敵対的生成ネットワークを用いた機械翻訳評価手法

松村 雪桜　　小町 守
首都大学東京
matsumura-yukio@ed.tmu.ac.jp, komachi@tmu.ac.jp

1 はじめに

近年，ニューラル機械翻訳（NMT）[1, 4] の登場により，機械翻訳が盛んに研究されている．機械翻訳の評価には一般的に BLEU [7] が用いられている．しかしながら，BLEU は n-gram 適合率に基づき精度を評価する手法であり，文の意味を考慮した評価はできていない．

画像生成の分野で注目を集めている敵対的生成ネットワーク（GAN）[2] は，Generator と Discriminator の 2 つのネットワークからなり，Discriminator はあるデータが正解データであるか Generator の出力であるかを識別する一方で，Generator は Discriminator が識別できないようなデータを生成するように敵対的な学習を行うことで，Generator が正解に近いデータを生成することを可能にしている．敵対的生成ネットワークは自然言語処理，とりわけ機械翻訳の分野でも使用が試みられており [8, 9, 10]，これらの研究では Generator をニューラル機械翻訳モデル，Discriminator を入力された原言語文と目的言語文から目的言語文が参照訳であるかシステム出力文であるか予測する分類器として敵対的に学習を行うことで，Generator であるニューラル機械翻訳モデルの精度の向上を図っている．

それに対して本研究では，目的言語文の分類器である Discriminator に注目し，Discriminator が予測する正解データらしさを機械翻訳の評価手法として用いることを提案した．提案手法では，Yang ら [9] の手法を参考にモデルを実装し，Generator と Discriminator の事前学習を行った後に敵対的生成ネットワーク全体を学習した．学習された Discriminator に原言語文とシステムの出力文を入力することで，文単位の翻訳精度を評価する．敵対的生成ネットワークの設定では正解データは人手による参照訳なので，原言語文と目的言語文のペアを見て正解データらしいということは人手による翻訳である可能性が高いということであり，翻訳文の評価に転用できると考えられる．提案手法では評価に正解の参照訳を必要としないため，単言語コーパスなどの参照訳がない文に対する翻訳の評価への使用も期待できる．

Asian Scientific Paper Excerpt Corpus（ASPEC）[6] を使用して学習した敵対的生成ネットワークモデルを用いて日英翻訳の評価に関する実験を行い，ケンドールの順位相関係数を用いて WAT 2015 [5] の人手評価データとの相関を計ったところ，Sentence BLEU [3] とほぼ同程度に人手評価との相関があることがわかった．また，参照訳と同様の意味であるにもかかわらず単語の表層の違いにより Sentence BLEU では低く評価されてしまうシステムの出力文も，提案手法では高く評価することができた．

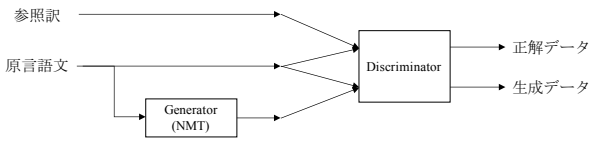

図 1: 敵対的生成ネットワークを用いたニューラル機械翻訳モデル．

2 ニューラル機械翻訳

ここで，本研究に用いたニューラル機械翻訳モデル[1]について説明する．我々は Luong ら [4] が提案したニューラル機械翻訳モデルを基に実装を行った．

2.1 Encoder

入力された原言語文は，one-hot ベクトル系列 ($\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{|\boldsymbol{X}|}]$) に変換される．各ステップ i において，原言語側の単語の埋め込み表現 $\boldsymbol{e}_i^s$ は，

$$\boldsymbol{e}_i^s = \tanh(\boldsymbol{W}_x \boldsymbol{x}_i) \tag{1}$$

と計算される．ここで，$\boldsymbol{W}_x \in \mathbb{R}^{q \times v_s}$ は重み行列であり，q は埋め込み層の次元数，v_s は原言語側の語彙サイズを表す．Encoder の隠れ層 $\bar{\boldsymbol{h}}_i^s$ は，

$$\bar{\boldsymbol{h}}_i^s = \overrightarrow{\boldsymbol{h}_i^s} + \overleftarrow{\boldsymbol{h}_i^s} \tag{2}$$

と表され，$\overrightarrow{\boldsymbol{h}_i^s}$ および $\overleftarrow{\boldsymbol{h}_i^s}$ は，それぞれ LSTM を用いて

$$\overrightarrow{\boldsymbol{h}_i^s} = \mathrm{LSTM}(\boldsymbol{e}_i^s, \overrightarrow{\boldsymbol{h}_{i-1}^s}), \quad \overleftarrow{\boldsymbol{h}_i^s} = \mathrm{LSTM}(\boldsymbol{e}_i^s, \overleftarrow{\boldsymbol{h}_{i+1}^s}) \tag{3}$$

と計算される．

[1] https://github.com/yukio326/nmt-chainer

2.2 Decoder

各ステップ j において，Decoder の隠れ層 $\boldsymbol{h}_j^t$ は，LSTM を用いて，

$$\boldsymbol{h}_j^t = \mathrm{LSTM}([\boldsymbol{e}_{j-1}^t : \tilde{\boldsymbol{h}}_{j-1}^t], \boldsymbol{h}_{j-1}^t) \tag{4}$$

と表される．ここで，$\boldsymbol{e}_{j-1}^t$ は目的言語側の 1 ステップ前の単語埋め込み表現，$\tilde{\boldsymbol{h}}_{j-1}^t$ は 1 ステップ前のアテンション計算後の隠れ層，$\boldsymbol{h}_{j-1}^t$ は 1 ステップ前の隠れ層である．目的言語側の単語埋め込み表現 $\boldsymbol{e}_{j-1}^t$ は，

$$\boldsymbol{e}_{j-1}^t = \tanh(\boldsymbol{W}_y \boldsymbol{y}_{j-1}) \tag{5}$$

と表される．ここで，$\boldsymbol{W}_y \in \mathbb{R}^{q \times v_t}$ は重み行列であり，v_t は目的言語側の語彙サイズを表す．単語 $\boldsymbol{y}_{j-1}$ は，学習時には原言語文と同様に目的言語文を one-hot ベクトル系列 ($\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{|\boldsymbol{Y}|}]$) に変換したものから用い，評価時には予測した単語 $\hat{\boldsymbol{y}}_{j-1}$ を用いる．

アテンション計算後の隠れ層 $\tilde{\boldsymbol{h}}_j^t$ は，

$$\tilde{\boldsymbol{h}}_j^t = \tanh(\boldsymbol{W}_a[\boldsymbol{h}_j^t : \boldsymbol{c}_j] + \boldsymbol{b}_a) \tag{6}$$

と表される．ここで，$\boldsymbol{W}_a \in \mathbb{R}^{r \times 2r}$ は重み行列，$\boldsymbol{b}_a \in \mathbb{R}^r$ はバイアスであり，r は隠れ層の次元数を表す．文脈ベクトル $\boldsymbol{c}_j$ は Encoder の隠れ層 $\bar{\boldsymbol{h}}_i^s$ の重み付き和であり，

$$\boldsymbol{c}_j = \sum_{i=1}^{|\boldsymbol{X}|} \alpha_{ij} \bar{\boldsymbol{h}}_i^s \tag{7}$$

と計算される．上式における重み α_{ij} は，ソフトマックス関数を用いて全体の和が 1 となるよう正規化される確率分布であり，

$$\alpha_{ij} = \frac{\exp(\bar{\boldsymbol{h}}_i^{s\mathrm{T}} \boldsymbol{h}_j^t)}{\sum_{k=1}^{|\boldsymbol{X}|} \exp(\bar{\boldsymbol{h}}_k^{s\mathrm{T}} \boldsymbol{h}_j^t)} \tag{8}$$

のように内積を用いて計算される．出力単語 $\hat{\boldsymbol{y}}_j$ の条件付き確率は，

$$\boldsymbol{p}(\hat{\boldsymbol{y}}_j | \boldsymbol{Y}_{<j}, \boldsymbol{X}) = \mathrm{softmax}(\boldsymbol{W}_g \bar{\boldsymbol{h}}_j^t + \boldsymbol{b}_g) \tag{9}$$

と計算される．ここで，$\boldsymbol{W}_g \in \mathbb{R}^{v_t \times r}$ は重み行列，$\boldsymbol{b}_g \in \mathbb{R}^{v_t}$ はバイアスである．

2.3 学習

学習時における目的関数は，

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{D} \sum_{d=1}^{D} \sum_{j=1}^{|\boldsymbol{Y}|} \log \boldsymbol{p}(\boldsymbol{y}_j^{(d)} | \boldsymbol{Y}_{<j}^{(d)}, \boldsymbol{X}^{(d)}, \boldsymbol{\theta}) \tag{10}$$

のように定義する．ここで，$\boldsymbol{\theta}$ はモデルにおける全てのパラメータとする．

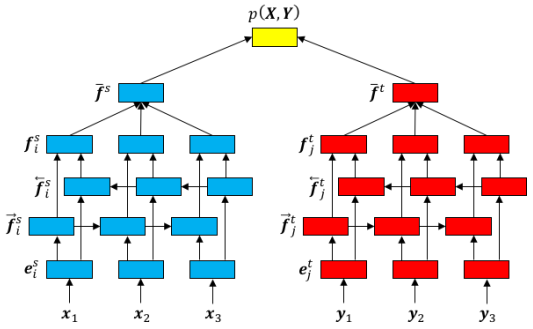

図 2: Discriminator の構造．

3 敵対的生成ネットワークを用いたニューラル機械翻訳

ここでは，提案手法として用いた敵対的生成ネットワークを用いたニューラル機械翻訳モデルについて述べる．我々は Yang ら [9] の提案モデルを参考に実装を行った．敵対的生成ネットワークは，図 1 に示したように，Generator と Discriminator の 2 つのネットワークから成る．

3.1 Generator

Generator は，入力された原言語文から Discriminator がシステムの出力文であると判断できないような正解データ（参照訳）に近い文の生成を図る．本研究では，前節で述べたニューラル機械翻訳モデルをそのまま Generator として用いる．

3.2 Discriminator

Discriminator は，図 2 に示したように，入力された原言語文と目的言語文から，その目的言語文の正解らしさを予測する．すなわち，目的言語文として参照訳が入力された場合は高いスコアを，Generator の出力文が入力された場合は低いスコアを予測することが Discriminator の目的である．

入力された原言語文の各単語は，式 (1) に従ってそれぞれ単語埋め込み表現 $\boldsymbol{e}_i^s$ へと変換される．各ステップ i において，原言語側の単語埋め込み表現 $\boldsymbol{e}_i^s$ に対応する隠れ層 $\boldsymbol{f}_i^s$ は，

$$\boldsymbol{f}_i^s = \left[\overrightarrow{\boldsymbol{f}_i^s} : \overleftarrow{\boldsymbol{f}_i^s}\right] \tag{11}$$

と表され，$\overrightarrow{\boldsymbol{f}_i^s}$ および $\overleftarrow{\boldsymbol{f}_i^s}$ は，それぞれ LSTM を用いて

$$\overrightarrow{\boldsymbol{f}_i^s} = \mathrm{LSTM}(\boldsymbol{e}_i^s, \overrightarrow{\boldsymbol{f}_{i-1}^s}), \quad \overleftarrow{\boldsymbol{f}_i^s} = \mathrm{LSTM}(\boldsymbol{e}_i^s, \overleftarrow{\boldsymbol{f}_{i+1}^s}) \tag{12}$$

と計算される．得られた隠れ層から原言語文の文ベクトル $\bar{\boldsymbol{f}}^s$ を，

$$\bar{\boldsymbol{f}}^s = \mathrm{average}\left(\left[\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, \cdots, \boldsymbol{f}_{|\boldsymbol{X}|}^s\right]\right) \tag{13}$$

のように求める．同様にして，入力された目的言語文の文ベクトル $\bar{\boldsymbol{f}}^t$ を求める．求めた原言語文および目的言語文の文ベクトルの内積を用いて，入力された原言語文に対する目的言語文の正解らしさを，

$$p(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{sigmoid}(\bar{\boldsymbol{f}}^s \cdot \bar{\boldsymbol{f}}^t) \tag{14}$$

のように予測する．

3.3 学習

敵対的生成ネットワークでは，Generator は Discriminator を騙すように，Discriminator は Generator の出力文を区別できるように敵対的な学習を行わなければならない．したがって，予測系列を $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \cdots, \hat{\boldsymbol{y}}_{|\hat{\boldsymbol{Y}}|}]$ とすると，学習時における目的関数は，

Generator

$$\begin{aligned}\mathcal{L}_G(\boldsymbol{\theta}, \boldsymbol{\gamma}) = &\frac{1}{D}\sum_{d=1}^{D}\Bigg\{\sum_{j=1}^{|\boldsymbol{Y}|}\log \boldsymbol{p}(\boldsymbol{y}_j^{(d)}|\boldsymbol{Y}_{<j}^{(d)}, \boldsymbol{X}^{(d)}, \boldsymbol{\theta}) \\ &+ \log p(\boldsymbol{X}^{(d)}, \hat{\boldsymbol{Y}}^{(d)}|\boldsymbol{\gamma})\Bigg\}\end{aligned} \tag{15}$$

Discriminator

$$\begin{aligned}\mathcal{L}_D(\boldsymbol{\gamma}) = &\frac{1}{D}\sum_{d=1}^{D}\Bigg\{\log p(\boldsymbol{X}^{(d)}, \boldsymbol{Y}^{(d)}|\boldsymbol{\gamma}) \\ &+ \log\Big\{1 - p(\boldsymbol{X}^{(d)}, \hat{\boldsymbol{Y}}^{(d)}|\boldsymbol{\gamma})\Big\}\Bigg\}\end{aligned} \tag{16}$$

のように定義する．ここで，$\boldsymbol{\theta}$ は Generator における全てのパラメータ，$\boldsymbol{\gamma}$ は Discriminator における全てのパラメータとする．

4 敵対的生成ネットワークを用いた機械翻訳評価手法

敵対的生成ネットワークを用いる一般的な目的は，Generator は Discriminator がシステムの出力文であると判断できないような正解データに近い文を生成するように，Discriminator は入力された原言語文と目的言語文からその目的言語文の正解らしさを予測し，Generator の出力文を区別できるように敵対的な学習を行うことで，Generator の出力文の質を向上させることにある．本研究では，Generator ではなく Discriminator に注目し，学習された Discriminator を用いて機械翻訳の評価を行うことを提案する．

提案手法では，はじめに Generator（NMT）の事前学習を行い，得られたモデルの出力を用いて Discriminator の事前学習を行う．その後，敵対的生成ネットワーク全体を学習し，学習された Discriminator に原言語文とシステムの出力文を入力することで，翻訳精度を評価する．敵対的生成ネットワークにおいて，Discriminator は Generator の出力文がどれほど正解データに近いかを判断しており，翻訳精度を評価できることが期待される．

また，提案手法では評価に原言語文とシステムの出力文を用いており正解の参照訳を必要としないため，単言語コーパスなどの参照訳がない文に対する翻訳の評価への使用も期待できる．

表 1: ケンドールの順位相関係数

評価手法	スコア
Sentence BLEU	0.234
GAN	0.195

5 文単位の機械翻訳評価実験

5.1 データ

実験に使用したコーパスは，Asian Scientific Paper Excerpt Corpus（ASPEC）[6] である．学習用データに関しては，約 300 万文のうち文アライメントの信頼度上位 150 万文を用いた．日本語の単語分割には形態素解析器 MeCab[2]（バージョン 0.996，IPADIC）を用い，英語の単語分割には Moses[3]の tokenizer.perl を用いた．原言語および目的言語の学習用データから 1 文あたり 60 単語を超える文対を削除したところ，コーパスの文数は学習用 1,456,278 文，開発用 1,790 文，評価用 1,812 文となった．

翻訳の評価には，WAT 2015 [5] の日英機械翻訳の人手評価データを用いた．このデータは，ASPEC の評価用データ中から抽出された 200 組の対訳文と，それぞれに対する 3 システム[4]分の出力文・2 人分の人手評価スコア（1〜5）からなる．人手評価スコアは，2 人分の評価スコアの平均値を正規化して用いた．

5.2 評価手法

本実験では，200 文 ×3 システムの合計 600 文に対して，ベースライン（Sentence BLEU [3]）および提案手法（GAN）を用いて文レベルで評価を行った．なお，ネットワークは，原言語側語彙サイズ 100,000，目的言語側語彙サイズ 30,000，埋め込み層次元数 512，Generator の隠れ層次元数 1,024，Discriminator の隠れ層次元数 1,024，バッチサイズ 128，最適化手法 AdaGrad（初期学習率：0.01）の設定で実験を行った．

5.3 結果

実験で得られたスコアについて，人手評価との相関を測った．相関を示すスコアとして，ケンドールの順位相関係数を使用した．結果は表 1 の通りである．実

[2] https://github.com/taku910/mecab
[3] http://www.statmt.org/moses/
[4] 統計的機械翻訳システム，用例翻訳システム，ルールベース翻訳システムの 3 システム．

表 2: 評価スコア例.

成功例	
原言語文	両 症例 とも 保存 療法 を 施行 し た 。
参照訳	The conservative treatment was applied to both cases .
システム出力文	Both cases underwent conservative treatment .
人手評価スコア：**1.000**，Sentence BLEU スコア：**0.232**，GAN 評価スコア：**0.999**	

失敗例	
原言語文	次世代 ネットワーク 下 で の ネットワーク 管理 システム に対する 要求 を 論じ , NEC により 提供 さ れる 幾つ か の 解決 策 を 紹介 し た 。
参照訳	Demands for the network management systems under the next generation networks are discussed , and some solutions offered by NEC are introduced .
システム出力文	The demand for the network management system under the next generation network is discussed , and some solutions offered by NEC are introduced .
人手評価スコア：**1.000**，Sentence BLEU スコア：**0.644**，GAN 評価スコア：**0.087**	

験の結果，敵対的生成ネットワークを用いた評価は参照訳を用いていないものの，Sentence BLEU とほぼ同程度の人手評価との相関を示した．

6 考察

表 2 に実際の評価スコア例を示す．成功例において，参照訳は受動態の文となっているが，システム出力文は能動態になっている．また，原言語文の “施行した” に対応する動詞が参照訳では “applied”，システム出力文では “underwent” となっており表層が異なるため，Sentence BLEU スコアでは低く評価されているが，GAN 評価スコアでは人手評価と同様にほぼ正解と評価できている．

しかし失敗例においては，参照訳とシステム出力文において意味も表層もほぼ同じであり，人手評価スコアも Sentence BLEU スコアも高い評価ができているにもかかわらず，GAN 評価スコアでは非常に低い評価をしてしまっている．この例のほかにも，特に長文や複文において誤った評価をしている例が多数見受けられた．原因としては，文ベクトルの作成において各単語に対応する隠れ層の平均を用いているため，文長が長い場合には文ベクトルを作成した段階で情報が多く欠落してしまうことが考えられる．特に，述語は文ベクトルを構成する上で重要度が高いと考えられ，複文のように述語が複数ある場合には，文ベクトルを作成した段階で非常に多くの情報が欠落してしまっている可能性が高いと推測される．

7 おわりに

本研究では，敵対的生成ネットワークを用いた機械翻訳評価手法を提案した．実験の結果，参照訳を用いずに機械翻訳の精度評価を行うことができ，Sentence BLEU とほぼ同程度の人手評価との相関が得られた．今後，長文や複文の評価に対応するためのモデルの検討や敵対的生成ネットワークの学習への強化学習の適用によるさらなる翻訳評価精度の向上を図るとともに，他ドメインやニューラル機械翻訳システムの出力文に対する評価も行いたい．

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR*, 2015.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in NIPS*, 2014.

[3] Chin-Yew Lin and Franz Josef Och. ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation . In *Proceedings of COLING*, 2004.

[4] Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of EMNLP*, 2015.

[5] Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. Overview of the 2nd Workshop on Asian Translation. In *Proceedings of WAT*, 2015.

[6] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC : Asian Scientific Paper Excerpt Corpus. In *Proceedings of LREC*, 2014.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, 2002.

[8] Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-yan Liu. Adversarial Neural Machine Translation. *arXiv*, 2017.

[9] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. *arXiv*, 2017.

[10] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of AAAI*, 2017.