はあいちゃんに託されているからね？」
あい「それは楓さんの態度次第さ」
楓「あら、そうなの？」
あい「……いま一度、自分が置かれている状況を把握するべきだよ」

```

図 1 台本形式小説の一例: 台本形式小説では、すべての発話の発話者が明示されているのが特徴である。

## 2. 手法: Three-step Unified LSTM Interlocation Producer (TULIP)

### 2.1 概要

本節では本研究で提案するモデル three-step unified LSTM interlocation producer (TULIP) の構造を記述する。TULIP は、Long Short Time Memory (LSTM) [Hochreiter 97] を用いるものであり、発話符号化 LSTM と文脈更新 LSTM、そして発話復号化 LSTM の三段階の LSTM を用いて、次の単語、および話者を再帰的に生成する。TULIP は、Hierarchical recurrent encoder-decoder (HRED) [Sordoni 15], [Serban 16] に話者の概念を加え拡張したものにとらえることができる。

### 2.2 記法

準備として、 $N$  次元確率ベクトルの集合  $\mathcal{P}_{(N)}$  を  $\mathcal{P}_{(N)} \stackrel{\text{def}}{=} \left\{ \mathbf{p} \mid \mathbf{p} \in [0, 1]^N, \sum_{n=1}^N [p]_n = 1 \right\}$  で定義する。ただし、ここで、数ベクトル  $\mathbf{v}$  に対し  $[v]_n$  は  $\mathbf{v}$  の  $n$  番目の要素を表す。 $N$  次元 onehot ベクトルの集合を  $\mathcal{E}_{(N)} \subset \mathcal{P}_{(N)}$

連絡先: 鈴木 惇: 東京大学 atsushi.suzuki.rd@gmail.com

<sup>\*1</sup> 楓「藍子ちゃんのお腹を触っていたら夜になっていた」  
<http://ex14.vip2ch.com/test/read.cgi/news4ssnip/1463151146/>

を  $\mathcal{E}_{(N)} \stackrel{\text{def}}{=} \{e \mid e \in \{0, 1\}^N, \sum_{n=1}^N [e]_n = 1\}$  で定義する。ただし、ここで、数ベクトル  $v$  に対し  $[v]_n$  は  $v$  の  $n$  番目の要素を表す。また、 $e_{(N,n)} \in \mathcal{E}_{(N)}$  を第  $n$  要素が 1、それ以外が 0 となる onehot ベクトルとする。さらに、softmax 関数  $a_{\text{SM}} : \mathbb{R}^N \rightarrow \mathcal{P}_{(N)}$  を  $[a_{\text{SM}}(\mathbf{x})]_n \stackrel{\text{def}}{=} \frac{\exp(x_n)}{\sum_{n'=1}^N \exp(x_{n'})}$  で、cross entropy  $l_{\text{CE}} : \mathcal{E}_{(N)} \times \mathcal{P}_{(N)} \rightarrow \mathbb{R}$  を  $l_{\text{CE}}(e, \mathbf{p}) \stackrel{\text{def}}{=} -\sum_{n=1}^N [e_n \log p_n + (1 - e_n) \log(1 - p_n)]$  で定義する。

最初に、空白を表す特殊単語 [pad]、発話の始まりを表す特殊単語 [go]、未知語を表す特殊単語を用意し [unk]、それぞれ 1, 2, 3 と番号を振る。また、出現頻度上位  $I - 3$  位までの単語に 4, 5, ...,  $I$  と番号を振る。そして、 $i$  番の単語を  $e_{(I,i)}$  と同一視する。出現頻度  $I - 2$  位以下の単語は  $e_{(I,3)}$  と同一視する。

発話者についても 1, 2, ...,  $M$  と番号を振り、 $m$  番の発話者を  $e_{(M,m)}$  と同一視する。学習または出力する小説の発話数を  $L$  とする。 $\ell$  番目の発話 ( $m = 1, 2, \dots, M$ ) の発話者を  $e_{\ell}^{(t)} \in \mathcal{E}_{(M)}$ 、 $\ell$  番目の発話内容の  $j$  番目の単語を  $e_{\ell,j}^{(w)} \in \mathcal{E}_{(I)}$  とする。この記法の下、 $\ell$  番目の発話は発話者  $e_{\ell}^{(t)}$  と発話内容  $\mathbf{E}_{\ell}^{(w)} \stackrel{\text{def}}{=} [e_{\ell,1}^{(w)} \ e_{\ell,2}^{(w)} \ \dots \ e_{\ell,J}^{(w)}]$  の組  $(e_{\ell}^{(t)}, \mathbf{E}_{\ell}^{(w)})$  で表現される。なお、元の発話文の長さ  $J'$  は一般に  $J$  に満たないため、 $J' + 1$  番目から  $J$  番目まで [pad] を挿入する。

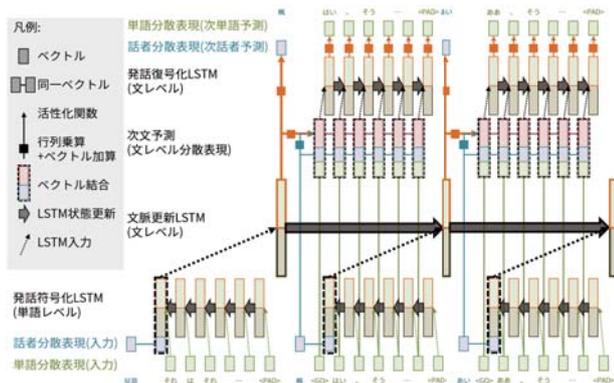


図2 TULIP の概要: 下部が発話符号化 LSTM, 中部が文脈更新 LSTM, 上部が発話復号化 LSTM である。

## 2.3 モデル

### 2.3.1 入出力関係

本研究で用いるモデルは以下の入出力関係を可能にする。

#### a) 完全な発話列から次の発話者の出力

1 番目から  $L'$  番目までの完全な発話の列  $(e_1^{(t)}, \mathbf{E}_1^{(w)}), (e_2^{(t)}, \mathbf{E}_2^{(w)}), \dots, (e_{L'}^{(t)}, \mathbf{E}_{L'}^{(w)})$  の入力に対し、 $L' + 1$  番目の発話者  $e_{L'+1}^{(t)}$  の確率に対応する確率ベクトル  $\mathbf{p}_{L'+1}^{(t)}$  を出力する。

#### b) 生成中の発話の次の単語の出力

1 番目から  $L'$  番目までの完全な発話の列  $(e_1^{(t)}, \mathbf{E}_1^{(w)}), (e_2^{(t)}, \mathbf{E}_2^{(w)}), \dots, (e_{L'}^{(t)}, \mathbf{E}_{L'}^{(w)})$  に  $L' + 1$  番目の発話の  $J'$  番目までの単語列  $[e_{L'+1,1}^{(w)} \ e_{L'+1,2}^{(w)} \ \dots \ e_{L'+1,J'}^{(w)}]$  を加えた入力に対し、 $J'$  番目の単語  $e_{L'+1,J'+1}^{(w)}$  の確率に対応する確率ベクトル  $\mathbf{p}_{L'+1,J'+1}^{(w)}$  を出力する。なお、ここでは  $J' = 0$  の場合、すなわち、 $L' + 1$  番目の発話について、発話者のみが与えられ、発話内容の単語が 1 つも入力されていない場合も含まれる。

このモデルを再帰的に適用することで台本形式小説を生成することができる。以下、入出力関係の詳細を記述する。ただし活性化関数への入力についてはバイアス項を省略する。

### 2.3.2 過去の発話の分散表現の出力

まず、パラメータ  $\mathbf{K}^{(t)} \in \mathbb{R}^{D^{(t)} \times M}$ ,  $\mathbf{K}^{(w)} \in \mathbb{R}^{D^{(w)} \times I}$  を用いて話者、および単語を  $\mathbf{t}_{\ell} = \mathbf{K}^{(t)} e_{\ell}^{(t)} \in \mathbb{R}^{D^{(t)}}$ ,  $\mathbf{w}_{\ell,j} = \mathbf{K}^{(w)} e_{\ell,j}^{(w)} \in \mathbb{R}^{D^{(w)}}$  と分散表現に変換する。 $\ell = 1, 2, \dots, L'$  の時、発話内容  $\mathbf{E}_{\ell}^{(w)}$  は発話符号化 LSTM に入力され、1 つの発話内容が 1 つの分散表現  $\mathbf{u}_{\ell} \in \mathbb{R}^{D^{(u)}}$  に変換される:

$$\begin{aligned} & (\mathbf{h}_{\ell,j}^{(e)}, \mathbf{m}_{\ell,j}^{(e)}) \\ &= \begin{cases} (\mathbf{h}_J^{(e)}, \mathbf{m}_J^{(e)}) & (j = J) \\ g_{\text{LSTM}}^{(e)}(\mathbf{w}_{\ell,j}, \mathbf{h}_{\ell,j+1}^{(e)}, \mathbf{m}_{\ell,j+1}^{(e)}; \Theta^{(e)}) & (j = 0, 1, \dots, J-1) \end{cases} \end{aligned} \quad (1)$$

また  $\mathbf{u}_{\ell} = \mathbf{m}_{\ell,0}^{(e)}$ 。ただし、ここで  $\mathbf{h}_J^{(e)}, \mathbf{m}_J^{(e)} \in \mathbb{R}^{D^{(u)}}$  はあらかじめ定められた定数ベクトルであり、 $j = 0, 1, \dots, J$  に対し、 $\mathbf{h}_{\ell,j}^{(e)}, \mathbf{m}_{\ell,j}^{(e)} \in \mathbb{R}^{D^{(u)}}$  である。また、 $g_{\text{LSTM}}^{(e)} : \mathbb{R}^{D^{(w)}} \times \mathbb{R}^{D^{(u)}} \times \mathbb{R}^{D^{(u)}} \rightarrow \mathbb{R}^{D^{(u)}} \times \mathbb{R}^{D^{(u)}}$  は発話符号化 LSTM の入出力関係を表す写像であり、 $\Theta^{(e)}$  は発話符号化 LSTM のパラメータである。発話符号化 LSTM への入力は、通常の文の単語順の逆順であることに注意されたい。この逆順入力のアイデアは [Sutskever 14] によるものである。

### 2.3.3 次の発話の分散表現の出力

発話内容の分散表現  $\mathbf{u}_{\ell}$  は、話者の分散表現  $\mathbf{t}_{\ell}$  と結合され、文脈更新 LSTM に入力される。 $\ell = 1, 2, \dots, L'$  に対し、以下の出力が得られる:

$$(\mathbf{h}_{\ell}^{(c)}, \mathbf{m}_{\ell}^{(c)}) = g_{\text{LSTM}}^{(c)} \left( \begin{bmatrix} \mathbf{u}_{\ell} \\ \mathbf{t}_{\ell} \end{bmatrix}, \mathbf{h}_{\ell-1}^{(c)}, \mathbf{m}_{\ell-1}^{(c)}; \Theta^{(c)} \right). \quad (2)$$

ただし、ここで  $\mathbf{h}_0^{(c)}, \mathbf{m}_0^{(c)} \in \mathbb{R}^{D^{(c)}}$  はあらかじめ定められた定数ベクトルであり、 $\ell = 0, 1, \dots, L'$  に対し、 $\mathbf{h}_{\ell}^{(c)}, \mathbf{m}_{\ell}^{(c)} \in \mathbb{R}^{D^{(c)}}$  である。また、 $g_{\text{LSTM}}^{(c)} : \mathbb{R}^{D^{(u)}+D^{(t)}} \times \mathbb{R}^{D^{(c)}} \times \mathbb{R}^{D^{(c)}} \rightarrow \mathbb{R}^{D^{(c)}} \times \mathbb{R}^{D^{(c)}}$  は発話符号化 LSTM の入出力関係を表す写像であり、 $\Theta^{(c)}$  は発話符号化 LSTM のパラメータである。

ここで得られた文脈更新 LSTM の出力  $\mathbf{m}_{L'}^{(c)}$  が、以降で述べるように  $L' + 1$  番目の発話の発話者と発話内容を決定する。

### 2.3.4 次の発話者の出力

$L'$  番目までの完全な発話が入力として与えられている場合、 $L' + 1$  番目の発話者については、パラメータ  $\mathbf{K}^{(t')} \in \mathbb{R}^{D^{(t')} \times D^{(c)}}$ ,  $\mathbf{L}^{(t')} \in \mathbb{R}^{M \times D^{(t)}}$  と活性化関数  $a^{(t')}$  によって、

$$\mathbf{p}_{L'+1}^{(t')} = a_{\text{SM}} \left( \mathbf{L}^{(t')} a^{(t')} \left( \mathbf{K}^{(t')} \mathbf{m}_{L'}^{(c)} \right) \right) \quad (3)$$

と確率ベクトル  $\mathbf{p}_{L'+1}^{(t')}$  が与えられる。

### 2.3.5 発話内容の出力

$L'$  番目までの完全な発話に加え、既に得られた  $L' + 1$  番目の発話の発話者  $e_{L'+1}^{(t)}$  と発話内容の  $J'$  番目までの単語が入力として与えられている場合、 $J' + 1$  番目の単語については、以下のように確率ベクトルが得られる。 $e_{L'+1}^{(t)}$  の分散表現  $\mathbf{t}_{L'+1} = \mathbf{K}^{(t)} e_{L'+1}^{(t)}$  と文脈更新 LSTM の出力  $\mathbf{m}_{L'}^{(c)}$  を結合し、パラメータ  $\mathbf{K}^{(u')} \in \mathbb{R}^{D^{(u')} \times (D^{(c)}+D^{(t)})}$  の乗算と活性化関数

$a^{(u')}$  をの適用により、 $L' + 1$  番目の発話の発話内容の分散表現  $\mathbf{v}_{L'+1}$  が次のように得られる:

$$\mathbf{v}_{L'+1} = a^{(u')} \left( \mathbf{K}^{(u')} \begin{bmatrix} \mathbf{m}_{L'}^{(c)} \\ \mathbf{t}_{L'+1} \end{bmatrix} \right) \quad (4)$$

この  $\mathbf{v}_{L'+1}$  と、話者の分散表現  $\mathbf{t}_{L'+1}$ 、および既に得られた単語  $e_{L'+1,j}^{(w)}$  の分散表現  $\mathbf{w}_{L'+1,j} = \mathbf{K}^{(w)} e_{L'+1,j}^{(w)}$  が結合され、発話復号化 LSTM へ入力されて、以下が得られる:

$$\begin{aligned} & \left( \mathbf{h}_{L'+1,j}^{(d)}, \mathbf{m}_{L'+1,j}^{(d)} \right) \\ & = \begin{cases} \left( \mathbf{h}_0^{(d)}, \mathbf{m}_0^{(d)} \right) & (j = 0) \\ \mathbf{g}_{\text{LSTM}}^{(d)} \left( \begin{bmatrix} \mathbf{v}_{L'+1} \\ \mathbf{t}_{L'+1} \\ \mathbf{w}_{L'+1,j} \end{bmatrix}, \mathbf{h}_{L'+1,j-1}^{(d)}, \mathbf{m}_{L'+1,j-1}^{(d)}; \Theta^{(d)} \right), & (j = 1, 2, \dots, J) \end{cases} \end{aligned} \quad (5)$$

ここで  $\mathbf{h}_0^{(d)}, \mathbf{m}_0^{(d)} \in \mathbb{R}^{D^{(u)}}$  はあらかじめ定められた定数ベクトルであり、 $j = 0, 1, \dots, J$  に対し、 $\mathbf{h}_{L'+1,j}^{(d)}, \mathbf{m}_{L'+1,j}^{(d)} \in \mathbb{R}^{D^{(u)}}$  である。また、 $\mathbf{g}_{\text{LSTM}}^{(d)} : \mathbb{R}^{D^{(c)}+D^{(t)}+D^{(w)}} \times \mathbb{R}^{D^{(u)}} \times \mathbb{R}^{D^{(u)}} \rightarrow \mathbb{R}^{D^{(u)}} \times \mathbb{R}^{D^{(u)}}$  は発話復号化 LSTM の入出力関係を表す写像であり、 $\Theta^{(d)}$  は発話復号化 LSTM のパラメータである。また、各発話の 0 番目の単語  $e_{L'+1,0}^{(w)}$  は、特殊単語 [go] である。

発話復号化 LSTM の出力にパラメータ  $\mathbf{K}^{(w')} \in \mathbb{R}^{D^{(w)} \times D^{(u)}}$ ,  $\mathbf{L}^{(w')} \in \mathbb{R}^{I \times D^{(w)}}$  を乗算し、活性化関数  $a^{(w')}$  を適用して、次のように発話内容の各単語の確率ベクトル  $\mathbf{p}_{L'+1,J'+1}^{(w')}$  を得る:

$$\mathbf{p}_{L'+1,J'+1}^{(w')} = \text{asM} \left( \mathbf{L}^{(w')} a^{(w')} \left( \mathbf{K}^{(w')} \mathbf{m}_{L'+1,J'+1}^{(d)} \right) \right). \quad (6)$$

なお、発話復号化 LSTM からの出力は通常の文の単語順通りである。また、通常は途中から [pad] のみが出力されるようになり、事実上そこでその発話は終了する。

以上のように、本モデルは入力から次の発話者または単語の確率ベクトルを出力する。図 2 に概要を示す。

## 2.4 学習

発話の列  $(e_1^{(t)}, \mathbf{E}_1^{(w)}), (e_2^{(t)}, \mathbf{E}_2^{(w)}), \dots, (e_L^{(t)}, \mathbf{E}_L^{(w)})$  (小説全体) を  $\mathbf{E}$  と、またすべてのパラメータをまとめて  $\Theta$  と書くことにする。学習における損失関数を cross entropy を使って以下のように定義する:

$$\ell_{\text{train}}(\mathbf{E}; \Theta)$$

$$\stackrel{\text{def}}{=} \sum_{\ell=1}^L \left( \lambda \ell_{\text{CE}} \left( e_{\ell}^{(t)}, \mathbf{p}_{\ell}^{(t')}(\Theta) \right) + \sum_{j=1}^J \ell_{\text{CE}} \left( e_{\ell,j}^{(w)}, \mathbf{p}_{\ell,j}^{(w')}(\Theta) \right) \right). \quad (7)$$

ここで  $\lambda \in \mathbb{R}$  は発話者と発話内容の学習の重要性を調節するハイパーパラメータである。提案モデルの学習は、この損失関数の最小化を行う。

## 2.5 生成

生成時は、学習時に用いた損失関数に加え、二つの罰則項を加えた状態で beam search を行う。一つ目の罰則項は発話内容の

短さに科される。これは極端に短い応対が連続する単調な出力を防ぐためである。ここでは  $\ell$  番目の発話の発話内容の長さ  $J_{\ell}$  を、その発話内容の中の最初の [pad] までの長さで定義し、罰則を科す。二つ目の罰則項は同一話者の登場の間隔の短さに科される。これは極めて少ない話者で対話が進行するのを防ぐためである。ここでは  $\ell$  番目の発話の発話者の連続度  $s_{\ell}$  を、その発話者が直近で何発話前に発話したかで定義し、罰則を科す。発話内容の短さ、および同一話者の連続度に対する罰則項は、いずれも単調非減少関数  $\ell_{\text{len}}, \ell_{\text{succ}} : \mathbb{N} \rightarrow \mathbb{R}$  によって実現される。三つ目の罰則項は発話の最初の三点リーダ「…」に科す。これは、実際の台本形式小説において、三点リーダのみで終了する発話が非常に多く存在し、学習がそれに引きずられるため、これを除外しないと三点リーダの応酬に陥るからである。この罰則項は指示関数を用いた罰則関数  $\ell_{\text{sil}}(e^{(w)}) \stackrel{\text{def}}{=} \lambda_{\text{sil}} 1\{e^{(w)} = \text{「}\dots\text{」}\}$  で定義される。これらの罰則項を加えて、出力に用いる損失関数を以下で定義する:

$$\ell_{\text{gen}}(\mathbf{E}; \Theta)$$

$$\stackrel{\text{def}}{=} \sum_{\ell=1}^L \left( \lambda \ell_{\text{CE}} \left( e_{\ell}^{(t)}, \mathbf{p}_{\ell}^{(t')}(\Theta) \right) + \sum_{j=1}^J \ell_{\text{CE}} \left( e_{\ell,j}^{(w)}, \mathbf{p}_{\ell,j}^{(w')}(\Theta) \right) + \ell_{\text{len}}(J_{\ell}) + \ell_{\text{succ}}(s_{\ell}) + \ell_{\text{sil}}(e_{\ell,1}^{(w)}) \right). \quad (8)$$

生成は、この損失関数を小さくするような発話列を一単語ずつ (あるいは一発話者ずつ)  $n$ -beam search で、すなわち  $n$  個まで状態を保持した貪欲法で行う。

## 2.6 その他

上記の他に、後述の実験ではいくつかのレイヤーに Dropout [Hinton 12] を適用した。また、発話者の入出力と同様の要領で発話の種類の違いに用いられる括弧記号の入出力も行った。

```
片桐早苗「それはそれ、これはこれ」# match
高垣楓「はい、そうです。私、今まで通りに頑張ってきました」
東郷あい「ああ、そうだね。それじゃ、また明日ね」
片桐早苗「うん、ありがとう。それじゃ、また明日ね」
高垣楓「はい、お疲れ様です。プロデューサー」# match
東郷あい「ああ、すまない、少し寝れ気味でね」
片桐早苗「んー、そうねえ…あ、これお土産なんだっけ」
高垣楓「んー、これは…フッフッフ、そうだね」
高垣楓「ふふ、ありがとうございます」# match
片桐早苗「んっ…んっ…」# match
```

図 3 提案手法による出力例 ( $D^{(w)} = 300$ ,  $D^{(u)} = 300$ ,  $D^{(c)} = 1600$ ,  $D^{(u')} = 200$ ): なお、# match は学習データに完全一致した出力を示す。

```
P「そうですね、それじゃ、おやすみなさい」
千川ちひろ「おやすみなさい…」
佐久間まゆ「pさんはどういう風にお茶を飲むのでしょうか」
P「そうですね、まゆが作ってくれたんだ」
佐久間まゆ「pさんはどういう風にお茶を飲むのでしょうか」
渋谷謙「いや、そういうわけじゃないんだけどさ」
P「そうですね。俺は潮のプロデューサーだからな」
島村卯月「プロデューサーさんは、私のことをどう思っているんですか」
渋谷謙「私は、卯月のことが、好きなんだ」
本田未央「しぶりは、私のことが大好きなんだね」
北条加蓮「うん、そうだね。私も、卯月のことが好きなんだよ」
島村卯月「えへへ、ありがとうございますっ」
```

図 4 比較手法による出力例 ( $D^{(w)} = 300$ ,  $D^{(u)} = 300$ ,  $D^{(c)} = 1600$ ,  $D^{(u')} = 200$ ):

表 1 提案手法及び比較手法の Perplexity 及び BLEU

	perplexity	BLEU
RNN-Encoder-Decoder	41.97	<b>0.6876</b>
TULIP	<b>39.51</b>	0.56642

### 3. 実験

#### データ

本研究で提案する枠組は、登場人物の固定を必要とし、同時には一作品しか扱えない。従って実験では、Web上の台本形式小説の中でも投稿数がトップクラスである「アイドルマスター シンデレラガールズ」の二次創作の台本形式小説に限定し行った。データの収集はWebのクロールにより、著作権法第四十七条の七の範囲内となるよう、電子計算機による情報解析のみを目的として複製した。当該作品の登場人物は180人以上、小説数は8817、総発話数は2211505であった。

#### 実験条件

ベースラインとして、翻訳で用いられるRNN encoder-decoderモデル [Cho 14] を、提案手法と同様話者を考慮したものに拡張したものと比較する。このモデルは、提案手法の文脈更新LSTMに相当する部分を、前の文から次の文へのDenseレイヤーに置き換えたものである。従って、二文以上前のデータを生成に使用しない。これにより、提案手法の文脈更新LSTMによる文脈の学習能力を測ることができると期待される。実装はTensorflowをバックエンドにKerasで行い、最適化手法はAdam [Kingma 14] を用いた。学習データ全体を16反復学習した。ハイパーパラメータは以下のとおり定めた。 $\lambda = 0.001$ ,  $D^{(l)} = 30$ , beam searchのbeam数は2とした。罰則項については、 $\ell_{\text{len}}(k) = \frac{5.0}{k+1}$ ,  $\ell_{\text{succ}}(k) = \frac{5.0}{k+1}$ ,  $\ell_{\text{sil}} = 5.0$ とした。定量評価ではパラメータのグリッドサーチを $D^{(w)} = 300, 600$ ,  $D^{(u)} = 300, 600$ ,  $D^{(c)} = 600, 1600$ ,  $D^{(u)} = 200, 400$ のすべてのパラメータの組で行った。比較手法についても、同様のグリッドサーチを行っている。この条件の下、図1(学習データから除外)の開始数文を種としてのみ用い、それぞれの手法で生成を行った。

#### 定性評価

提案手法の出力例として $D^{(w)} = 300$ ,  $D^{(u)} = 300$ ,  $D^{(c)} = 1600$ ,  $D^{(u)} = 200$ としたものを図3にあげる。このパラメータ及び例は著者によるcherry-pickingである点に注意されたい。同じ小説を種にした場合の比較手法の生成例を図4に示す。どちらも、文脈の学習という点では、十分な性能とは言えないが、Web小説のみを学習から発話単位で自然な日本語を生成している。これは、文脈を学習する上では、モデルの複雑さに対し小説数8000が十分でないのに対し、発話の学習の上では220万という発話数が十分であることを示唆している。

#### 定量評価

定量評価では、テストデータに対するperplexityと、n-gramの一致度に基づく、翻訳分野での定量評価指標BLEU [Papineni 02] を用いた。ただしBLEUは文のみを評価できるため、発話者については所与とし発話内容のみ評価を行った。表2.6に結果を示す。提案手法は比較手法に比べperplexityでは上回ったもののBLEUスコアでは下回った。興味深いのはこれらのスコアでバリデーションによって選ばれたハイパーパラメータ( $D^{(w)} = 300$ ,  $D^{(u)} = 300$ ,  $D^{(c)} = 600$ ,  $D^{(u)} = 200$ )が先ほど筆者が選んだハイパーパラメータと異なる点である。定量評価指標には、依然として改善点が残るといえる。

### 4. 結論と課題

本研究では台本形式小説の自動生成を行い、提案手法の発話レベルでの高い性能を示した。課題としては文脈レベルでの性能の向上のほか、提案手法の各コンポーネントの性能への寄与の精査が考えられる。また、1発話1分以上かかる生成時間に

ついては改善が必要である。

### 参考文献

- [Cho 14] Cho, K., Merriënboer, van B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734 (2014)
- [Hinton 12] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R.: Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580* (2012)
- [Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997)
- [Jain 17] Jain, P., Agrawal, P., Mishra, A., Sukhwani, M., Laha, A., and Sankaranarayanan, K.: Story Generation from Sequence of Independent Short Descriptions, *arXiv preprint arXiv:1707.05501* (2017)
- [Kashiwabara 12] Kashiwabara, T.: 「2ちゃんねるスレッドまとめブログ」によるニュース・コミュニケーションに関する一考察, *哲学*, Vol. 128, pp. 207–234 (2012)
- [Kingma 14] Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014)
- [Li 16] Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B.: A persona-based neural conversation model, *arXiv preprint arXiv:1603.06155* (2016)
- [Ouchi 16] Ouchi, H. and Tsuboi, Y.: Addressee and response selection for multi-party conversation, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2133–2143 (2016)
- [Papineni 02] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation, in *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318 Association for Computational Linguistics (2002)
- [Serban 16] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A. C., and Pineau, J.: Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models., in *AAAI*, Vol. 16, pp. 3776–3784 (2016)
- [Sordoni 15] Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., and Nie, J.-Y.: A hierarchical recurrent encoder-decoder for generative context-aware query suggestion, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 553–562 ACM (2015)
- [Sutskever 14] Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to sequence learning with neural networks, in *Advances in neural information processing systems*, pp. 3104–3112 (2014)
- [Takagi 14] Takagi, D., Sato, S., and Komatani, K.: 会話を中心とした超短編小説の自動生成, in *The 28th Annual Conference of the Japanese Society for Artificial Intelligence* (2014)