ニューラル機械翻訳での訳抜けした内容の検出

後藤 功雄　　田中 英輝

NHK 放送技術研究所

goto.i-es@nhk.or.jp　tanaka.h-ja@nhk.or.jp

1 はじめに

ニューラル機械翻訳（NMT）[12, 1] は流暢な訳を出力できるが，入力文の内容を全て含んでいることを保証できないという問題がある．このため，翻訳結果に入力文の内容の一部が欠落することがある．欠落は単語レベルの内容だけでなく，節レベルの場合もある．NMT による日英翻訳での訳抜けを含む翻訳例を図 1 に示す．訳抜けは，実用で大きな問題となる．

従来の統計的機械翻訳（SMT）[9, 2] は，デコード中に，カバレッジベクトルを使って入力文のどの部分が翻訳済でどの部分が未翻訳であるかを単語レベルで明示的に区別して，未翻訳の部分がなくなるまで翻訳するため，この問題はほとんど起きない．しかし，NMT では対訳間の対応関係は，アテンションによる確率的な関係しか得られないため，翻訳済の単語と未翻訳の単語を明示的に区別することができない．このため，カバレッジベクトルによって訳抜けを防ぐ SMT の方法をそのまま NMT に適用することはできない．入力文中の各単語位置に応じた動的な状態ベクトルを導入して，この状態ベクトルをソフトなカバレッジベクトル（カバレッジモデル）と見なす手法がある [13, 10]．これらの手法は，この問題を軽減できる可能性がある．しかし，未翻訳部分が残っているかどうかを明示的に検出して翻訳の終了を決定しているわけではない．そのため，これらの手法を用いても訳抜けが発生する問題は残る．

我々は，2 種類の統計量に対して，入力文の内容の欠落に対する検出効果を調べる．統計量の 1 つはアテンション確率の累積（累積アテンション確率）である．もう 1 つは，MT 出力から入力文を生成する逆翻訳の確率である．後者は，言語間の単語の対応関係の特定を必ずしも必要とせずに，MT 出力に入力文の内容が含まれているかどうかを推定できるという特徴がある．また，これらの統計量を訳抜けの検出に使う場合に，値をそのまま使う方法と，MT の n-best 出力で負の対数が最小の場合の値との比を用いる方法の 2 つを比較する．さらに，これらの統計量を NMT のリランキングに応用した場合の効果も評価する．日英特許翻訳での NMT の出力 100 文に対して，訳抜けした内容を検出する実験を行った．統計量を直接用いるよりも，n-best 出力で負の対数が最小の場合との比を用いる方が検出精度が高かった．また，逆翻訳確率に基づく検出の方が，累積アテンション確率に基づく検出より効果が高かった．累積アテンション確率と逆翻訳確率を同時に用いると，さらに検出精度が向上することを確認した．累積アテンション確率と逆翻訳確率は，NMT のリランキングで用いた場合に，いずれも BLEU スコアの向上が確認された．そして，2 つを同時に用いるとさらに BLEU スコアが向上することを確認した．

2 ニューラル機械翻訳

ここで，本稿でのベースラインとなる，[1] に基づく NMT を説明する．この NMT は入力文をエンコードするエンコーダーと訳文を生成するデコーダーからなる．

入力文 $\mathbf{x} = x_1, ..., x_{Tx}$ が与えられると，$\mathbf{x}$ の単語位置 j において，エンコーダーは単語の分散表現の行列 E_x を用いて，前向きの LSTM[6] の出力ベクトル $\overrightarrow{h}_j = \mathrm{LSTM}(\overrightarrow{h}_{j-1}, E_x x_j)$ と後ろ向きの LSTM の出力ベクトル $\overleftarrow{h}_j = \mathrm{LSTM}(\overleftarrow{h}_{j+1}, E_x x_j)$ を生成し，それらをつなげたベクトル $h_j = [\overrightarrow{h}_j^\top; \overleftarrow{h}_j^\top]^\top$ を作成する．ここで，x_j は one-hot ベクトルである．

デコーダーは入力 $\mathbf{x}$ を条件とする出力 $\mathbf{y} = y_1, ..., y_{Ty}$ の確率を計算する．$\mathbf{y}$ の単語位置を i で表す．y_i も one-hot ベクトルである．文の確率を個々の単語の確率の積に分解して計算する．

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|y_1, ..., y_{i-1}, \mathbf{x}) \tag{1}$$

各単語の条件付き確率は，次のようにモデル化される．$p(y_i|y_1, ..., y_{i-1}, \mathbf{x}) = \mathrm{softmax}(y_i^\top W_\mathrm{t} t_i)$，$t_i = \mathrm{maxout}(U_\mathrm{s} s_i + U_\mathrm{y} E_\mathrm{y} y_{i-1} + U_\mathrm{c} c_i)$．ここで，$s_i$ は LSTM の状態ベクトルで，c_i は文脈ベクトル，W と U は重み行列，E は単語の分散表現の行列を示す．s_i と c_i は次のようにして計算する．$s_i = \mathrm{LSTM}(s_{i-1}, [c_i^\top; E_\mathrm{y} y_{i-1}^\top]^\top)$，$c_i = \sum_j \alpha_{i,j} h_i$．ここで，

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_j \exp(e_{i,j})}, \tag{2}$$

$$e_{i,j} = v^\top \tanh(W_\mathrm{s} s_{i-1} + W_\mathrm{y} E_\mathrm{y} y_{i-1}) \tag{3}$$

である．v は重みベクトルである．$\alpha_{i,j}$ がアテンション確率を表しており，y_i と x_j との確率的な対応関係をある程度表しているとみなすことができる．

入力	その後 、 第 1 段 から 順に 第 M 段 まで 、 ADC # 1 と ADC # 2 の パイプ ライン ゲイン エラー 補正 を 交互に 繰り返す （ ステップ S 6 と S 7 、 ステップ S 8 と S 9 、 ステップ S 10 と S 11 ） 。
参照訳	After that , the correction of a pipeline gain error of ADC # 1 and ADC # 2 is sequentially repeated alternately from the first stage to the Mth stage (steps S6 and S7 , steps S8 and S9 , steps S10 and S11) .
MT 出力	After that , the pipeline gain error correction of the ADC # 1 and the ADC # 2 is alternately repeated (steps S6 and S7 , steps S8 and S11) .

図 1　訳抜けを含む NMT の日英翻訳結果の例．入力の網掛け部の訳が MT 出力に含まれていない．参照訳の網掛け部は入力の網掛け部に対応する部分を表している．

3　訳抜けした内容の検出

訳抜けの検出に用いる 2 種類の統計量とそれらの利用方法を説明する．[*1]

3.1　累積アテンション確率

高いアテンション確率が割り当てられた原言語単語は訳出された可能性が高く，アテンション確率がほとんど割り当てられなかった原言語単語は訳出されていない可能性が高いと考えられる．そのため，入力文の各単語位置でのアテンション確率の累積は，訳抜け検出の手がかりになると考えられる．入力 x_j の内容が $\mathbf{y}$ から欠落している度合いを表すスコア ATN-P (attention probability) a_j を (2) 式の $\alpha_{i,j}$ を用いて次のように定義する．

$$a_j = -\log\Big(\sum_i \alpha_{i,j}\Big) \tag{4}$$

(4) 式の括弧内は $\mathbf{x}$ の単語位置 j での累積アテンション確率である．i は出力 $\mathbf{y}$ での単語位置を表している．

ただし，本来，原言語単語が対応する目的言語単語を持たない場合や[*2]，1 つの原言語単語が複数の目的言語単語に対応する場合があり，a_j の値がそのまま訳抜けの度合いを表しているとは限らない．そこで，新たなスコア ATN-R (attention ratio) を定義する．ここで，n-best 出力を $\mathbf{y}^1, ..., \mathbf{y}^n$ と表す．まず次の仮定を設定する．

仮定：任意の入力単語の訳の存在　入力中の任意の単語 $x_j, (1 \leq j \leq T_x)$ の訳は、n-best 出力 $\mathbf{y}^d, (1 \leq d \leq n)$ のどこかに存在する．ただし，本来，対応先を持たない原言語単語 x_j を除く．

先と同様に入力 x_j の内容が $\mathbf{y}^d$ から欠落している度合いを表すスコア ATN-P を a_j^d と表す．そして，$\min_d a_j^d$ は訳抜けしていない a_j^d と見なす．入力 x_j の内容が MT 出力 $\mathbf{y}^d$ から欠落している度合いを表すスコア ATN-R r_j^d を次のように定義する．

$$r_j^d = a_j^d - \min_{d'}(a_j^{d'}) \tag{5}$$

この値は確率の比の対数を表している．

[*1] 2 種類の統計量を組み合わせた利用方法は 5.2 節で説明する．
[*2] 例えば，日本語の助詞「を」は英語では対応する語がない．

3.2　逆翻訳確率

目的言語から原言語へ翻訳することを逆翻訳と呼び，MT 出力から入力文を逆翻訳で生成する確率を逆翻訳確率と定義する．入力文の内容が訳抜けしている場合は，MT 出力から訳抜けした内容を表す原言語単語を生成する確率が低くなると考えられる．これを訳抜け検出の手がかりとして利用する．この方法は，言語間の単語の対応関係の特定を必ずしも必要としないという特徴がある．$\mathbf{y}^d$ における x_j に対する逆翻訳確率に基づくスコア BT-P (back translation probability) b_j^d を次のように定義する．

$$b_j^d = -\log(p(x_j|x_1, ..., x_{j-1}, \mathbf{y}^d)) \tag{6}$$

入力文の内容の一部が欠落している MT 出力では，欠落部分に対応する原言語単語の生成確率が，欠落していない MT 出力から生成した原言語単語の生成確率より小さくなることが期待される．これらの比較により各 MT 出力での訳抜けを検出できると考えられる．ここで前節の “任意の入力単語の訳の存在” を仮定する．そして，$\min_{d'}(b_j^{d'})$ を x_j の内容が MT 出力に含まれている場合のスコアと見なす．入力 x_j の内容が $\mathbf{y}^d$ から欠落している度合いを表すスコア BT-R (back translation ratio) q_j^d を次のように定義する．

$$q_j^d = b_j^d - \min_{d'}(b_j^{d'}) \tag{7}$$

この値は確率の比の対数を表している．

4　翻訳スコアへの適用

前節で説明したスコア r_j^d, q_j^d は MT の n-best 出力から訳抜けの少ない出力を選択するのに役に立つと考えられる．そこで，これらを n-best 出力のリランキングに用いる翻訳スコアに使う．r_j^d を使う時は次式の翻訳スコアを使う．

$$\log(p(\mathbf{y}^d|\mathbf{x})) - \beta \sum_j r_j^d \tag{8}$$

β は重みである．r_j^d は訳抜けの程度を表しており，これを翻訳の尤度から引いている．q_j^d を使う時は，r_j^d の部分を q_j^d に置き換える．なお，リランキングでは同じ入力に対する出力間の比較となるため，ATN-R と ATN-P のリランキング結果は同じになる．同様に BT-R と BT-P

のリランキング結果も同じになる．ここでは ATN-R と BT-R を用いる．

さらに，r_j^d と q_j^d を同時に使う場合は，重み γ, λ を用いて次の値を用いる．

$$\log(p(\mathbf{y}^d|\mathbf{x})) - \gamma \sum_j r_j^d - \lambda \sum_j q_j^d \quad (9)$$

5 実験

長い文を含む日英翻訳での訳抜け検出での効果と翻訳への効果を調べた．

5.1 設定

NTCIR-9 と NTCIR-10 の日英特許翻訳タスク [5, 4] のデータを用いた．訓練データの対訳文は，約 320 万文対である．このうち 100 単語以下の文を日英翻訳の訓練に用いた．計算コスト削減のため，逆翻訳のモデルの訓練には 50 単語以下の文を用いた．開発データは 2,000 文のうち 1,000 文を用いた．テストデータは，NTCIR-9 は 2,000 文，NTCIR-10 は 2,300 文である．英語の単語分割には Stepp tagger[*3]，日本語の単語分割には Juman 7.01[*4]を用いた．

NMT のシステムとして，Kyoto-NMT [3] を用いた[*5]．原言語と目的言語の語彙数はそれぞれ頻度が高い 30,000 語を用い，それ以外の語は特別な語（UNK）に置き換えた．エンコーダーの前向きと後ろ向きの LSTM のユニット数はそれぞれ 1,000，デコーダーの LSTM のユニット数は 1,000，単語の分散表現のベクトルサイズは 620，出力層の前のベクトルの次元は 500 とした．これらの設定は [1] と同じである．ミニバッチサイズには 64 を用いた．ただし，逆翻訳のモデルの訓練には 128 を用いた．パラメータの推定には adam[7] を用いた．モデルの訓練は 6 エポック行った．開発データは，訓練中で BLEU スコアが高いモデルを選択するために用いた．翻訳の探索は，ビーム幅 20 で実施した．出力長は入力長の 2 倍以下に制限した．ビーム探索中に文末記号まで出力された全ての出力を n-best の出力として用いた[*6]．4 節の重み β, γ, λ は開発データで BLEU スコアが高くなるように $\{0.1, 0.2, 0.5, 1, 2\}$ から選択した．

5.2 入力文の内容の訳抜け検出

NTCIR-10 のテストデータを NMT で英語に翻訳し，訳抜けした内容を人手で特定した．そして，入力文の内容の欠落に対する検出効果を比較した．次の手順で評価用データを作成した．まず NTCIR-10 のテストデータからテスト文（日本語）とその参照訳の長さがいずれも 100 単語以下のテスト文を NMT で英語に翻訳した．MT 出力には各テスト文の 1 位の出力を用いた．訳抜けがありそうな MT 出力を選ぶため，(MT 出力の長さ / min(参照訳の長さ, 原文の長さ)) の値が小さいテスト文から順番に選択し，テスト文中で翻訳されていない内容語に人手でタグ付けした．これによって 100 文を選択し，選択した文の全単語 4,457 語のうち 632 語の訳抜けした内容語を認定した．

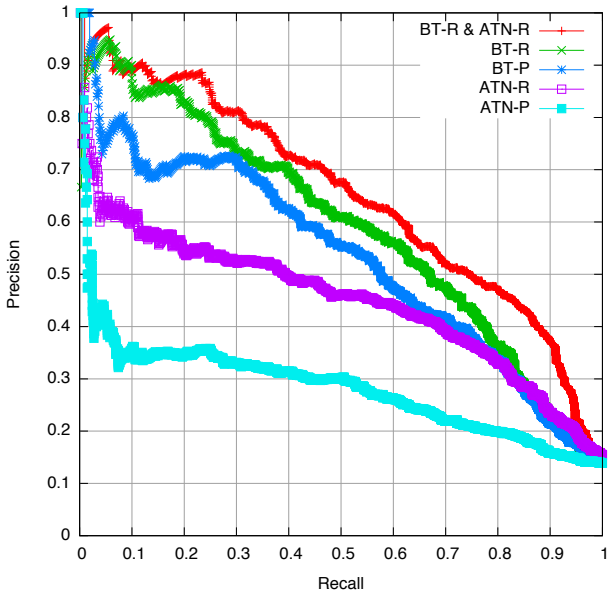

図 2　検出結果

100 文を選択する際，テスト文中で訳抜けした内容語を特定できなかった文は除いた．また，内容語は，漢字，数字，カタカナ，アルファベットのいずれかの文字を含む語とした．なぜなら，日本語では平仮名は文中で主に機能的表現に用いられ，品詞が動詞などの場合でも特許やビジネスの文章では平仮名で表記されるもの（例えば「する」など）の多くは形式的な働きが強く実質的な内容を持たないためである．さらに，接続詞は英訳時にしばしば省略されるため，接続詞も除いた．

r_j^d と q_j^d を同時に用いて訳抜けを検出する場合（BT-R & ATN-R）は，5.1 節で選択した (9) 式の重み γ, λ を用いて，$\gamma r_j^d + \lambda q_j^d$ をスコアとして用いた．

正解を付与した 100 文のテスト文中の全ての語を 3 節のそれぞれのスコアに基づいてランキングし，正解（632 語）と比較した．結果を図 2 に示す．この結果から，次のことが確認された．

- ATN-R は ATN-P より検出精度が高く，BT-R は BT-P より検出精度が高い．
- 逆翻訳確率 (BT-R) は累積アテンション確率 (ATN-R) より検出精度が高い．
- ２種類のスコアを同時に利用する (BT-R & ATN-R) とそれぞれのスコア (BT-R, ATN-R) のみを利用する場合より検出精度が高い．

ここで，BT-R で検出しにくかった例を図 3 に示す．入力文に同じ内容を表す語（ISO）が 2 回出現している．

[*3] http://www.nactem.ac.uk/enju/index.html
[*4] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?JUMAN
[*5] 3 式に合うように変更した．
[*6] n は入力によって異なる．

入力	**ISO** 感度値が小さいときには増幅度が小さく、**ISO** 感度値が大きいときには増幅度が大きい。
参照訳	The amplification is small when the **ISO** sensitivity value is low , while the amplification is large when the **ISO** sensitivity value is high .
MT 出力	When the **ISO** sensitivity value is small , the gain is small .

図 3　逆翻訳で訳抜けを検出しにくかった例．網掛け部が訳抜けした部分を表している．

表 1　翻訳結果（BLEU）

	NTCIR-10	NTCIR-9
Phrase-based SMT	30.58	30.21
Hierachical phrase-based SMT	31.99	31.48
NMT Baseline	38.68	37.83
Rerank with ATN-R	39.82	38.88
Rerank with BT-R	40.14	39.16
Rerank with ATN-R & BT-R	40.36	39.46
NMT with coverage model	38.89	37.90

出力中に “ISO” が含まれているために，BT-R では入力文中の下線の “ISO” が訳抜けしていることを検出しにくかったためと考えられる．逆翻訳確率では，MT 出力での “ある内容語” の有無に対する感度は高いと考えられるが，MT 出力に “ある内容語” が出現したときに，その数がいくつであるかにはそれほど感度が高くないと考えられる．それに対して，ATN-P では，累積確率は単語の出力数に依存して値が増えるため，翻訳結果に “ある内容語” が出現したときに，その数がいくつであるかの影響は現れやすいと考えられる．そして，ATN-R は ATN-P と同じ性質がある．すなわち，BT-R と ATN-R は一部において相補的な関係にあるため，両方のスコアを使うことで，性能が向上したと考えられる．

5.3　n-best 出力のリランキング

Section 4 の (8) 式と (9) 式のスコアに基づいて，MT の n-best 出力をリランキングし，翻訳への効果を調べた．比較のために，ベースライン NMT システムにカバレッジモデル [10][*7]を導入した結果も計算した．[10] では，カバレッジモデルに GRU を用いているが，ここでは LSTM を用いた[*8]．また，参考として，Moses で設定を distortion-limit=20，max-chart-span=1000 とした SMT による結果も計算した．

表 1 に BLEU-4 [11] の結果を示す．表 1 の NMT Baseline と NMT with coverage model とでは差があまりないことから，このデータセットではカバレッジモデルの効果はあまりみられなかった．それに対して，Section 4 のスコアを用いた場合には，NMT Baseline と比較していずれも 1 BLEU ポイント以上の向上が得られており，効果があることが確認できた．

要素毎の効果について議論する．ATN-R と BT-R のどちらもリランキングに効果があった．BT-R のほうが ATN-R よりスコアが少し高かった．ATN と BT の両方を用いるとそれぞれを単独で用いるよりもスコアが高かった．これらの結果は 5.2 節の結果と一致する．BR-R と ATN-R & BT-R の差は，bootstrap resampling test[8] で $\alpha = 0.01$ で統計的に有意であった．

6　おわりに

NMT での訳抜けした内容の検出について，2 種類の統計量の効果を評価した．さらにこれらを NMT の n-best 出力のリランキングに用いた場合の有効性を確認した．NMT の性能が向上すれば，任意の入力単語の訳が存在する仮定が満たされる可能性が高まり，訳抜けした内容の検出精度の向上が期待できる．

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] David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, Vol. 33, No. 2, pp. 201–228, 2007.

[3] Fabien Cromieres. Kyoto-nmt: a neural machine translation implementation in chainer. In *Proceedings of COLING 2016*, pp. 307–311, Osaka, Japan, December 2016.

[4] Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of NTCIR-10*, pp. 260–286, 2013.

[5] Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proceedings of NTCIR-9*, pp. 559–578, 2011.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ArXiv*, 2014.

[8] Philipp Koehn. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pp. 388–395, 2004.

[9] Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, 2003.

[10] Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. Coverage embedding models for neural machine translation. In *Proceedings of EMNLP 2016*, pp. 955–960, Austin, Texas, November 2016.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL 2002*, pp. 311–318, 2002.

[12] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of NIPS 2014*, pp. 3104–3112, 2014.

[13] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of ACL 2016*, pp. 76–85, 2016.

[*7] 競合する手法ではなく，協調して利用できる手法である．

[*8] NMT ベースラインで GRU より LSTM を用いた方が BLEU スコアが高く，カバレッジモデルで chainer の GRU を使うより chainer の LSTM を使った方が学習速度が速かったためである．