Average Token Delay: 同時翻訳の遅延評価尺度

加納保昌　須藤克仁　中村哲
奈良先端科学技術大学院大学
{kano.yasumasa.kw4,sudoh,s-nakamura}@is.naist.jp

概要

同時翻訳は、話し手が話し終わる前に翻訳を開始するタスクである。大きな遅延を許容すれば、翻訳の質を向上させやすいが、会話など即時性が求められる場合には、翻訳の質をできるだけ下げずに遅延を抑える必要がある。この遅延を測る既存の指標は、翻訳の出力開始時に着目し、出力の終了時を十分に考慮していなかった。しかし、ある部分の翻訳の出力終了が遅れると、次の部分の翻訳の出力開始を遅らせることにもつながり、即時性が失われる。そこで、本稿では、翻訳の終了時に着目した新しい遅延評価尺度である Average Token Delay (ATD) を提案する。同時機械翻訳の実験を行い、既存の遅延評価尺度と比較して、より様々な訳出タイミングの決め方の評価に適していることを確かめた。

1　はじめに

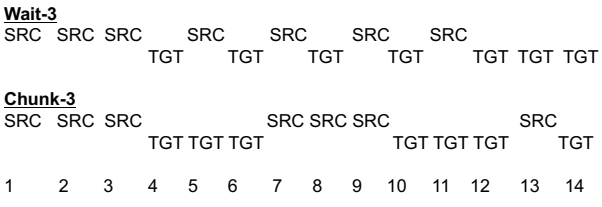

図 1　step ごとの wait-3 と chunk-3

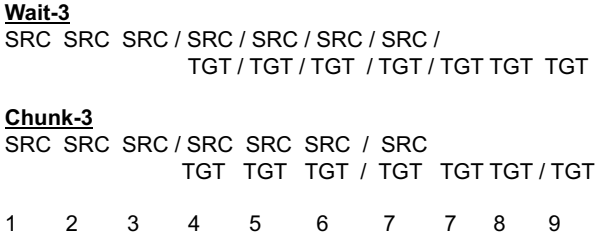

図 2　time-synchronous な wait-3 と chunk-3

同時翻訳とは、話し手が話し終わる前に翻訳を始めるタスクである。図 1 は、二種類の基本的な同時機械翻訳戦略による入出力過程をステップごとに示したものである。SRC と TGT はそれぞれ、入出力の 1 トークンを示す。wait-k [1] は、最初に k トークン待って、その後は　1 トークン入力を待ち、1 トークン　翻訳を出力するというのを繰り返す。chunk-k は、k トークン待って k トークン出力するというのを繰り返す。近年の同時機械翻訳モデルの翻訳の遅延評価には Average Lagging (AL) [1] が用いられてきた。これは、各翻訳単語の出力に必要となる入力単語の数をもとに計算される。図 2 は、入出力を同時に、かつ 1 トークンの入出力を同じ時間で実行できると仮定した wait-k と chunk-k の訳出タイミングを示している。ここでは簡略化のために計算時間を省略している。この図からわかる通り、wait-k と chunk-k は、この仮定の下では入出力のタイミングが完全に一致する。よって、これらの遅延は同一になるはずである。しかし、実際にこれらの AL を実際に計算すると、wait-k の AL は、$\frac{15}{5} = 3$, chunk-k の AL は $\frac{13}{7} \approx 1.857$ で、chunk-k の方が大幅に遅延が小さいと評価される。このように、チャンク翻訳が長くなるほど、AL は遅延を小さく評価する傾向がある。しかし、実際には、前のチャンクの翻訳が出力し終わらないと次の翻訳が出力できないため、チャンク翻訳の長さは遅延につながる。そこで、本稿では、出力翻訳の終わるタイミングに着目した遅延尺度である Average Token Delay (ATD)[1)] を提案する。同時機械翻訳の実験を行い、複数のモデルを AL と ATD で比較し、ATD は wait-k のみでなく、チャンクベースのモデルの遅延もより適切に測れることが分かった。

2　関連研究

これまで AL 以外にも、いくつかの遅延尺度が提案されてきた。Consecutive Wait (CW) [2] は文の局所的な遅延を評価する。Average Proportion (AP) [3] は文全体の遅延を評価するが、入出力のタイミング決定戦略が同じでも、文の長さによって、遅延の値が

1)　https://github.com/facebookresearch/SimulEval に実装予定

大きくことなってしまうという問題があった。AL は、理想的なタイミングとのズレをもとに計算することによって、その問題を改善した。AL は以下の式で定義され、

$$AL_g(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\tau_g(|\boldsymbol{x}|)} \sum_{\tau=1}^{\tau_g(|\boldsymbol{x}|)} \left(g(\tau) - \frac{\tau - 1}{r} \right). \quad (1)$$

$$\tau_g(|\boldsymbol{x}|) = \min\{\tau \mid g(\tau) = |\boldsymbol{x}|\} \quad (2)$$

$g(\tau)$ は単調増加関数で、τ 番目の出力単語を出力するために読んだ入力単語の数を表す。$\tau_g(|\boldsymbol{x}|)$ は、最後の入力単語を読み終わった直後に出力した翻訳単語のインデックスを表す。r は $|\boldsymbol{y}|/|\boldsymbol{x}|$ で定義される。

3 提案尺度

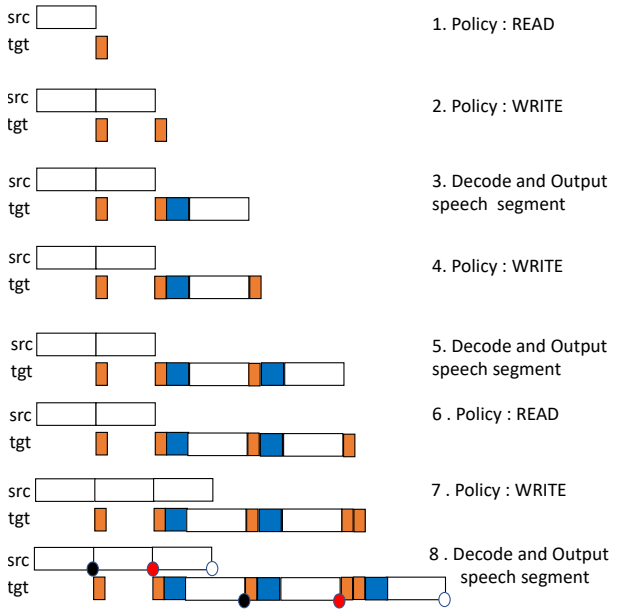

図 3 Step ごとの speech-to-speech 同時機械翻訳の例

本稿では、Average Token Delay という新しい指標を提案する。まずは、speech-to-speech の翻訳において説明し、それを speech-to-text、text-to-text への同時翻訳へと一般化する。

3.1 speech-to-speech の同時翻訳

図 3 は音声同時翻訳の音声や処理の継続長を考慮した処理時間を模式的に示したものである。白い部分は、固定長の音声セグメントを示し、青い部分は入力を待つか出力するかの決定する時間を示し、オレンジの部分は、翻訳デコード時間を示す。ステップ 1 から翻訳が始まり、ステップ 8 で翻訳を終え、ステップ 8 における同じ色の点 (黒、赤、白) の時間差の平均として ATD は計算される。

表 1 (4) 式の例

$L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}) \leq 0$						
x_1	x_2	x_3 /	x_4	$\underline{\boldsymbol{x_5}}$	x_6 /	
			y_1	y_2	y_3 /	y_4 $\underline{\boldsymbol{y_5}}$
$L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}) > 0$						
x_1	x_2	x_3 /	$\underline{\boldsymbol{x_4}}$	x_5	x_6 /	
			y_1	y_2	y_3	y_4 / $\underline{\boldsymbol{y_5}}$

表 2 (5) 式の例

$S \leq L_{acc}(\boldsymbol{x}^{c(t)})$					
x_1	x_2	$\underline{\boldsymbol{x_3}}$ /			
		y_1	y_2	$\underline{\boldsymbol{y_3}}$	
$S > L_{acc}(\boldsymbol{x}^{c(t)})$					
x_1	x_2	$\underline{\boldsymbol{x_3}}$ /			
		y_1	y_2	y_3	$\underline{\boldsymbol{y_4}}$

入力文 $\boldsymbol{x}$ と出力文 $\boldsymbol{y}$ がそれぞれ、$\boldsymbol{x} = \boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^C$ と $\boldsymbol{y} = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^C$ というチャンクに区切られるとする。それぞれのチャンクは、さらにサブセグメントに分かれる。テキストの場合は単語または文字がサブセグメントとなる。音声の場合は、0.3 秒で 1 単語の発話があると仮定し、チャンクを 0.3 秒の長さのサブセグメントに分割し、余りも 1 つのサブセグメントとする。その結果、入力文と出力文はそれぞれ $\boldsymbol{x} = x_1, ..., x_{|\boldsymbol{x}|}$、$\boldsymbol{y} = y_1, ..., y_{|\boldsymbol{y}|}$ というサブセグメントの系列として表せる。ATD は以下の式で定義される。

$$\mathrm{ATD}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{|\boldsymbol{y}|} \sum_{t=1}^{|\boldsymbol{y}|} \left(T(y_t) - T(x_{s(t)}) \right) \quad (3)$$

ここで、

$$S = t - max(L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}), 0) \quad (4)$$

$$s(t) = \begin{cases} S & S \leq L_{acc}(\boldsymbol{x}^{c(t)}) \\ L_{acc}(\boldsymbol{x}^{c(t)}) & otherwise \end{cases} \quad (5)$$

とする。$L_{acc}(\boldsymbol{x}^c)$ は c 番目のチャンクまでの累積長であり、$\sum_{j=1}^{c} |\boldsymbol{x}^j|$ と表せる。$c(t)$ は y_t の属するチャンク番号 c を表し、$L_{acc}(\boldsymbol{x}^0) = 0$、$L_{acc}(\boldsymbol{y}^0) = 0$ とする。

$T(\cdot)$ はそれぞれのサブセグメントの終わるタイミングを示し、図 3 で色のついた点である。ATD は、出力サブセグメント y_t と対応する入力サブセグメント $y_{s(t)}$ の時間差の平均として計算される。入出力のサブセグメントの対応は、特に語順の大きく異なる言語ついにおいては意味的に必ずしも正しくな

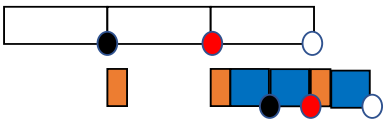

図 4 speech-to-text 同時翻訳の遅延測定概略図

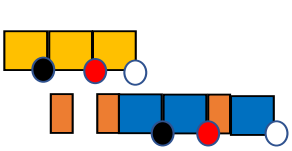

図 5 text-to-text 同時翻訳の遅延測定概略図

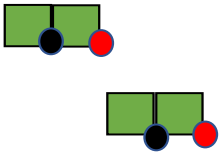

図 6 non-computation aware text-to-text 同時翻訳の遅延測定概略図

いが、遅延の計算においては AL と同様に、このように単純化している。

表 1 は、(4) 式の max の中の例を表す。$L_{acc}(\boldsymbol{y}^{c(5)-1}) - L_{acc}(\boldsymbol{x}^{c(5)-1}) = 0 \leq 0$ の時は、y_5 は x_5 に対応する。$L_{acc}(\boldsymbol{y}^{c(5)-1}) - L_{acc}(\boldsymbol{x}^{c(5)-1}) = 1 > 0$ の時は、1 チャンク前までの部分翻訳が 1 チャンク前までの部分入力より 1 トークン長いため、$S = 5 - 1 = 4$ により、y_5 は x_4 に対応する。このように、ATD では前のチャンク翻訳の終了が遅くなると遅延が増える。

表 2 は、(5) 式の例を表す。$S = 3 \leq L_{acc}(\boldsymbol{x}^{c(3)})$ の時は、y_3 は x_3 に対応する。$S = 4 > L_{acc}(\boldsymbol{x}^{c(4)})$ の時は、x_4 が存在せず、最後の入力トークン x_3 に対応する翻訳が長めに出力していると捉える。そこで、$L_{acc}(\boldsymbol{x}^{c(4)}) = 3$ により、y_4 は x_3 に対応する。

3.2 {speech,text}-to-text の同時翻訳

図 4 は speech-to-text への同時翻訳を表している。字幕のように、テキストは同時に複数の単語を出力できるため、テキストのサブセグメント (単語) の長さは無視している。図 5 は text-to-text 同時翻訳を示している。この場合、入力のテキストは基本的に streaming ASR(自動音声認識) の結果として受け取るため、黄色の入力サブセグメントの長さは、ASR の処理時間となる。

3.3 Non-computation-aware ATD

コンピュータの性能や実装効率に依存せずに、同時翻訳モデルの遅延を図ることがある。そのような場合には、図 3, 4, 5, のオレンジ、青、黄色の部分を除去し、音声セグメントの長さのみを残して、ATD の遅延を計算する。しかし、このままでは、text-to-text の翻訳では、遅延が常に 0 になってしまう。そこで、図 6 に示される通り、CW や AP のように入出力の単語がそれぞれ 1 step の時間を使うとして、ATD を計算する。また、ATD は、図 2 で示されるように、モデルが入出力を並列に行うことができると仮定して計算される。

4 シミュレーション

ATD による遅延計測の特性を示すためのシミュレーションの結果を以下に示す。

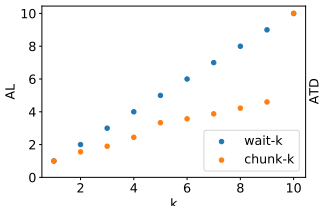

図 7 AL による遅延測定

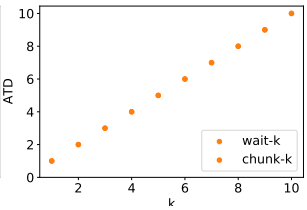

図 8 ATD による遅延測定

4.1 wait-k と chunk-k の比較

長さ 10 トークンの入力文を wait-k と chunk-k で、長さ 10 トークンの文へ翻訳する際に、k を 1 から 10 まで変化させて比較した。図 7,8 がその結果を示す。図 7 は 1 節で述べられたように、chunk-k と wait-k の AL に大きな差が出てしまうことを示している。その一方で、ATD は図 8 で示されている通り、chunk-k と wait-k が一致している。また、AL では、wait-9 と wait-10 は遅延の大きさが近いにも関わらず、ギャップが生まれてしまっている。付録にさらに大きな図を示す。このように、ATD は、wait-k のような 1 トークン読んで 1 トークン出力するというのを繰り返すモデルでも、複数トークンを一度に出力するチャンクベースのモデルでも、より適切な結果が得られると考えられる。

4.2 翻訳例

図 9 はチャンクベースの 3 つの異なるモデルによる同時翻訳例である。モデル 1 は入力文全体を待ってから、翻訳を始めているため、最も遅延が大きく、翻訳の質は高い。モデル 2 は、「I」の直後に区切って翻訳を始めているため、モデル 1 より遅延が小さいが、その分翻訳の質は落ちる。モデル 3 はモデル 2 に似ているが、チャンク翻訳の出力長が長く、その分翻訳の質が落ちている。

AL は、モデル 3 とモデル 2 の間に急激な差があり、モデル 3 が最も遅延が少ないと評価している

Model 1
ATD:5.4, AL:5.0, Quality: **1st**
I bought a pen . /
私 は ペン を 買っ た 。
1 2 3 4 5 6 7 8 9 10 11 12

Model 2
ATD:**3.4**, AL:1.1, Quality: 2nd
I / bought a pen . /
私 。 / ペン を 買っ た 。
1 2 3 4 5 6 7 8 9 10

Model 3
ATD:4.1, AL:**0.8**, Quality: 3rd
I / bought a pen . /
私 で ござい ます 。/ペン を 買っ た 。
1 2 3 4 5 6 7 8 9 10 11

図 9 翻訳例

が、これはこれまで述べてきた通り、直感に反している。その一方で、ATD は出力長の遅延を考慮しているため、翻訳が終わるのが遅いモデル 3 よりモデル 2 の方が遅延が小さいと評価することができている。

5 実験

遅延尺度の効果を確かめるため、text-to-text の英独同時機械翻訳の実験を行い、AL と ATD を比較した。

5.1 データ

IWSLT evaluation campaign のデータを用いた。WMT 2014 training set (450 万文) をトレーニングデータとし、IWSLT 2017 training set (20.6 万文) をファインチューニングのデータとして用いた。dev2010, tst2010, tst2011 and tst2012 (合計 5,589 文) を開発データとし、tst2015 (1,080 文) で評価した。

加納ら [4] に従った実験設定で、wait-k [1], Meaningful Unit (MU) [5], Incremental Constitute Label Prediction (ICLP) [6], そして Prefix Alignment (PA) [4] のモデルを比較した。翻訳の質と遅延の評価には、SimulEval を用いた。

5.2 結果

図 10,11 はその実験結果を示す。図 10 の AL に比べると、図 11 の ATD は各モデルの遅延の違いをより明確に示している。MU と ICLP は ATD において、遅延と精度のトレードオフが低下している。この原因は、図 12 の AL が小さい範囲で length ratio が 1.0 を超えているところにも現れている。AL が小さい部分では、翻訳が参照文に比べて長めに出力されてしまい、図 9 のモデル 3 と同様の現象が起こっている。MU は、ATD が大きくなっても、必ずしも BLEU [7] が向上していない。これは、本来より長く翻訳が出力されたことによって ATD が大きくなると同時に、翻訳の質も下がってしまったからである。

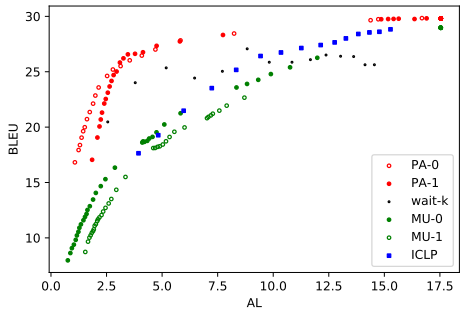

図 10 AL による遅延測定

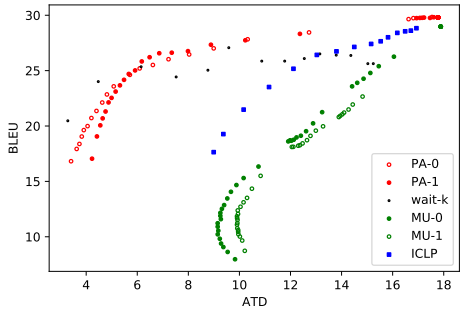

図 11 ATD による遅延測定

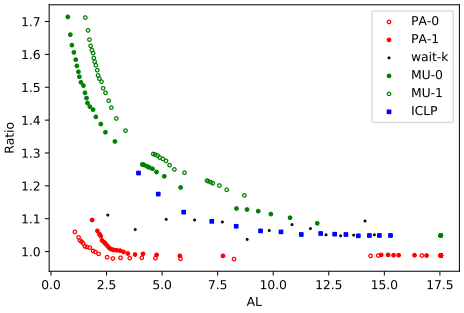

図 12 Length ratio と AL

6 おわりに

同時機械翻訳における新しい遅延尺度を提案した。出力翻訳長を考慮した ATD は、AL ではうまく測れないチャンクベースのモデルにも対応した。今後は、タイミング情報のみならず、入出力単語の意味の対応関係も考慮していくことが重要である。

謝辞

本研究の一部は科研費 21H05054 の助成を受けたものである。

参考文献

[1] Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3025–3036, Florence, Italy, July 2019. Association for Computational Linguistics.

[2] Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. Learning to translate in real-time with neural machine translation. In **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers**, pp. 1053–1062, Valencia, Spain, April 2017. Association for Computational Linguistics.

[3] Kyunghyun Cho and Masha Esipova. Can neural machine translation do simultaneous translation? **arXiv preprint arXiv:1606.02012.**, 2016.

[4] Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. Simultaneous neural machine translation with prefix alignment. In **Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)**, pp. 22–31, Dublin, Ireland (in-person and online), May 2022. Association for Computational Linguistics.

[5] Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. Learning adaptive segmentation policy for simultaneous translation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 2280–2289, Online, November 2020. Association for Computational Linguistics.

[6] Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. Simultaneous neural machine translation with constituent label prediction. In **Proceedings of the Sixth Conference on Machine Translation**, pp. 1124–1134, Online, November 2021. Association for Computational Linguistics.

[7] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

A　付録

A.1　シミュレーション

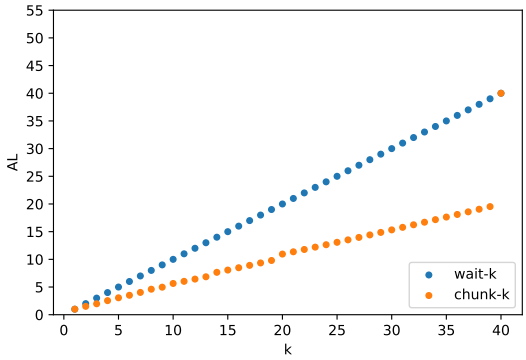

図 13　AL による遅延測定

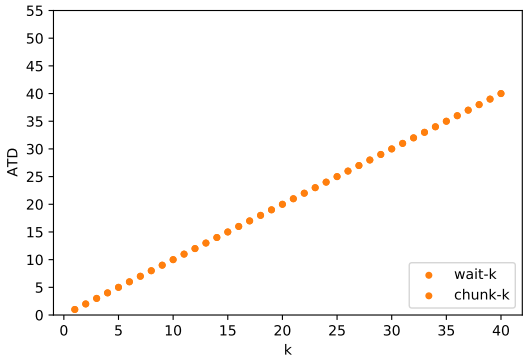

図 14　ATD による遅延測定

長さ 40 トークンの入力文を wait-k と chunk-k で、長さ 40 トークンの文へ翻訳する際に、k を 1 から 40 まで変化させて比較した。本文の 10 トークンの場合に比べ、より chunk-k と wait-k のギャップが大きくなっている。