ニューラル機械翻訳を活用した日本語作文の別解生成

澤山 熱気　松岡 保静　内田 渉　礒田 佳徳
株式会社 NTT ドコモ
{atsuki.sawayama.cm, matsuokaho, uchidaw, isoday}@nttdocomo.com

1 はじめに

近年，国内外国人労働者に関わる法改正に伴い，外国人労働者の受け入れが拡大しつつある。そのため，外国人労働者が日本で働くにあたって必要な語学学習環境を整備することが急務である。現在の日本語学習における課題のひとつとして，参考書や短期間の学習では，様々な言い回しが学びにくいことが挙げられる。なぜなら，母国語から日本語に訳出させる問題では，学習者の解答文と出題者が決めた一通りの日本語（模範解答）文が (表層的に) 近いか否かを基準に採点することが一般的なためである。本来であれば，両言語で表現できる意味・内容にズレがあるため，学習者の解答を複数の母国語文と複数の模範解答文を用いて採点すべきだが，母国語文と日本語文の多対多の対訳を整備するコストは膨大であり，現実的ではない。そのため，まずは日本語文が複数文用意された 1 対多の対訳を整備し，解答文の採点や模範解答・別解の提示ができることを目指す必要がある。

そこで，本研究では，英語から日本語への訳出問題を想定し，多様な日本語文の別解生成を試みる。提案手法では，機械翻訳モデルとして学習済みのアテンション付き RNN エンコーダ・デコーダ [1] モデルを用い，英語文をエンコードしたエンコーダの隠れ状態に係数ベクトルを与えて日本語文を生成し，表層が離れた文を収集する。実験の結果，特に大規模なコーパスを活用した機械翻訳モデルを用いて生成した際に多くの文を得ることができ，言い回しが変化することを確認できた。

2 日本語作文採点における課題

近年，日本語学習者をサポートする目的で，自然言語処理技術を活用する研究がおこなわれており [2, 3]，自然言語処理技術を用いた作文採点をおこなう語学学習アプリが登場している。

単純な作文採点方法の一つとして，機械翻訳で用いられる精度指標である BLEU スコア [4] を活用し，学習者の解答文と模範解答の表層的類似度を算出する方法が考えられる。しかしながら表層的な情報だけでは，文の意味がほとんど同じであるが，言い回しの異なる文を妥当に採点することが難しい。例えば，学習者に英語文を日本語文へ訳出させる問題を出し，日本語模範解答が与えられたとする。

- 問題文：I was surprised to see him there.
- 模範解答: 私はそこで彼に会って驚いた。

この問題文に対し，学習者の解答は以下のさまざまな文が考えられる。

- 彼とその場所で会ってビックリしちゃったよ。
- そちらで彼にお目にかかれて仰天いたしました。
- そこで彼見て，僕ぶったまげたんです。

これらの解答のように，言い回しが異なるが，意味が伝わる文を採点する場合，編集距離や BLEU スコアといった文の表層のみを用いて比較する評価尺度だけでは，模範解答と表層が大きく違うことで学習者の解答が低く採点されてしまう。

このような様々な言い回しに対し，柔軟（あるいは頑健）な採点ができるよう，採点システム性能を向上させたり，別解を提示することが必要である。これができなければ，日本語学習者は，自身の解答を信用できず，上記の模範解答を暗記せざるを得なくなる。

採点性能を向上させる方法のひとつとして，複数の模範解答を用いて採点する方法が考えられる。これは，学習者が入力する目的言語側の模範解答を複数用意し，学習者の解答と複数の模範解答で評価・採点をおこなうことで採点を改善する方法で，人手評価との相関が高くなることが知られている [5]。しかしながら，日本語のような多様な言い回しが考えられる言語で，表層が異なることを考慮した 1 対多の対訳作成は，単純な 1 対 1 の対訳作成と比較してコストがかかる。

このような表層的に離れた，意味的に類似する別解を自動生成し，採点性能の向上に活用したり，別解と

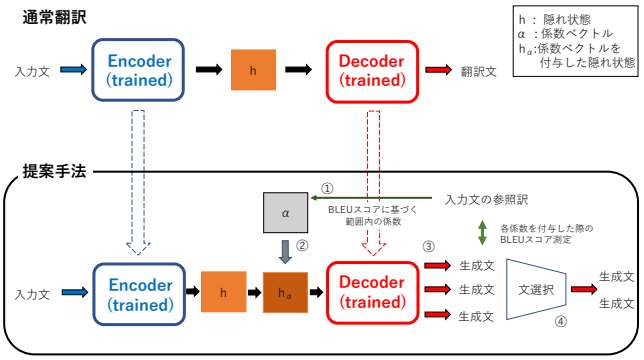

図 1: 通常翻訳と提案手法

して提示できれば，日本語学習者は多様な言い回しを学習できると考えられる。

3 提案手法

提案手法では，学習済みの機械翻訳モデルを活用し，隠れ状態に微小な係数ベクトルを付与することでわずかに意味表現を変化させ，別解の生成を試みる。理由は二つある。一つ目は，事前学習された言語モデルの活用である。近年，大規模言語モデルによる事前学習知識を活用する研究 [6] がおこなわれているように，学習済みの言語モデルを文生成に活用できるのではないかと考えたためである。二つ目は，エンコーダ・デコーダを用いる機械翻訳モデルにおいて，原言語をエンコードした隠れ状態に同じ次元サイズで要素が同じである微小な数値を与えることは，画像処理における明るさやコントラスト変化に相当すると考えられる。これらが変化しても画像の被写体自体が変わらないように，係数を付与しても文の意味を保持しながら言い回しが変化することを期待している。

提案手法の概要を図 1 に示す。はじめに，学習済みニューラル機械翻訳モデルに対象とする原言語文を入力し，エンコーダから隠れ状態 h を得る。次に，得られた隠れ状態に係数ベクトル α を付与しデコーダに渡し，文を生成する。これをテストデータの参照訳と生成文の BLEU スコアに基づく α の範囲内で α の付与と文生成を複数回繰り返す。テストデータの参照訳と生成文の BLEU スコアに基づいて α の範囲を決定する理由は，精度が大きく低下すると生成文が大きく劣化することが想定されるためである。次に，生成された目的言語文から参照訳と表層的に離れた文を選択し，それを別解とする。

3.1 係数ベクトル α を付与した隠れ状態を用いた文生成

提案手法ではアテンション付きの RNN ベースのエンコーダ・デコーダモデル [1] を用いる。RNN ベースのモデルを用いる理由として，入力文をひとつの隠れ状態とするので，これに適切な範囲で係数ベクトルを与えることで文全体の言い回しに影響を与えることができると考えられるためである。

原言語文（系列）X, 目的言語文 (系列)Y を以下のように定義する。

$$X = (x_1, x_2, \cdots, x_I)$$

$$Y = (y_1, y_2, \cdots, y_J)$$

エンコーダは原言語文 X を受け取り，RNN を通じて順方向に隠れ状態 h を返す。原言語文 X の最後のトークン x_I が入力された際の隠れ状態は以下になる。

$$h_I = RNNencoder(\overrightarrow{h_{I-1}}, x_I) \tag{1}$$

この h に，h と同じ次元サイズで，各要素が同じ値の係数ベクトル α を付与する。このとき，α を二種類の設定で付与する。ひとつは，h に足し合わせる設定 (Add) で，もうひとつは，h と掛け合わせる設定 (Mul) である。

$$h_{\alpha(Add)} = \alpha + h_I \tag{2}$$

$$h_{\alpha(Mul)} = \alpha \times h_I \tag{3}$$

α が付与された隠れ状態 h_α をデコーダに渡し，目的言語文 Y を生成する。これにより学習済み機械翻訳モデルのパラメータを活用し，文生成が可能になる。

3.2 生成文からの文選択

次に，生成された文の中から，既存の文間評価尺度である編集距離を用いて参照訳と生成文を比較し，参照訳と表層的に異なる文を選択する。

4 実験

実験では，英語から日本語への訳出問題を想定し，目的言語側である日本語文を生成する。

4.1 実験設定

実験では，サイズの異なる二つのコーパスを用いる。

一つは，田中コーパス (tanaka)[9] である。これは日本の大学生によって収集・翻訳された対訳データである。もう一つはドコモがもつ丁寧語の汎用話し言葉コーパス (docomo) である。加えて，語学学習用に作成した対訳データ 242 文 (Japanese training data, JT とする) を別解生成対象文として用いる。表 1 に各対訳のデータサイズを示す。

英日対訳の前処理として，英語は Moses toolkit の tokenizer.perl[7] を，日本語は MeCab[1] を利用した。加えて，NFKC 正規化と，文末の句点付与をおこなった。機械翻訳モデルとして，アテンション付きエンコーダ・デコーダモデル [1] が実装されている，OpenNMT-py[8] をデフォルト設定で利用した。

評価尺度として BLEU スコアを用い，Moses toolkit[7] の multi-bleu.perl で測定した。結果を表 2 に示す。

表 1: 実験に用いた日英対訳データ

	tanaka(文)	docomo(文)	JT(文)
train	149,796	4,553,076	-
valid	3,000	3,000	-
test	3,000	3,000	242

表 2: 通常翻訳時の BLEU スコア

	tanaka	docomo
test	32.29	31.01
JT	42.78	54.92

4.2 係数ベクトル α の範囲選定

α を付与することで，様々な文の生成が期待できるが，α の大きさによって，大幅に BLEU スコアが低下し，文意や文構造が大きく劣化した文を生成する可能性がある。そのため，テストデータに α を付与した際の BLEU スコアが低下しすぎない範囲を各モデル・テストデータにおいて調査した。Add 設定では α の範囲を [-1.0,1.0]，Mul 設定では範囲を [0,2.0] とし，0.1 刻みで BLEU スコアの変化を確認した。

結果を表 3, 4 に示す。Add 設定が Mul 設定よりも α の影響を受けやすく，α の絶対値が大きくなるにつれて，オーバートランスレーションや訳抜け，文意の変化といった文劣化の度合いが大きくなった。加えて，コーパスサイズによって α の付与によるテストデータの BLEU スコアへの影響が大きくなることがわかった。これは，学習文数が増加することで文意を表す h のベクトル表現が密になり，わずかな数値の付与であっても h の意味表現が大きく変化するためだと考えられる。docomo モデルでは，α の付与による BLEU スコアの変動が大きいことから，適切な範囲で α を与えることで，言い回しを変化させられると考えられる。

表 3: 各テストデータの BLEU スコア変化 (Add)

α	tanaka (test)	docomo (test)	tanaka (JT)	docomo (JT)
-1.0	20.94	2.3	30.07	2.4
-0.9	22.16	2.74	31.99	2.19
-0.8	23.11	3.46	32.37	3.65
-0.7	24.28	4.86	33.92	6.07
-0.6	25.7	7.22	37.05	10.09
-0.5	27.39	9.27	38.95	14.42
-0.4	28.97	11.75	39.77	19.49
-0.3	30.45	16.09	40.75	28.83
-0.2	31.52	23.15	41.64	41.08
-0.1	32.04	29.09	42.42	50.54
0	32.29	31.01	42.78	54.92
0.1	32.38	26.78	43.41	47.79
0.2	32.22	7.08	42.38	17.57
0.3	31.53	1.29	42.34	0.99
0.4	30.07	1.37	39.9	2.5
0.5	28.08	1.71	37.21	2.56
0.6	26.09	1.55	34.29	2.05
0.7	23.82	1.23	30.23	1.37
0.8	22.15	0.87	28.16	1.43
0.9	20.57	0.77	24.3	0.48
1.0	18.74	0.68	22.3	0.0

表 4: 各テストデータの BLEU スコア変化 (Mul)

α	tanaka (test)	docomo (test)	tanaka (JT)	docomo (JT)
0.0	18.13	11.26	25.55	20.37
0.1	20.75	17.65	29.46	31.69
0.2	22.97	21.27	32.15	37.25
0.3	25.57	23.71	35.17	41.51
0.4	27.63	25.68	38.23	44.43
0.5	29.08	27.41	38.87	47.35
0.6	30.27	28.82	41.13	49.39
0.7	31.1	29.62	42.16	51.04
0.8	31.61	30.38	42.97	52.53
0.9	32.13	30.73	42.82	53.94
1.0	32.29	31.01	42.78	54.92
1.1	32.45	30.98	42.48	55.3
1.2	32.44	31.25	42.54	55.46
1.3	32.34	31.22	42.58	55.06
1.4	32.33	30.61	42.61	54.29
1.5	31.78	29.87	42.64	54.95
1.6	31.12	29.06	42.71	54.67
1.7	30.58	28.22	43.18	54.04
1.8	30.15	27.61	42.35	53.63
1.9	29.74	26.99	41.95	52.78
2.0	29.21	26.45	42.41	51.27

4.3 日本語文の別解生成

事前調査に基づき，α の付与による BLEU スコアの変化が大きかった docomo モデルと JT データを活用し，JT データの別解生成を試みた。本研究での α の範囲は α の変動で BLEU スコアが 10 ポイント程度急落した値より前の値とする。Add 設定では [-0.15, 0.15] の範囲で，0.01 刻みで変動させた値を，Mul 設定では

[1] http://taku910.github.io/mecab/

[0.1, 2.0] の範囲で 0.1 刻みで変動させた値を h に付与し，文を生成した。加えて，OpenNMT-py のデフォルト設定では，2 レイヤのスタックされた LSTM を用いており，レイヤごとに別の数値を与えられるため，実験では，範囲内の α をレイヤごとに全ての組み合わせで付与をし，文を生成した。文生成後，生成された文集合と参照訳を比較し，以下条件を満たす文の中から，編集距離が遠い順に文 (重複なし) を選択した。

- 参照訳とのトークン一致率が 3 割以上
- 参照訳とのトークン数の比が 0.5 倍以上, 1.5 倍以下

5 結果と考察

表 5: JT データで言い回し変化が得られた文例

参照訳 (英)	Do you often clean your room?
参照訳 (日)	あなたは自分の部屋をよくそうじしますか。
docomo (4-best)	よく部屋を掃除しますか。 部屋をよく掃除しますか。 よく部屋を掃除しますか？ あなたはよく部屋を掃除しますか。
docomo (文選択後)	よく部屋の掃除をしますか。 部屋をよくきれいにしますか。 部屋の掃除はよくしますか。 お部屋はよく掃除しますか。

JT データを用いて α の付与によって生成された文は，1 文あたり平均 12.2 文となった。図 2 に JT データの各参照訳 (日) のトークン数と，参照訳（英）を用いて生成した日本語文の文数を示す。元々のトークン数が多い文ほど，生成文数も増加傾向にある。トークン数が少ない場合でも生成文が多かった文は，「兄/弟/兄弟」や「バッグ/カバン/鞄」など単語のバリエーションが多かった。

言い回し変化が得られた文例を表 5 に示す。文例の 4-best では，全ての文で“部屋を掃除し”という言い回しになっており，トークンの語順もほとんど同じであるが，条件によって選択された生成文では助詞の言い回しが異なっていたり，“きれいにする”というように語彙が異なっている。一方で，選択した生成文の中にも“彼”や”彼女”といった三人称が訳抜けした劣化文が存在した。また，文選択の条件外だった文でも，別解として選択すべき文が存在した。その例を表 6 に示す。劣化文の除去を含め，生成された文からの文選択には改善の余地がある。

6 おわりに

本研究では，学習済み機械翻訳モデルを活用し，別解生成と分析をおこなった。実験では複数の文を生成

表 6: 文選択条件外の選択すべき生成文

参照訳 (日)	私たちは自分たちの寝室を片付けなければなりません。
生成文	私たちはベッドルームを掃除しなければならない。
参照訳 (日)	彼はいつも怒っている先生だ。
生成文	彼はいつも腹を立てる教師である。
参照訳 (日)	最新の機種が最高の機種とは限らない。
生成文	最新のモデルは必ずしも最高ではない。

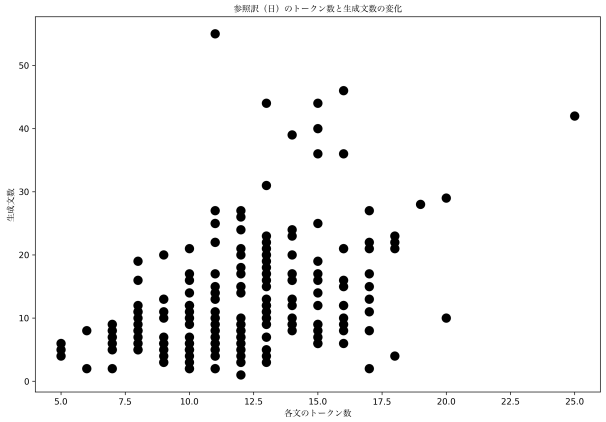

図 2: JT データの参照訳 (日) のトークン数と α の付与によって生成された文数

できたが文選択に改善の余地がある。今後は，デコーダ側の言語モデルを活用した文選択手法，および，翻訳モデルに構文情報を取り込む手法 [10] との組み合わせを検討したい。

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, pp. 1-15, 2015.

[2] 小川耀一朗, 山本和英. 分類モデルを用いた日本語学習者の格助詞誤り訂正. 言語処理学会第 25 回年次大会, 2019.

[3] 新井美桜, 金子正弘, 小町守. 日本語学習者向けの文法誤り検出機能付き誤用例文検索システム. 言語処理学会第 25 回年次大会, 2019.

[4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th ACL*, pages 311 - 318, 2002.

[5] Andrew M Finch, Yasuhiro Akiba, and Eiichiro Sumita. How does automatic machine translation evaluation correlate with human scoring as the number of reference translations increases? In *LREC*, 2004.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] P.Koehn,et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of 45th ACL*, pp. 177 - 180, 2007.

[8] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of the 55th ACL*, pp. 67-72, 2017.

[9] Yasuhito Tanaka. Compilation of a multilingual parallel corpus. In *Proceedings of PACLING 2001*, pp. 265-268, 2001.

[10] Raphael Shu, Hideki Nakayama and Kyunghyun Cho. Generating Diverse Translations with Sentence Codes. In *Proceedings of the 57th ACL*, pp. 1823-1827, 2019.