人工負例による識別器を用いたニューラル機械翻訳

白井 圭佑[†] 橋本 和真[‡*] 江里口 瑛子[††*] 森 信介[§] 二宮 崇[¶]
[†] 京都大学 情報学研究科 知能情報学専攻
[‡]Salesforce Research [††]Microsoft Research
[§] 京都大学 学術情報メディアセンター
[¶] 愛媛大学 大学院理工学研究科 電子情報工学専攻
{[†]shirai.keisuke.64x@st., [§]forest@i.}kyoto-u.ac.jp
[‡]k.hashimoto@salesforce.com [††] Akiko.Eriguchi@microsoft.com
[¶]ninomiya@cs.ehime-u.ac.jp

1 はじめに

機械翻訳はそのタスクにおいて, 与えられた翻訳元文に対応する翻訳先文を生成するという点で文生成タスクであるといえる. ニューラル機械翻訳（Neural Machine Translation; NMT）は深層学習における機械翻訳モデルであり, 近年盛んに研究されている. しかし, 学習後のモデルによって出力された翻訳文には様々な翻訳エラーが含まれうることが問題として知られている.

近年, 強化学習は様々な文生成タスクに適用されている. 強化学習を用いる利点の一つは, 教師あり学習とは異なる指標を用いてモデルを学習させられる点である. NMT への応用としては, Wu らによる GLEU [1] の他, Gu らによる Gumbel-Greedy Decoding [2] 等が知られている. しかし, これらの手法は必ずしも特定の翻訳エラーの出力を抑制するために提案されたものではないため, そういった効果は期待出来ない.

本論文では, 任意の翻訳エラーの出力を抑制可能な NMT モデルを提案する. このモデルでは, 特定の翻訳エラーを検出する識別モデルを評価関数として扱い, NMT へ強化学習を適用する. 本手法の有効性を示すため, 特定の翻訳エラーの例として繰り返しと欠落を挙げ, これらの翻訳エラーの出力を抑制可能なことを示す. 学習後のモデルの挙動が期待通りであるか評価するため, 機械翻訳タスクの評価指標として一般的な BLEU と METEOR に加え, REP と DROP を用いた. REP と DROP は翻訳文に含まれる繰り返しと欠落の度合いを計るために, Malaviya らによって提案された評価指標である [3]. また, 従来では事前学習時と同じパラレルコーパスを用いて強化学習を行うことが多かったが, 本研究では, よりテスト時に近い挙動を模倣するためにモノリンガルコーパスを用いた実験も行った. 実験結果から, 本手法が特定の翻訳エラーの改善に対して有効であることが示された.

2 研究背景

2.1 ニューラル機械翻訳

NMT は翻訳元文から翻訳先文への変換を直接学習するモデルである. 本研究では, エンコーダとデコーダは 再帰型ニューラルネットワーク（Recurrent Neural Network; RNN）で実装し, アテンション構造は Luong らのモデルを用いる [4]. NMT モデルのパラメータは, 翻訳元文 $\boldsymbol{s} = s_1 s_2 \cdots s_m$ と翻訳先文 $\boldsymbol{t} = t_1 t_2 \cdots t_n$ が与えられたとき, その対数尤度を最大化（Maximum Likelihood Estimation; MLE）するように学習されるが, 誤差関数としては以下のように表現出来る.

$$L_{\mathrm{MLE}} = -\sum_j \log p(t_j | \boldsymbol{t}_{<j}, \boldsymbol{s}). \quad (1)$$

2.2 ニューラル機械翻訳における翻訳エラー

NMT のモデルによる出力文には様々な翻訳エラーが含まれうる. 本実験で用いる開発データを用いて NMT モデルによる翻訳を行ったとき, 繰り返しと欠落が翻訳エラーとして顕著に見られた. 従って, 後述する提案手法ではこれらの翻訳エラーの抑制を目指す.

繰り返しは NMT モデルが連続する 1 個以上の単語を誤って繰り返すことで起こる. 例えば, “since then

* 本研究は著者らが東京大学に在籍中に行われたものである.

, the index has climbed above 10,000 .” という参照訳文に対して, “since then , the index has the index has the index has climbed above 10,000 .” という翻訳文は, 連続する 3 語 “the index has” を 3 回繰り返している為に繰り返しを起こしているといえる.

欠落は NMT モデルが出力した翻訳文において, 参照訳文中に存在する単語が欠落することによって起こる. 例えば, “since then , the index has climbed above 10,000 .” という参照訳文に対して, “since then , the index has climbed 10,000 .” という翻訳文は単語 “above” が参照訳文から欠落しているため, 欠落を起こしているといえる.

3 人工負例による識別器を用いた NMT の強化学習

3.1 強化学習

NMT モデルのパラメータは式 1 の最小化による事前学習の後, 強化学習により再調整される. 強化学習時には REINFORCE を用いて学習を行う. ここでは, NMT モデルをエージェント, カテゴリ分布 $p(t'_j|\boldsymbol{s}, \boldsymbol{t}'_{<j})$ からの単語 t'_j の選択をアクションだと捉えることで強化学習を適用する. NMT モデルによる翻訳文が得られたとき, 以下の誤差関数を最小化することを考える.

$$L_{\mathrm{RL}} = -\sum_j \{\log p(t'_j|\boldsymbol{s}, \boldsymbol{t}'_{<j}) \\ (R(\boldsymbol{s}, \boldsymbol{t}') - b(\boldsymbol{s}, t'_j))\}, \tag{2}$$

ここで, R は報酬関数を, b はベースラインをそれぞれ表す. 式 2 のみでは学習が不安定になるため, 実際には, 次のように教師あり学習による学習シグナルとの重み付け和を最小化する.

$$\alpha\, L_{\mathrm{MLE}} + \beta\, L_{\mathrm{RL}}, \tag{3}$$

ここで, α と β はハイパーパラメータを表し, それぞれの誤差の強さを調節する役割を持つ.

3.2 識別器

強化学習時に用いる報酬関数には, 特定の翻訳エラーが含まれているかを識別し, 適切な報酬を返すことが期待される. これを実現するため, 学習可能なモデルを導入し, 以後このモデルを識別器と呼ぶ. 翻訳元文と翻訳先文のペアが与えられたとき, 識別器は翻訳先文に特定の翻訳エラーが含まれているかを識別する.

識別器は参照訳文 $\boldsymbol{t}$ と翻訳エラーを含む文 $\boldsymbol{e}$ を識別する二値分類器である. 後述する通り, 翻訳エラーを含む文は参照訳文を基に人工的に生成する. 従って, 識別器 D のパラメータは以下の誤差関数を最小化するように学習される.

$$L_{\mathrm{DIS}} = -\,\mathbb{E}_{\boldsymbol{s},\boldsymbol{t}}[\log \mathrm{D}(\boldsymbol{s}, \boldsymbol{t})] \\ -\,\mathbb{E}_{\boldsymbol{s},\boldsymbol{e}}[\log(1 - \mathrm{D}(\boldsymbol{s}, \boldsymbol{e}))], \tag{4}$$

ここで, D は $[0, 1]$ の実数値を返す. 識別器は NMT モデルとは別に学習され, 強化学習時には パラメータを固定した事前学習済みの識別器を $R = D$ とすることで報酬関数として用いる.

3.3 人工負例

識別器は参照訳文と特定の翻訳エラーを含む人工負例文を識別する. 人工負例文は機械翻訳文ではなく参照訳文から生成するが, これは識別器が特定の翻訳エラーを含むかではなく翻訳文の質を基に識別するように学習されることを防ぐためである. 本研究では, 翻訳先文のみから人工負例を生成する. 人工負例を用いることの利点の一つとして, 翻訳エラーを人工的に模倣することが出来れば, 提案手法を適用できることが挙げられる. 本研究では繰り返しと欠落を含む翻訳文を人工的に生成し, 人工負例文として用いる.

人工的な繰り返し文は, 参照訳文の i 番目から連続する j 単語を k 回繰り返すことで生成される. ここで, j は $(1, 2, \cdots, n)$ から, k は $(2, 3, \cdots, r)$ からそれぞれ乱雑に選択される. 例えば, 2.2 節の繰り返し文の例は $(i, j, k) = (4, 3, 3)$ の場合である. 本研究では $(n, r) = (4, 4)$ と設定する. これは乱雑に選択された連続する $1 \sim 4$ 単語を $2 \sim 4$ 回繰り返すことで, 人工的に繰り返し文を含む文を生成することを意味する.

人工的な欠落文は, 参照訳文の i 番目から連続する j 単語を取り除くことによって生成される. ここで, j は $(1, 2, \cdots, n)$ から乱雑に選択される. 例えば, 2.2 節の欠落文の例は $(i, j) = (8, 1)$ の場合である. 本研究では $n = 4$ と設定する. これは乱雑に選択された連続する $1 \sim 4$ 単語を参照訳文から取り除くことで, 人工的に欠落文を生成することを意味する.

4 実験設定

4.1 データ・セット

ASPEC の日英翻訳タスクを用いて実験を行った. 学習データ 300 万文のうち, 先頭 200 万文対を抽出し, パ

ラレルコーパスとして用いた. その際, さらに先頭 10 万文対を小データ, 200 万文対全てを大データとして区別した. また, 後半 100 万文対の翻訳元文をモノリンガルデータとして扱い, 強化学習時にパラレルデータとの差分を考察するために用いた. 一方で, 開発データは 1,790 文対, テストデータは 1,812 文対であった. 英文は `mosesdecoder` を, 和文は `KyTea` をそれぞれ用いて単語分割を行った. 語彙は `SentencePiece` により獲得したが, その語彙サイズは小データに対しては 8,000 を, 大データに対しては 16,000 をそれぞれ選択した. 学習中は学習データのうち, その文長が 1 以上 64 以下の文対のみを用いた他, 未知語は全て ⟨UNK⟩ で置換した.

4.2 モデルパラメータと学習

本実験では一貫して再帰的な構造をもつ NMT を使用し, 双方向 LSTM からなるエンコーダと単方向 LSTM からなるデコーダを用いた. エンコーダとデコーダの層数は同一であり, 小データに対しては 2 層, 大データに対しては 4 層をそれぞれ選択した. 埋め込み層の次元と LSTM の隠れ層の次元は同一であるとし, 小データに対しては 256 を, 大データに対しては 512 をそれぞれ選択した. 一般的に, デコーダは LSTM の最終層から得られた分散表現を語彙サイズの次元に変換する為の全結合層を持つが, 本稿では便宜上これをソフトマックス層と呼ぶことにする. 本実験では Inan らに従い, 翻訳先言語の埋め込み層とソフトマックス層の重みを共有した [5]. これらに加えて, dropout を係数 0.1 で適用し, 勾配は 1.0 でクリップした.

識別器は翻訳元文を読み込み, 分散表現に変換する単方向 LSTM とその最終状態を初期状態とし, 翻訳先文を読み込み実数値に変換する単方向 LSTM を持つ. 従って, 識別器は NMT モデルと似た構造を持つが, 出力が文ではなく実数値であること, アテンション構造を持たないことが違いである. 識別器の持つ LSTM は学習データのサイズに依らず 2 層の LSTM を用い, その隠れ層の次元は 256 とした. また, dropout を係数 0.1 で適用した.

提案手法では, まず NMT モデルと識別器の事前学習を教師あり学習を用いて行い, 次に強化学習を行う. 強化学習時には識別器のパラメータは固定する. 事前学習には Adam を用い, 初期学習率は 1.0×10^{-3} とした. 1,000 イテレーション毎に, NMT は perplexity を, 識別器は精度をそれぞれ評価データ上で評価した. それらの値が悪化した場合には学習率を半減させ, 8 回半減させた時点で事前学習を終了した. 強化学習時には, 確率的勾配降下法を初期学習率 1.0×10^{-2}, モーメンタム 0.9 で用いた. 強化学習時にはモデルの評価は BLEU で行った.

テストデータの翻訳はビームサーチで行い, 小データに対しては幅 5 を, 大データに対しては幅 12 を設定した. また, テスト時の評価の際には WAT'15 に規定されている事後処理を施した.

4.3 評価指標

機械翻訳タスクにおいて一般的な評価指標である BLEU と METEOR に加えて, 本実験では REP と DROP を用いた. REP と DROP はそれぞれ機械翻訳文に含まれる繰り返しと欠落の度合いを計るために Malaviya らによって提案された指標である [3]. BLEU や METEOR と反し, これらの指標は低い値である程良いスコアであることを示す.

REP は翻訳文に含まれる n-gram レベルの繰り返しを評価するが, n は個別に与えられる. そこで, 1-gram から 4-gram の繰り返しを同時に評価するため, これを拡張した eREP を導入する. これは元の REP スコアにおける $\sigma(t, r)$ を次のように修正することで実現される.

$$\sigma(t,r) = \lambda_1 \sum_{n=2}^{4} \sum_{s \in V^n, t(s) \geq 2} \max\{0, t(s) - r(s)\} + \lambda_2 \sum_{w \in V} \max\{0, t(ww) - r(ww)\},$$

ここで, ww は連続した 1-gram を表す. また, $t(s), r(s)$ はそれぞれ, 機械翻訳文, 参照訳文内に存在する n-gram s の数を表している. λ_1, λ_2 は REP 内で導入されているハイパーパラメータであるが, 本実験では $(\lambda_1, \lambda_2) = (1, 1)$ と設定した. また, $\sigma(t, r)$ は参照訳文中に存在する n-gram の数で正規化した.

DROP は翻訳元文中の単語のうち, 翻訳先文中の単語とアライメントによって結び付けられた単語の率を計算する. 本実験では, Malaviya らに従い, アライメントのモデルとして `fast_align` を用いた.

5 実験結果

表 1 に小データにおける実験結果を示す. 表中のパラレルとは強化学習時に事前学習時と同じパラレルデータを用いることを, モノリンガルとはモノリンガルデータを用いることをそれぞれ表す. 提案手法では

	eREP (↓)	DROP (↓)	BLEU	METEOR
ベースライン	8.98	18.76	18.98	27.16
提案手法（パラレル）+ 識別器（繰り返し）	6.33	19.28	19.76	27.08
提案手法（モノリンガル）+ 識別器（繰り返し）	**4.94**	20.34	**20.53**	26.80
提案手法（パラレル）+ 識別器（欠落）	10.58	18.22	18.62	27.37
提案手法（モノリンガル）+ 識別器（欠落）	11.76	**17.95**	18.38	**27.60**

表 1: 小データにおける実験結果.

	eREP (↓)	DROP (↓)	BLEU	METEOR
ベースライン	5.43	17.33	23.97	29.87
提案手法（パラレル）+ 識別器（繰り返し）	5.13	17.86	24.31	29.70
提案手法（モノリンガル）+ 識別器（繰り返し）	**4.63**	17.75	**24.35**	29.63
提案手法（パラレル）+ 識別器（欠落）	5.90	**16.26**	23.64	**30.26**
提案手法（モノリンガル）+ 識別器（欠落）	5.11	16.64	23.86	30.00

表 2: 大データにおける実験結果.

特定の翻訳エラーの人工負例を用いて学習した識別器による強化学習を行ったモデルを表している. 表の結果から, 提案手法は eREP と DROP のスコアをそれぞれベースラインから大幅に改善されていることがわかり, これは提案手法は特定の翻訳エラーの改善に対して有効であることを示している. さらに, 結果から eREP は BLEU と, DROP は METEOR とそれぞれ値の推移に相関があることが見て取れた. また, 強化学習時にモノリンガルデータを用いた場合, パラレルデータを用いる場合以上に精度が改善されていることがわかる. これは, 事前学習時とは異なるデータを用いて NMT モデルに翻訳文を出力させることで, よりテスト時に近い挙動を模倣出来ているからだと考えられる. NMT モデルは事前学習の時点でパラレルデータに対して幾らかの過学習を起こしているため, パラレルデータに対する翻訳文の質は非常に高い. しかし, 強化学習時には識別器の出力を用いて特定の翻訳エラーの出現を抑制したいため, NMT モデルの出力には幾らかの翻訳エラーが含まれることが期待される. 従って, この観点から, 強化学習時にモノリンガルデータを用いるのはパラレルデータを用いる以上に効果的であるといえる.

表 2 に大データにおける実験結果を示す. 結果から提案手法は大データに対しても有効であり, それぞれ eREP, DROP を改善することが確認された. しかし, 小データにおける結果と比較すると, その改善度合いは小さい. これは学習データの量が増加したことにより, NMT モデルによる翻訳文の質も向上したために, より翻訳エラーを起こしにくくなったことが原因として挙げられる.

6 おわりに

本研究では NMT における特定の翻訳エラーに対処するために, 識別器と人工負例を導入し, 強化学習を行う手法を提案した. 実験結果から提案手法は意図した翻訳エラーの出現を抑制可能であることを示した. 今後は Transformer 等の非再帰的な構造を持つ NMT に対する本手法の有効性を調査したい.

参考文献

[1] Y. Wu, M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. “Google ’s neural machine translation system: Bridging the gap between human and machine translation.” In arXiv. (2016).

[2] J. Gu, J Daniel, and O. Victor. “Neural machine translation with gumbelgreedy decoding.” In AAAI. (2018).

[3] C. Malaviya, P. Ferreira, and A. Martins. “Sparse and constrained attention for neural machine translation.” In ACL. (2018).

[4] T. Luong and H. Pham and C. Manning. “Effective approaches to attention-based neural machine translation.” In EMNLP. (2015).

[5] H. Inan, K. Khosravi, and R. Socher. “Tying word vectors and word classifiers: A loss framework for language modeling.” In ICLR. (2017).