英日同時通訳におけるニューラル機械翻訳の検討

帖佐 克己　　須藤 克仁　　中村 哲
奈良先端科学技術大学院大学 先端科学技術研究科 情報科学領域
{k-chousa, sudoh, s-nakamura}@is.naist.jp

1 はじめに

同時通訳は文の入力が終了する前にその文の通訳を行うタスクである．同時通訳を行うことで音声によって行われる講演や会話などをリアルタイムに理解する助けとなり，円滑なコミュニケーションを促進することができるようになる．これまでにも機械翻訳システムによる同時通訳手法として，統計的フレーズベース機械翻訳のフレーズテーブルを用いて翻訳単位を短くする手法 [1] などが試みられている．

従来の機械翻訳システムでは文の終端が来るまで翻訳を行わない．しかし，講義や講演のような話し言葉での文章では 1 文が長くなる傾向があるため，その場合だと翻訳結果が得られるまでにかなりの遅延が発生してしまう．また，話し言葉ではしばしば文同士の境界が曖昧になることがあり，文境界の検出を行う必要性が発生する．この結果として，複数文が結合されたものや不完全な文が翻訳器への入力として与えられる場合があり，文の終端が来るまで翻訳を行わないことを仮定している従来の機械翻訳システムでは学習時と異なった環境で翻訳を行うこととなってしまう．これらの問題に対して，これまでの機械翻訳による同時通訳手法では，文を小さいチャンクに分割して翻訳することにより翻訳結果が得られるまでの時間を削減する試みがなされてきた．しかし，逆に遅延を小さくすればするほど翻訳精度は下がってしまうため，同時通訳システムを構築する際には翻訳を行うタイミングを適切に決定し，遅延と翻訳精度との間のトレードオフを調整する必要がある．また，英語と日本語のような語順が大きく異なる言語対での翻訳は特に遅延が大きくなる傾向にあるため，これらの言語対での同時通訳は難易度が高いと考えられている．この場合，語順の入れ替わりが可能な限り減らせるような訳出をするなどの解決策が考えられる．

この問題を解決するニューラル機械翻訳（Neural Machine Translation ; NMT）モデル [2, 3] の研究としていくつかの手法が提案されている．Gu et al. [4] は，既存の翻訳システムに対して 1 単語を入力する *READ* と 1 単語を訳出する *WRITE* の 2 つのアクションを定義し，各タイミングにおいてシステムがどちらのアクションを行うべきなのかを決定する分類器を強化学習によって学習する手法を提案している．また，Alinejad et al. [5] ではこの手法を拡張し，*PREDICT* という次に入力される単語を予測するアクションを追加した手法を提案している．これらの手法は一定の翻訳精度を保ったまま遅延を削減することに成功しているが，翻訳器が文の部分的な情報から翻訳することに対して最適化されていないことや遅延の大きさを調整できないという問題が残されている．

Ma et al. [6] では “Wait-k” モデルと呼ばれる非常にシンプルな手法が提案されている．このモデルは原言語側の文の入力に対して常に k トークン遅れた状態でリアルタイムに翻訳文の生成を行う．この方法により翻訳を行う機構と動詞などの予測を行う機構の両方を統合して扱うことができ，それを End-to-End で学習することができる．この手法は非常にシンプルにもかかわらず英語からドイツ語や中国語から英語の翻訳タスクにおいて高い精度を達成している．また，k を変化させることで遅延の大きさを調整することができるという利点もある．

統計的機械翻訳に対して NMT の翻訳精度は向上したが，語順が大きく異なるため難しいとされている英語から日本語への同時通訳タスクに対して NMT を適用する手法についてはほとんど検討されていない．そこで本研究では，“Wait-k” モデルを英語から日本語への同時通訳タスクに対して適用し，その翻訳結果の精度や問題点について検討する．

2 Attention 機構付き Encoder-Decoder モデルによる NMT

はじめに，背景知識として Attention 機構付き Encoder-Decoder モデル [2] について説明する．

入力文（入力系列）X および出力文（出力系列）Y を以下のように定義する．

$$X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I\},$$
$$Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_J\}.$$

ここで，$\mathbf{x}_i \in \mathbb{R}^{S\times 1}$ は i 番目の入力単語を表す one-hot ベクトル，I は入力文の長さ，$\mathbf{y}_j \in \mathbb{R}^{T\times 1}$ は j 番目の出力単語を表す one-hot ベクトル，J は出力文の長さ

を表す．

この時，原言語から目的言語への翻訳という問題は，以下の文に対する条件付き確率を最大化する最適な翻訳文 $\hat{Y}$ を見つけてくることによって解くことができる．

$$\hat{Y} = \arg\max_{Y} \ p_\theta(Y|X) \tag{1}$$

この文に対する条件付き確率は，原言語文 X と時刻 j までに生成した翻訳文 $\mathbf{y}_{<j}$ から単語に対する条件付き確率の積の形として以下のように分解される．

$$p_\theta(Y|X) = \prod_{j=1}^{J} p_\theta(\mathbf{y}_j|\mathbf{y}_{<j}, X). \tag{2}$$

ここで θ はモデルのパラメータを表す．

モデルは Encoder（§2.1）と Attention + Decoder（§2.2）の 2 つの機構から構成され，そのどちらも RNN（Recurrent Neural Network）を用いて構成される．

2.1 Encoder

Encoder は入力文 X を入力として受け取り，RNN を通じて順方向の隠れ状態ベクトル $\overrightarrow{\mathbf{h}_i}(1 \leq i \leq I)$ を返す．

$$\overrightarrow{\mathbf{h}_i} = RNN(\overrightarrow{\mathbf{h}_{i-1}}, \mathbf{x}_i). \tag{3}$$

同様に，逆順に並べた入力文を入力することで逆方向の隠れ状態ベクトル $\overleftarrow{\mathbf{h}_i}(1 \leq i \leq I)$ が得られる．これらの 2 つの方向の隠れ状態ベクトルを結合することで以下のように入力文の隠れ状態ベクトルを得る．これにより全てのタイムステップにおいて前後の文脈を考慮した隠れ状態ベクトルを得ることができる．

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]. \tag{4}$$

2.2 Attention + Decoder

Attention + Decoder では Encoder で計算された入力文の隠れ状態ベクトルから翻訳文の単語を 1 つずつ生成する．Decoder の RNN は初期隠れ状態ベクトル $\mathbf{h}_I$ から始まり，隠れ状態と過去の出力系列から再帰的に単語を生成する．出力単語 $\mathbf{y}_i$ の条件付き確率は以下のように定義される．

$$p_\theta(\mathbf{y}_j|\mathbf{y}_{<j}, X) = softmax(\mathbf{W}_s\tilde{\mathbf{d}_j}), \tag{5}$$

$$\tilde{\mathbf{d}_j} = tanh(\mathbf{W}_c[\mathbf{c}_j; \mathbf{d}_j]), \tag{6}$$

$$\mathbf{d}_j = RNN(\mathbf{d}_j, \mathbf{y}_{j-1}). \tag{7}$$

ここで，$\mathbf{W}_c, \mathbf{W}_p$ は学習されるパラメータである．また，$\mathbf{c}_j$ は文脈ベクトルである．この $\mathbf{c}_j$ を求めるために Attention と呼ばれる機構を用いる．Attention 機構では，入力文の隠れ状態ベクトル $\mathbf{h}_i$ をその各ベクトルに対応する時間ステップ j における重み α_{ij} を計算し，その重みと隠れ状態ベクトルの重み付き平均を取ることで $\mathbf{c}_j$ が以下のように求められる．

$$\mathbf{c}_j = \sum_{i=1}^{I} \alpha_{ij}\mathbf{h}_i, \tag{8}$$

$$\alpha_{ij} = \frac{exp(\mathbf{d}_j^T\mathbf{h}_i)}{\sum_{i'=1}^{I} exp(\mathbf{d}_j^T\mathbf{h}_{i'})} \tag{9}$$

3 “Wait-k” モデルによる同時通訳

次に，同時通訳に用いる “Wait-k” モデルについて説明する．

従来の機械翻訳システムが文全体が入力されることを仮定した学習を行っているのに対して，同時通訳システムでは文の先頭のみが入力された状態から訳出を行う必要がある．そのため，従来の機械翻訳モデルでは文に対する条件付き確率として式 (2) のように定義されるのに対して，“Wait-k” モデルでは以下のように定義される．

$$p_\theta(Y|X) = \prod_{j=1}^{J} p_\theta(\mathbf{y}_j|\mathbf{y}_{<j}, \mathbf{x}_{<g(j)}). \tag{10}$$

ここで，$\mathbf{y}_{<j}$ は時刻 j までに生成した翻訳文，$\mathbf{x}_{<g(j)}$ は時刻 $g(j)$ までに入力された原言語文を表す．また，$g(j)$ は Decoder が時刻ステップ j までトークンを生成する時に Encoder によって処理されるトークン数を表し，以下のように定義する．

$$g(j) = \begin{cases} k + j - 1 & (j < I - k) \\ I & (otherwise) \end{cases} \tag{11}$$

この時，k は翻訳文の生成が原言語文の入力よりも常に k トークン遅延していること表すハイパーパラメータである．言い換えると，この式は Decoder がトークンを生成する際に観測できる原言語側のトークン数を表す．最初のトークンを生成する際には k トークン分の情報がエンコードされ観測することができ，2 ステップ目以降については各タイムステップにおいて目的言語側の観測できるトークンの数が 1 つづつ増えていく．そして，$I - k$ ステップでは全ての原言語側のトークンが入力された状態となるため，それ以降（$j \geq I - k$）のステップでは観測できるトークンの数の増加は止まり，原言語文のトークン数で一定となる．

また，同時通訳では文末が確定しない状況で処理しなければならないため，式 (4) のような逆方向のベクトルを参照できず，順方向のベクトルのみを利用することとなる．そのため，従来の機械翻訳システムでは式（8）と式（9）で計算されていた Attention 機構による文脈ベクトル $\mathbf{c}_j$ も，以下のように計算されるように

表 1: 実験に用いたコーパス．

Corpus	Number of Sentence		
	Train	Valid.	Test
ASPEC	964k	1790	1812

なる．

$$\mathbf{c}_j = \sum_{t=1}^{g(j)} \alpha_{ij} \overrightarrow{\mathbf{h}_i}, \tag{12}$$

$$\alpha_{ij} = \frac{exp(\mathbf{d}_j^T \overrightarrow{\mathbf{h}_i})}{\sum_{t'=1}^{g(j)} exp(\mathbf{d}_j^T \overrightarrow{\mathbf{h}_{t'}})}. \tag{13}$$

4 実験

“Wait-k” モデルによる日本語から英語への翻訳タスクでの実験を行い，その翻訳結果の精度や問題点について検討した．

4.1 実験設定

モデルの実装には primitiv[*1] を用いた．また，Encoder と Decoder の RNN はそれぞれ 2 層の LSTM とし，input feeding を行った．単語埋め込みベクトルや隠れ状態ベクトルの次元数はどちらも 512，ミニバッチのサイズは 64 とした．語彙は原言語と目的言語で共有し，そのサイズは 16000 とした．最適化アルゴリズムには Adam[7] を使用し，gradient clipping は 5，weight decay は 10^{-6} に設定して学習を行った．ドロップアウトの確率 p は 0.3 とし，learning rate は各 epoch ごとに validation loss が低下しない場合にのみ $1/\sqrt{2}$ を掛けることを減衰を行った．また，テストは validation loss を記録したモデルによって行った．評価尺度には，機械翻訳の自動評価尺度として一般的に使用されている BLEU[8] を使用した．

4.2 データセット

日本語と英語へのタスクでの実験を行うにあたって，パラレルコーパスとして ASPEC[9] を使用した．ASPEC は中規模のコーパスで，比較的長文で専門用語が多いなど複雑な文章から構成されている．表 1にコーパスの詳細を示す．

英語および日本語の入力単位はサブワード [10, 11] とし，Sentencepiece[*2] を用いてトークナイズを行った．また，文の長さが 60 トークンを超えるもの，どちらか一方の文の長さがもう一方の長さの 9 倍以上になっているものに関しては，その文のペアを学習データから削除するフィルタリングを行った．

[*1] https://github.com/primitiv/primitiv

[*2] https://github.com/google/sentencepiece

表 2: BLEU による評価結果．Attention Encoder-Decoder の遅延は評価用データセットでの原言語文の平均入力トークン数を表す．

モデル	遅延トークン数 k	BLEU
Attention EncDec[2]	(29.86)	35.70
“Wait-k” モデル [6]	3	20.21
	5	23.01

4.3 実験結果

“Wait-k” モデルの遅延トークン数 k を 3 および 5 に設定して実験を行った．

BLEU による評価結果を表 2に示す．Attention EncDec の BLEU スコアと比べると “Wait-k” モデルのスコアは少し低い結果となった．しかし，attention encdec の平均遅延トークン数が 29.86 なのに対して “Wait-k” モデルの遅延が 3 トークンから 5 トークンと非常に小さいことを考えると，このモデルは高い精度が得られていると考えられる．

4.4 考察

以上の実験結果より，“Wait-k” モデルは日英の同時通訳タスクにおいても非常に小さな遅延において一定の翻訳精度を実現できることがわかった．

表 3に遅延が k=5 のときの翻訳結果の例を示す．Example (1) では，原言語文の by this の部分や using 以降の部分が，参照訳や従来手法による訳では語順が入れ替わって早い段階で訳出されていることがわかる．それに対して “Wait-k” モデルでは遅延を減らすため，原言語文との語順が大きく変わらないようにそれらの部分の訳出を遅らせることができていることがわかる．

一方で，Example (2) は翻訳に失敗している例である．この例では {Details, of, does, rate, of, ”} の 6 単語が入力された状態で最初の単語を出力するため，そのタイミングまでに入力されていた「線量率の詳細」が主語として出力されている．そして，そのあとに続くべきである「ふげん発電所」という情報が抜け落ちていて，そのかわりに下線部で示す部分が生成されている．他の翻訳結果を見ても，このように文脈からこの後に入力される単語を予測することが難しい名詞句などの翻訳において，そのフレーズが非常に長い場合にうまく翻訳できていない例が見られる．これは，1 つのフレーズのサイズが k よりも大きい場合にフレーズの区間や構文情報を得ることができなくなるため，翻訳失敗しているケースが発生しているのでは無いかと考えられる．また，このモデルの生成方法を考えると，このような場合には後ろから情報を追加するような文章を生成できる必要がある．しかし，一般に使用されるパラレルコーパスにはそのような文章が含まれていない．

表 3: “Wait-k” モデル (k=5) での翻訳例.

Example (1)	
原言語文:	The germ line was peculiarly manifested by this, and the analysis was carried out using fluorescence correlation spectroscopy and laser scanning type fluorescence microscope .
参照訳:	これによって生殖細胞系列を特異的に発現させ, 蛍光相関分光法及び, レーザ走査型蛍光顕微鏡を用いて解析を行った。
従来手法:	これによって生殖系列特異的に発現し, 蛍光相関分光法とレーザー走査型蛍光顕微鏡を用いて解析を行った。
“Wait-k”:	生殖系列はこの細胞で特異的に発現し, その解析を蛍光相関分光法とレーザ走査型蛍光顕微鏡を用いて行った。

Example (2)	
原言語文:	Details of does rate of ”Fugen Power Plant” can be calculated by using DERS software .
参照訳:	DERS ソフトウェアを用いて「ふげん発電所」の線量率を詳細に計算できる。
従来手法:	「ふげん発電所」の線量率の詳細は DERS ソフトウェアを用いて計算できる。
“Wait-k”:	線量率の詳細は，平成 10 年度から実施された「燃料計画」のDERS ソフトウェアを用いて計算できる。

これらの問題に対する解決方法としては，原言語文との語順が大きく変わらない翻訳文を学習データとして学習を行うことや事前にチャンクを推定し，チャンク単位で入力を行うことなどが考えられる．

5 まとめ

本論文では，“Wait-k” モデルを英語から日本語への同時通訳タスクに対して適用し，その翻訳結果の精度や問題点について検討を行った．その結果，実験において “Wait-k” モデルは 3 トークンや 5 トークンという非常に小さい遅延において一定の翻訳精度を実現することできていることがわかった．

今後の課題としては，原言語文との語順が大きく変わらないような翻訳文の生成やそのようなデータを使用した翻訳器の学習，話し言葉への対応などが考えられる．

謝辞

本研究の一部は JSPS 科研費 JP17H06101 の助成を受けたものである．

参考文献

[1] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In InterSpeech, pages 3487–3491, Lyon, France, August 2013.

[2] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Proceedings of EMNLP, pages 1412–1421, September 2015.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[4] Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. Learning to translate in real-time with neural machine translation. In Proceedings of EACL, volume 1, pages 1053–1062, 2017.

[5] Ashkan Alinejad, Maryam Siahbani, and Anoop Sarkar. Prediction improves simultaneous neural machine translation. In Proceedings of EMNLP, pages 3022–3027, 2018.

[6] Mingbo Ma, Liang Huang, Hao Xiong, Kaibo Liu, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, and Haifeng Wang. Stacl: Simultaneous translation with integrated anticipation and controllable latency. arXiv preprint arXiv:1810.08398, 2018.

[7] Diederik P. Kingma and Jimmy Lei Ba. Adam: a method for stochastic optimization. In Proceedings of ICLR2016, 2015.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of ACL, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002.

[9] 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 2016, pages 2204–2208, Portoro, Slovenia, may 2016.

[10] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of ACL, pages 1715–1725, Berlin, Germany, August 2016.

[11] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Proceedings of ACL, pages 66–75, 2018.