共参照解析を利用した複数文翻訳モデルの提案

大谷 拓海[1]　上垣外 英剛[1]　永田 昌明[2]　奥村 学[1]
[1] 東京工業大学　[2] NTT コミュニケーション科学基礎研究所

1 はじめに

近年の機械翻訳システムは，再帰的ニューラルネットワーク (RNN) などを用いる Sequence-to-Sequence モデル (Seq2Seq モデル)[1] に基づく手法によって，翻訳精度が飛躍的に向上している．現在提案されている Seq2Seq モデルに基づく手法の多くは，いずれも文を一つずつ独立に翻訳することを前提に設計されている．

しかし，実際の文は多くの場合，文書という，より大きな単位の要素として存在し，文を正確に解釈するためには前後の文を参照して文脈を考慮する必要がある．これはすなわち，文が単独では意味的に完結していないということであり，文単位で翻訳を行う従来のモデルでは，文間関係から得られる有益な情報を利用できないことを意味している．

このような問題を解決するために，翻訳する文にその直前の文を連結して入力する手法 [2] も提案されているが，Seq2Seq モデルにおける注意機構の計算量が入力長の二乗であることから，計算時間や使用メモリ量が増大してしまうという問題が存在する．このことは，より多くの文を入力とし，広範囲の文間関係を考慮する上で障害となる．さらに，入力中の全単語が一様に扱われることは，文間関係を適切に捉えることを困難にしている可能性がある．

本稿では，入力文書に予め共参照解析を行い，その結果をエンコーダで利用することで，前後の文の情報を効果的に取り込むモデルを提案する．提案モデルでは，文間関係を考慮する上で重要性が高い情報を，共参照解析器を介してモデルが直接参照できるため，全ての入力情報に対して注意を行う必要がない．このような特徴により，提案モデルは従来のモデルよりも多くの文を入力として扱うことが可能になり，複数文に対する翻訳精度の向上が期待できる．さらに，提案モデルでは従来のモデルと同様の性能を維持しつつ，より計算量を抑えることが可能である．

OpenSubtitles2018[6] の英日方向の翻訳を対象とした実験において，提案モデルは直接文を連結して翻訳を行う従来のモデルに対して，BLEU スコアで最大 0.8 ポイントの統計的に有意な改善を観測し，Seq2Seq モデルに基づく翻訳における，共参照解析結果を用いることの重要性を示した．

2 Seq2Seq モデルに基づく翻訳

本節では，Bahdanau ら [1] によって提案された標準的な Seq2Seq モデルの構成について説明する．Seq2Seq モデルは，入力文 $\mathbf{x} = (x_1, \cdots, x_{T_x})$ を以下のように隠れ状態へとエンコードする．

$$h_t = f(x_t, h_{t-1}) \tag{1}$$

ここで，T_x は入力文の長さ，x_t は文の t 番目の単語の分散表現，h_{t-1} は RNN の隠れ状態を表す．エンコーダの出力系列 $(h_1, \cdots, h_{T_x})$ を用いて，出力される文 $\mathbf{y} = \left(y_1, \cdots, y_{T_y}\right)$ の確率は

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T_y} p\left(y_t | \{y_1, \cdots, y_{t-1}\}, \mathbf{x}\right) \tag{2}$$

のように表現される．ここで，右辺の条件付き確率は RNN を用いて以下のようにモデル化される．

$$p\left(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}\right) = g\left(y_{i-1}, s_i, c_i\right) \tag{3}$$

s_i は RNN の時刻 i の状態であり，以下のように計算される．

$$s_i = f\left(s_{i-1}, y_{i-1}, c_i\right) \tag{4}$$

c_i は注意によって生成される文脈ベクトルであり，以下のように生成される．

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{5}$$

ここで α_{ij} は h_j に対する時刻 i の注意であり，アライメント関数 a を用いて以下のように表現される．

$$\alpha_{ij} = a\left(s_{i-1}, h_j\right) \tag{6}$$

3 提案モデル

本節では，前節で述べた Seq2Seq モデルを改良した提案モデルの詳細について説明する．提案モデルは，前節で紹介した Seq2Seq モデルのエンコーダを拡張したものになっており，翻訳対象の文だけではなく，その前後の文を共参照解析結果と共に入力することで，文間情報を効果的に活用することが可能である．

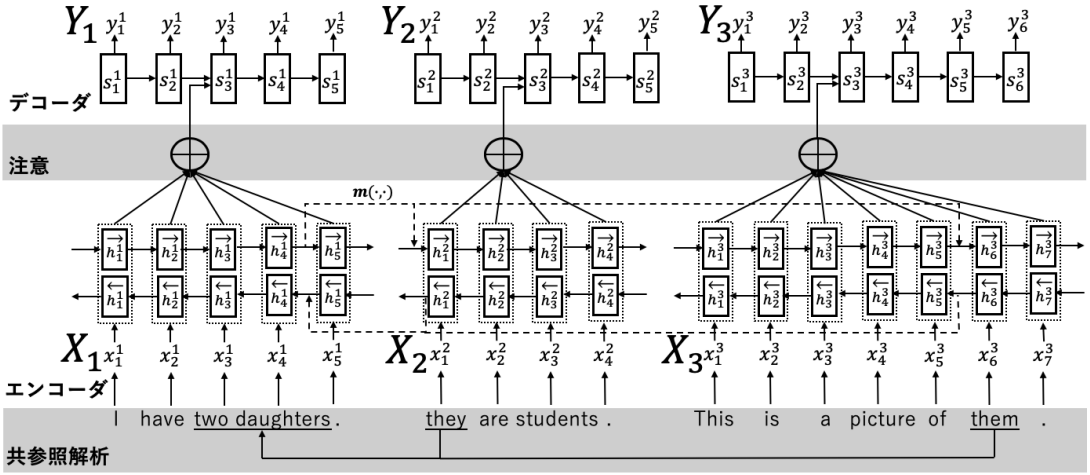

図 1: 提案モデルのネットワーク構成図

図 1 に提案モデルのネットワーク構成図を示す．提案モデルでは，入力となる複数文は事前に共参照解析され，その結果と共にエンコーダへと入力され，注意とデコーダを介して最終的な翻訳結果が出力される．次節以降では，これらの各手順についての詳細を説明する．なお，以降の説明では，入力される N 文を $(X_1, \cdots, X_N)$，X_i の j 番目の単語を x_j^i と表記する．

3.1 共参照解析

まず翻訳対象となる複数文 $(X_1, \cdots, X_N)$ は連結して共参照解析器に入力される．文 X_i の長さを T_i とすると連結後の単語列は

$$(x_1^1, \cdots, x_{T_1}^1, x_1^2, \cdots, x_{T_2}^2, \cdots, x_1^N, \cdots, x_{T_N}^N)$$

と表記される．共参照解析器は N_c 個の共参照 $(c_1, \cdots, c_{N_c})$ を以下のように抽出する．

$$c_k = (source_k, target_k)$$

ここで $source_k$ は共参照の代表表現となる区間であり，$target_k$ は代表表現を参照する区間である．一般に共参照は一つの代表表現に対して複数の参照があるため，共参照には同一の $source_k$ が複数回現れることがある．区間 $source_k$ の先頭の単語を $head(source_k)$，末尾の単語を $tail(source_k)$ と表記した際に，x_j^i が $x_{j'}^{i'}$ を参照するとは，ある c_k が存在して，条件

$$\begin{cases} x_{j'}^{i'} = tail(source_k) \\ x_j^i = head(target_k) \end{cases}$$

を満たすことである．これらの参照関係の具体例を図 2 に示す．

さらに，単語 x_j^i が参照する単語のインデックスの集合を $ref(x_j^i)$ と表記する．単語が参照する先は高々

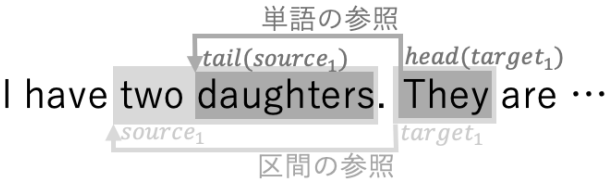

図 2: 共参照解析結果の例

1 単語であるから，$ref(x_j^i)$ の要素数は 1 か 0 である．さらに，$ref(x_j^i)$ は参照する単語の位置が i より前方に存在するかどうかで，以下のように前方照応 ref_f と後方照応 ref_b に分割される．

$$ref_f(x_j^i) = \{(i', j') \in ref(x_j^i) | i' < i \vee (i' = i \wedge j' < j)\}$$
$$ref_b(x_j^i) = \{(i', j') \in ref(x_j^i) | i' > i \vee (i' = i \wedge j' > j)\}$$

$(i', j') \in ref(x_j^i)$ は x_j^i が $x_{j'}^{i'}$ を参照していることを表している．提案手法では，前方照応 ref_f，後方照応 ref_b 共に，エンコーダのネットワーク構造の決定に使用される．

3.2 共参照関係に基づくエンコーダ

本節ではエンコーダにおける共参照関係の利用について説明する．従来の Seq2Seq モデルに基づく翻訳と同様に，提案手法のエンコーダは順方向 LSTM と逆方向 LSTM により構成されている．各入力文 $X_i = (x_1^i, \cdots)$ に対して，順方向のエンコーダは，単語 x_t^i の順方向隠れ状態ベクトル $\overrightarrow{h}_t^i$ を，直前の隠れ状態 $\overrightarrow{h}_{t-1}^i$ と，x_t^i が前方照応する単語の集合 $ref_f(x_t^i)$ に基づく隠れ状態ベクトルの結合関数 $m(\cdot, \cdot)$ を用いて，

$$\overrightarrow{h}_t^i = \overrightarrow{LSTM}\left(x_t^i, m(\overrightarrow{h}_{t-1}^i, ref_f(x_t^i))\right) \quad (7)$$

のように計算する．前方照応に基づく隠れ状態ベクトルの結合関数 $m(\cdot, \cdot)$ として，本研究では以下の二種類を提案する．

- **Coref-mean:** 対象となる隠れ状態の平均値を隠れ状態の結合結果として扱う手法．次式により定義される．

$$m(\overrightarrow{h}^{i}_{t-1}, ref_f(x^i_t)) = \frac{1}{|ref_f(x^i_t)|+1}(\overrightarrow{h}^{i}_{t-1} + \sum_{(i',j')\in ref_f(x^i_t)} \overrightarrow{h}^{i'}_{j'}) \quad (8)$$

- **Coref-gate:** 対象となる隠れ状態の重み付き和を隠れ状態の結合結果として扱う手法．次式により定義される．

$$m(\overrightarrow{h}^{i}_{t-1}, ref_f(x^i_t)) = \overrightarrow{h}^{i}_{t-1} + \sum_{(i',j')\in ref_f(x^i_t)} \beta^{i'}_{j'} \odot \overrightarrow{h}^{i'}_{j'} \quad (9)$$

$\odot$ は次元ごとの要素積を表す．ここで $\beta^{i'}_{j'}$ は $\overrightarrow{h}^{i'}_{j'}$ の重要度を表し，以下のように計算される．

$$\beta^{i'}_{j'} = sigmoid(W_t \overrightarrow{h}^{i'}_{j'} + W_s \overrightarrow{h}^{i}_{t-1}). \quad (10)$$

W_t，W_s は重み行列を表す．

このように，関数 $m(\cdot,\cdot)$ は，参照する単語をエンコードした直後の時点での隠れ状態を，直前の時刻での隠れ状態に結合する処理を実現している．この計算において ref_f に含まれるすべての単語の位置は，現在の時刻の単語よりも前方に存在するため，順方向 LSTM によりエンコードが可能である．

なお逆方向のエンコードは，逆方向 LSTM と後方照応 ref_b とを用いて，順方向と同様の手順で実行される．これらの処理により，エンコーダは単語列とそれらが持つ前方照応，後方照応の情報を隠れ状態として表現している．

最終的に，エンコーダは順方向 LSTM と逆方向 LSTM が出力する各時刻の隠れ状態ベクトルを結合し，両方向の隠れ状態ベクトル $h^i_t = [\overrightarrow{h}^{i}_{t}; \overleftarrow{h}^{i}_{t}]$ を計算する．

3.3 デコーダ

入力された複数文それぞれに対して，エンコードの結果，隠れ状態ベクトルが，$\mathbf{h^i} = (h^i_1, \cdots, h^i_n)$ のように出力される．デコーダは式 (3) に従って従来の Seq2Seq モデルに基づく翻訳と同様に，各文を独立にデコードする．各文を独立にデコードする際には，入力中の注意の対象は翻訳対象となっている文に限定される．提案手法では，実際の翻訳時に翻訳対象となっている一文のみをデコードすれば良いが，学習時には，学習結果に偏りが生じないようにするため，入力中に含まれる全ての文を同時に翻訳している．

	文数			
	1	*2*	*3*	*5*
Coref-mean	7.8	8.1	8.2	**8.7**
Coref-gate	**8.1**	**8.3**	**8.6**	**8.7**
Concat(baseline)	7.6	7.9	7.9	7.7

表 1: 入力文数に対しての各モデルの BLEU スコア．太字は baseline 手法から統計的に有意な改善 ($p < 0.05$) をしていることを表している．

4 実験

4.1 実験設定

実験には OpenSubtitles2018[6] を使用し，英日方向の翻訳で評価した．元データセットから連続する複数文を一つのまとまりとして切り出し，その中からランダムに選択した 2000 データをテストデータとし，残りの約 187 万データを訓練データとして用いた．

英語の共参照解析には NeuralCoref[1] を利用した．日本語の単語分割には MeCab[2] を使用し，語彙数の上限を 32000 とした．なお，MeCab の辞書として NEologd[7][3] を使用した．

入力，出力側両方の単語埋め込み，LSTM の隠れ層，注意の次元数は 500 次元に設定し，エンコーダは 2 層双方向 LSTM，デコーダは 2 層 LSTM で構成される．重みの初期値については，-1 から 1 の範囲で一様分布でランダムにサンプリングした [3]．

パラメータの更新には Adam [4] を使用し，学習率は 0.001，$\beta = (0.9, 0.999)$ に設定した．学習は経験的に十分収束する回数として訓練データ全体に対し，20 万 step 行った．ミニバッチのサイズは 32 に設定し，ミニバッチごとに誤差を平均した．なお，ミニバッチの順序は学習時にランダムに入れ替えた．モデルの実装には Pytorch を使用した．

モデルへの入力は翻訳対象文と，その直前の n-1 文を入力した．入力文数の変化によるスコアの変化を観察するために n を $\{1, 2, 3, 5\}$ と変化させて実験した．評価には，翻訳対象文を MeCab で分割し，単語単位の BLEU スコアを用いた．

ベースラインモデルとして，Bawden ら [2] が提案した，複数の文を直接結合して入力する手法 (Concat) を使用し，提案手法において隠れ状態の結合に Coref-mean を使用したモデルと Coref-gate を使用したモデルそれぞれとの比較を行った．各モデルのパラメータ数はベースラインと Coref-mean では 111,057k，Coref-gate では 111,558k となっている．

[1] https://github.com/huggingface/neuralcoref
[2] http://taku910.github.io/mecab/
[3] https://github.com/neologd/mecab-ipadic-neologd

入力	It's rosa. / Rosa, if you're looking for marika, she's not with me.
正解出力	ローザよ / マリカだったら，ここには居ないよ
Coref-gate	ローザよ / **ローザ**，彼女は俺の仲間じゃない
Coref-mean	ローザだ / <unk> を探してるなら彼女は一緒じゃない
Concat(baseline)	ローザだ / **ロサ**，もし君が <unk> を探しているなら彼女は私とは違う
入力	It took some doing, but now we can download. / This beast has all the Men of Letters files.
正解出力	手こずったけどダウンロード出来るわ / この子には MoL の全ファイルが入ってる
Coref-gate	**それはいくつかの ことをしていましたが，今は，ダウンロードできます．** / この獣は MoL の ファイルを持っている
Coref-mean	**でも今はローンが取れる** / この野獣は MoL のファイルを持っている
Concat(baseline)	**何かやったが今は受信 できる** / この獣は MoL のファイルを持っ ている

表 2: テストデータ中の従来手法と提案モデルの翻訳例．入力文の下線は共参照関係にある単語を示している．（上段）Coref-gate の出力では，人名 (rosa) の表記が文を跨いで統一されている．（下段）共参照解析の結果が間違っており，Coref-mean の出力が悪化している．

4.2 実験結果

表 1 に実験結果を示す．検定にはブートストラップ法 [5] を使用した．提案手法 Coref-mean と Coref-gate はベースライン手法に比べ，すべての入力文数において，一貫して高い BLEU スコアを示している．入力文数が $n = 1$ の場合においても提案手法がベースラインを上回っている．この差は，提案手法が文内の共参照についても利用可能であることから生じていると考えられる．入力文数が $n = 2$ の場合では，いずれのモデルにおいても $n = 1$ の場合よりも BLEU スコアが高い．この傾向は，Bawden ら [2] の結果と一致している．一方で，従来試みられていなかった，文脈となる文数が二文以上となる $n > 2$ の設定では，ベースライン手法では $n = 3$ で BLEU スコアの向上が停止し，$n = 5$ においては逆にスコアが低下してしまっていることが分かる．これは，入力となる文が多くなることで，注意の要素が増えてしまい，ベースライン手法が周辺文からの適切な情報をうまく取り込めなくなっていることを示唆している．一方で，提案手法では入力文数の増加に伴い，$n = 5$ となるまで単調なスコアの向上が見られる．この結果は，提案手法では多くの文が入力される状況においても，効果的に翻訳に必要な情報を抽出可能であることを示している．また，Coref-gate のスコアは Coref-mean のスコアと比較して $n = 1, 2, 3$ の場合で上回っている．これは，Coref-gate が各隠れ状態に対して適切に重みを推定することで，効果的に情報の取捨選択を行えていることを示している．

表 2 に，テストデータ中の実際の翻訳例を示す．表 2 の上段の例では，人名である “rosa” が二文に跨って共参照関係にある．ベースライン手法では “rosa” の表記が文ごとに異なり統一されていないが，Coref-gate の出力では改善されていることが分かる．この例では，共参照関係を明示的に利用することによる表記の一貫性の向上が示されている．表 2 の下段の例では，Coref-mean は共参照解析器の誤りの影響を受けて，一つ目の文に対して誤った翻訳結果を出力してしまっている．一方で，Coref-gate では，共参照解析の誤りの影響を受けず，正しい翻訳結果が出力されている．このことから，前処理において共参照解析の誤りが生じたとしても，Coref-gate のゲートは，その情報の重要度を下げることで適切な対処が可能であることが分かる．

5 まとめ

本稿では，共参照関係を考慮して前後の文の情報を効果的に取り込むことが可能な Seq2Seq モデルに基づく翻訳モデルを提案した．実験の結果，提案モデルは従来手法と比較し，多くの文を入力とした際により翻訳精度が向上することを示した．この結果から Seq2Seq モデルに基づく翻訳モデルにおいて，共参照関係を考慮することは，実際に翻訳精度の向上に寄与することが示された．今後は，LSTM 以外の Seq2Seq モデルに基づく翻訳モデルを対象とした提案手法の有効性を調査することを課題としたい．

参考文献

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

[2] Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. Evaluating discourse phenomena in neural machine translation. In *NAACL-HLT*, 2018.

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the EMNLP2004*.

[6] Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. Opensubtitles 2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC 2018), Miyazaki, Japan.(accepted)*, 2018.

[7] Toshinori Sato, Taiichi Hashimoto, and Manabu Okumura. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In *NLP*, 2017.