漸進的な音声分割を用いたストリーミング同時音声翻訳

福田 りょう　須藤 克仁　中村 哲
奈良先端科学技術大学院大学
{fukuda.ryo.fo3, sudoh, s-nakamura}@is.naist.jp

概要

同時音声翻訳モデルを実用化する際には、連続音声を逐次的に処理する必要があるが、近年の研究は音声が文等の短い単位に分割されたベンチマークデータを対象に行われており、連続音声の処理については十分に検討が行われていない。そこで本稿では漸進的な音声分割モデルを作成し、同時音声翻訳モデルと組み合わせて連続音声を処理するストリーミング同時音声翻訳システムを構築した。実験では、音声分割手法がトップラインの 94%以上の翻訳精度を維持できることを確認し、また音声分割における将来と過去の音声情報の重要性を検証した。

1　はじめに

同時音声翻訳は、発話の完了を待たずに翻訳処理を開始する音声機械翻訳であり、言語の壁を超えた即時性の高い情報伝達を実現する技術として実用化が期待されている。同時音声翻訳の従来方式では、音声認識、発話より短いチャンクへの自動分割、機械翻訳という 3 つの処理が必要であった [1, 2, 3]。ニューラル機械翻訳 [4] の発展に伴い、音声を直接翻訳する End-to-end 型の音声翻訳モデル [5] を用いてよりシンプルな構成で同時音声翻訳が実現できるようになった。Ma ら [6] は、テキストからテキストへの同時翻訳の手法 [7, 8] を同時音声翻訳に適用した。Liu ら [9] は訳出タイミングと翻訳精度を同時に最適化する手法を提案した。近年では、事前学習済みの大規模な音声モデルや多言語モデルを用いた同時音声翻訳が提案されている [10, 11]。

このような発展の一方で、同時音声翻訳における連続音声の処理方法については十分な検討がなされていない。音声は本来、長さに制限のない連続的なデータ (ストリーミングデータ) である。しかしほとんどの既存研究ではそのような入力が想定されておらず、実験では事前に発話単位に分割された数秒〜十数秒程度の音声セグメントが入力として用いられてきた。

即時性が求められない通常の音声機械翻訳 (オフライン音声翻訳) では、連続音声を適切に処理するための手法が研究されている。既存の音声翻訳モデルは単体で連続音声を処理することが難しいため、翻訳前に音声分割を実行する方法が一般的である。音声区間検出による分割 [12] は最も典型的な音声分割手法であるが、発話の過剰な分割により翻訳精度を低下させる問題が指摘されている [13]。この問題を緩和するため、音声区間で分割した音声セグメントを一定の長さになるまで連結する方法が提案されている [14, 15]。近年では、音声翻訳コーパスに含まれる発話単位のセグメント境界を予測する分類器を用いたモデルベースの手法が提案され、従来手法を大きく上回ることが報告されている [16, 17, 18]。

既存の音声分割手法の多くはオフライン音声翻訳のために設計されており、発話より長い音声を入力することが想定されている。一方で同時音声翻訳においては、翻訳処理と同様に、発話の完了を待たずに漸進的な音声分割処理を行う必要がある。そこで本稿では、オフライン音声翻訳のための音声分割手法の、同時音声翻訳への適用可能性を検討した。具体的には、モデルベースの音声分割手法 [18] を漸進的な処理に対応させるために、学習時に参照できる将来の情報に制約をかけて音声分割モデルの学習を行った。その後、作成した音声分割モデルと同時音声翻訳モデルを組み合わせてストリーミング同時音声翻訳システムを構築した。TED talks の音声翻訳コーパス MuST-C を用いた英独音声翻訳実験で、提案手法がベースラインを上回り、音声翻訳コーパスに含まれる文単位のセグメントの 94%以上の翻訳精度を維持できることを示した。また、音声分割モデルが参照できる将来の情報に制約をかけることで学習と推論の条件の不一致が緩和され、翻訳精度が向上することを確認した。最後に、音声分割における過去の情報の重要性を検証した。

2　漸進的な音声分割

2.1　モデルベースの音声分割

音声翻訳コーパスには通常、文にアライメントされた音声セグメントを含む。これらのセグメントの境界は多くの場合発話の区切りと一致するため、翻訳に適していると考えられる。SHAS [16] は、音声翻訳コーパスのセグメント境界を予測する分類器を用いてオフライン音声翻訳において高い翻訳精度を達成した。SHAS の分類器の構造は、事前学習済みの自己教師あり音声モデル wav2vec 2.0 [19] のエンコーダに 1 層の Transformer Encoder 層 [20] を繋げたニューラルネットワークモデルである。モデルは、フレーム単位の系列ラベリング問題として音声翻訳コーパスのセグメント境界の予測を学習する。学習時及び推論時には固定長 (20 秒) の音声が入力される。SHAS+FTPT [18] は、SHAS の拡張であり、分類器の wav2vec 2.0 のパラメータを追加学習することでより高い翻訳精度を達成した。また SHAS+FTPT は、音声をより短いセグメントに分割するため、処理時間の観点から漸進的処理に適している。

2.2　Attention masking

本研究では音声分割モデル SHAS+FTPT を同時音声翻訳に適用する。wav2vec 2.0 及び追加の Transformer Encoder 層では、self-attention により、最大 20 秒先の音声情報を参照できる。同時音声翻訳では、発話より短いチャンクを漸進的に処理する必要があるため、SHAS+FTPT の学習時のように長い将来の情報を参照できない。本研究では、モデル学習時に self-attention が参照できる将来の情報に制約をかけることで推論時とのギャップを低減することを検討した。Self-attention は下式で定義される scaled dot-product attention である。

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \qquad (1)$$

Attention の重み行列 $\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \in R^{T\times T}$ は、j 番目の Query Q_j が i 番目の Value V_i を参照する際の重み $\alpha_{i,j}$ $(i <= T, j <= T)$ の要素から構成されている。もし $\alpha_{i,j} = 0$ であれば、Q_j は V_i を参照しない。ここで、Attention の参照範囲を制御するために、重み行列にマスク行列 $M \in R^{T\times T}$ を掛け合わせることができる。もし $M(i, j) = 0$ であれば、Q_j は V_i を参照しない。

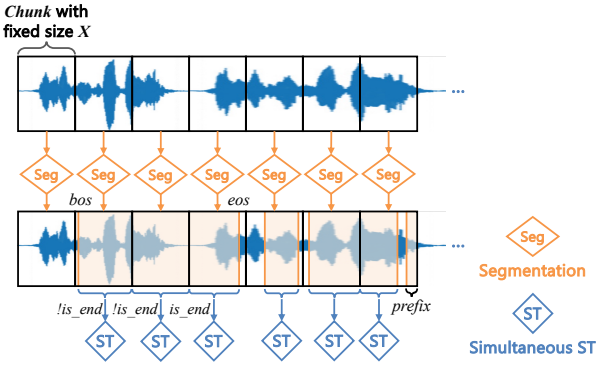


図 1: ストリーミング同時音声翻訳システムの概要

本研究では異なるマスク行列 M のパターンを用いて学習した 3 つの音声分割モデルを比較した。

- **Unmasked**: M を用いない。学習時に self-attention は最大 20 秒先の情報を参照できる。
- **Monotonic masking** [21, 22, 23]: i, j $(j < i)$ について $M(i, j) = 0$ となるような M を用いる。学習時に self-attention は過去の情報のみ参照できる。
- **Chunk-wise masking** [24, 25]: 入力フレームを固定のチャンクサイズに分割して現在と過去のチャンクだけを参照できるようにする。具体的には、

$$M(i, j) = \begin{cases} 0 & \text{if } c(i) \neq c(j) \text{ and } j < i, \\ 1 & \text{otherwise.} \end{cases} \qquad (2)$$

とする。$c(i)$ は i 番目のフレームが属するチャンクのインデックスである。Monotonic masking と比べ、チャンク単位で漸進的に処理を行う推論時と近い条件で学習を行う。

3　ストリーミング同時音声翻訳

図 1 に本稿で構築するストリーミング同時音声翻訳 (Streaming Simultaneous ST) システムの概要図を示す。システムは音声分割 (Segmentation) と同時音声翻訳 (Simultaneous ST) で構成されており、2 つのモデルが交互に処理を行うことで動作する。擬似コードを Algorithm 1 に示す。システムは receive_speech_stream() 関数を用いて音声データを連続的に受け取り *chunk* に追加する (行 8)。音声チャンク *chunk* の長さが固定長 X に達すると、音声分割 Segmentation(·) を実行する (行 10-11)。次に、音声分割結果として得られる音声セグメントの一部である *part_seg*、同一セグメント内の過去の音声情報 *context*、セグメント終了フラグ *is_end* を引数にと

Algorithm 1 Streaming Simultaneous ST

```
1:  Input: chunk size X (in seconds), threshold thr
2:  Initialize:
3:      chunk ← []                          ▷ pool to store speech data
4:      context ← []   ▷ context for segmentation and translation
5:      next_prefix ← []    ▷ pool to store next segment's prefix
6:      INPUT_SPS ← 16000                   ▷ sps of audio; 16kHz
7:  while True do
8:      append receive_speech_stream() to the chunk
9:      if length(chunk) ≥ X · INPUT_SPS then
10:         part_seg, prefix, is_end ← Segmentation(
11:                     chunk, cat(next_prefix+context), thr)
12:         if len(next_prefix) ≥ 0 then
13:             part_seg ← cat(next_prefix, part_seg)
14:         next_prefix ← prefix
15:         SimulST(part_seg, context, is_end)
16:         if is_end == False then
17:             context ← cat(context, part_seg)
18:         else
19:             context ← []
20:         chunk ← []
```

る、同時音声翻訳 SimulST(·) を実行する (行 15)。ここで、*is_end* == True の場合、SimulST(·) は文末まで生成を行った後、次のセグメントの処理に備えて状態をリセットする。その後、過去の情報 *context* を更新し（行 16-19)、*chunk* をリセットして（行 20）行 8 の処理に戻る。

Segmentation(·) では音声分割モデルの推論を実行する。音声分割モデルは、過去の情報と現在のチャンクを連結した系列を入力として受け取り、各フレームが音声セグメントに含まれる確率を出力する。次に、過去の情報に対応する予測を切り捨てた、現在のチャンク部分の確率の系列に閾値 thr ($0 \leq thr \leq 1$) で閾値処理を行うことでセグメントの境界（開始 *bos*, 終了 *eos*）を決定する。ここで $thr = 0.1$ は $thr = 0.9$ よりも消極的な分割を行うことを意味する。最後に *bos* と *eos* の間にある音声を *part_seg* として返す。この時、次のセグメントのプレフィックス *prefix* とセグメント終了フラグ *is_end* も同時に返す。Segmentation(·) の詳細を付録 A.1 に示す。

4 実験設定

ストリーミング同時音声翻訳システムを構築し、提案手法の有効性を検証するために異なる音声分割手法を比較した。

4.1 タスク

Ted talks の音声翻訳コーパス MuST-C v2 に含まれる、英語の講演音声とそれに対応付いた書き起こしテキスト、及びドイツ語の翻訳テキストを用いて英独音声翻訳の実験を行った。学習データ、開発データ及び評価データのセグメント数はそれぞれ 250,942、1,415、2,580 である。音声分割モデルの学習に学習・開発データを用いた。システムの評価のために評価データを用いた。

音声分割手法 (4.2) と同時音声翻訳モデル (4.3) を用いてストリーミング同時音声翻訳システムを構築した。システム構築のために SimulEval toolkit [26] [1] を用いた。

評価時は、自動分割した音声に対する翻訳結果を評価データのセグメントとトーク毎に対応付けを行った後、BLEU [27] を測定した。対応付けには編集距離に基づくテキスト整列アルゴリズム [28] を用いた。

4.2 音声分割

6 つの音声分割手法を比較した。

- *Topline*: 評価データのセグメント境界
- *Fixed-length*: 固定長で分割するベースライン
- Unmasked: マスクなしの音声分割モデル (2 章)
- Monotonic: Monotonic masking (2 章)
- Chunk-wise: Chunk-wise masking (2 章)

音声分割モデルにおける閾値処理の設定値 *thr* を 0.1 から 0.7 の間で探索し、最も翻訳精度が高かった 0.3 を選択した。Chunk-wise の学習時のチャンクの長さを 1 秒とした。*Fixed-length* の設定は固定長 10 秒から 20 秒の間で探索し、最も翻訳精度が高い 15 秒を選択した。

4.3 同時音声翻訳

事前学習済みの音声モデル HuBERT [29] と、多言語モデル mBART50 [30] に基づく同時音声翻訳モデル [11] を使用した。翻訳の出力タイミングを決定する方策として、過去の仮説と現在の仮説の間の最長共通プレフィックスを出力する Local agreement [31] を用いた。推論時のチャンクサイズを $X = \{400, 600, 800, 1000, 1200\}$ (ms) として遅延を調節した。

1) https://github.com/facebookresearch/SimulEval

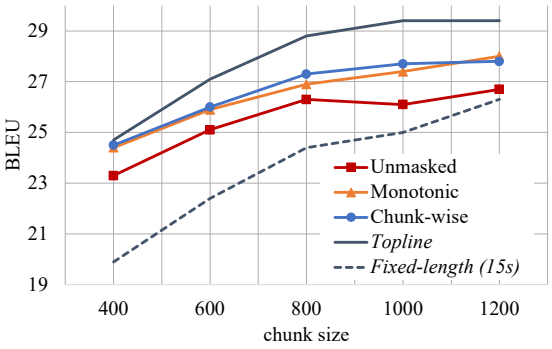


図 2: 異なるチャンクサイズにおける音声分割手法の比較。閾値 $thr = 0.3$。

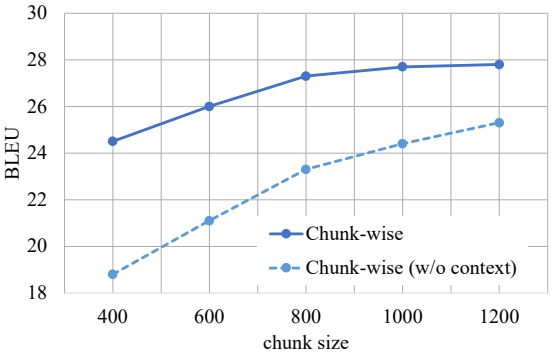


図 3: 過去の情報 *context* のアブレーション

5 実験結果

図 2 に各音声分割手法を用いたシステムの翻訳精度を示す。音声分割モデルを用いた 3 つの手法 (Unmasked、Monotonic、Chunk-wise) はいずれもベースライン *Fixed-length* を上回り、Chunk-wise は *Topline* の 94-99%の翻訳精度を維持した。各音声分割手法による翻訳精度の詳細を付録 A.2 に示す。Chunk-wise は Unmasked の BLEU を平均 1.2 ポイント上回った。マスク行列により参照できる将来の情報に制約をかけることで学習と推論の条件の不一致が緩和されたと考えられる。一方で、Monotonic と Chunk-wise の間に大きな差は見られなかった。ただし、詳細は省略するが、セグメント境界を決定するための閾値 thr を 0.3 より大きくした時の翻訳精度の低下は Monotonic でより顕著であり、$thr = 0.5$ 以上では Monotonic は Chunk-wise を大きく下回った。$thr = 0.1$ から 0.7 までの結果を付録 A.3 に示す。thr が大きいと分割がより積極的になるため、音声分割精度の差が強調されやすい。この結果は Chunk-wise が利用できるチャンクレベルの短い将来の情報が、分割精度向上に寄与することを示唆している。

音声分割モデルの比較から、将来の音声情報が分割精度に影響することが確認された。続いて過去の音声情報の重要性を検証するため、音声分割モデルの推論時に用いていた同一セグメント内の過去の情報 *context* を取り除いて結果を比較した。図 3 に結果を示す。いずれのチャンクサイズにおいても大きく翻訳精度が低下したことから、過去の情報も分割精度に大きく影響することが確認された。

6 関連研究

先行研究として、原理的に連続音声を処理できる同時音声翻訳モデルが提案されている [32] が、連続音声での実験は行われていない。Polák ら [33] は、音声分割を統合した同時音声翻訳モデルを提案した。これは我々の知る限り、End-to-end 型の音声翻訳モデルを用いて連続音声の同時音声翻訳実験を行った唯一の先行研究である。Polák らの手法は外部の音声分割手法を必要としない利点がある一方で、学習コストが高く、また任意の音声翻訳モデルを用いることができない欠点がある。

7 おわりに

本稿では、オフライン音声翻訳のための音声分割手法を同時音声翻訳に適用し、ストリーミング同時音声翻訳システムを構築した。実験では、提案した音声分割モデルがトップラインの翻訳精度を 94-99%維持することを示し、また音声分割の判断において将来と過去の情報が重要であることを明らかにした。

謝辞

本研究の一部は JSPS 科研費 JP21H05054、JP21H03500、JP23KJ1583 の助成を受けた。

参考文献

[1] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. Real-time incremental speech-to-speech translation of dialogs. In **Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 437–445, June 2012.

[2] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In **Proc. Interspeech 2013**, pp. 3487–3491, 2013.

[3] Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. Segmentation strategies for streaming speech translation. In **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language**

Technologies, pp. 230–238, June 2013.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. **Advances in neural information processing systems**, Vol. 27, , 2014.

[5] Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In **NIPS Workshop on end-to-end learning for speech and audio processing**, 2016.

[6] Xutai Ma, Juan Pino, and Philipp Koehn. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In **Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing**, pp. 582–587, December 2020.

[7] 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, July 2019.

[8] Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. Monotonic multihead attention. In **International Conference on Learning Representations**, 2020.

[9] Dan Liu, Mengge Du, Xiaoxi Li, Ya Li, and Enhong Chen. Cross attention augmented transducer networks for simultaneous translation. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 39–55, November 2021.

[10] Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In **Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)**, pp. 277–285, May 2022.

[11] Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Yuka Ko, Tomoya Yanagita, Kosuke Doi, Mana Makinae, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. NAIST simultaneous speech-to-speech translation system for IWSLT 2023. In **Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)**, pp. 330–340, July 2023.

[12] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. A statistical model-based voice activity detection. **IEEE signal processing letters**, Vol. 6, No. 1, pp. 1–3, 1999.

[13] David Wan, Chris Kedzie, Faisal Ladhak, Elsbeth Turcan, Petra Galuščáková, Elena Zotkina, Zheng Ping Jiang, Peter Bell, and Kathleen McKeown. Segmenting subtitles for correcting asr segmentation errors. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 2842–2854, 2021.

[14] Marco Gaido, Matteo Negri, Mauro Cettolo, and Marco Turchi. Beyond voice activity detection: Hybrid audio segmentation for direct speech translation. **CoRR**, Vol. abs/2104.11710, , 2021.

[15] Hirofumi Inaguma, Brian Yan, Siddharth Dalmia, Pengcheng Guo, Jiatong Shi, Kevin Duh, and Shinji Watanabe. ESPnet-ST IWSLT 2021 offline speech translation system. In **Proceedings of the 18th International Conference on Spoken Language Translation**, pp. 100–109, August 2021.

[16] Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. SHAS: Approaching optimal Segmentation for End-to-End Speech Translation. In **Proc. Interspeech 2022**, pp. 106–110, 2022.

[17] Ryo Fukuda, Katsuhito Sudoh, and Satoshi Nakamura. Speech Segmentation Optimization using Segmented Bilingual Speech Corpus for End-to-end Speech Translation. In **Proc. Interspeech 2022**, pp. 121–125, 2022.

[18] Ryo Fukuda, Katsuhito Sudoh, and Satoshi Nakamura. Improving speech translation accuracy and time efficiency with fine-tuned wav2vec 2.0-based speech segmentation. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, Vol. 32, pp. 906–916, 2024.

[19] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. **Advances in Neural Information Processing Systems**, Vol. 33, , 2020.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in neural information processing systems**, pp. 5998–6008, 2017.

[21] Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. In **International conference on machine learning**, pp. 2837–2846. PMLR, 2017.

[22] Chung-Cheng Chiu and Colin Raffel. Monotonic chunkwise attention. In **International Conference on Learning Representations**, 2018.

[23] Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. Monotonic multihead attention. In **International Conference on Learning Representations**, 2019.

[24] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li. Developing real-time streaming transformer transducer for speech recognition on large-scale dataset. In **ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 5904–5908, 2021.

[25] Jian Xue, Peidong Wang, Jinyu Li, Matt Post, and Yashesh Gaur. Large-Scale Streaming End-to-End Speech Translation with Neural Transducers. In **Proc. Interspeech 2022**, pp. 3263–3267, 2022.

[26] Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. SIMULEVAL: An evaluation toolkit for simultaneous translation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 144–150, October 2020.

[27] 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, July 2002.

[28] Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. Evaluating machine translation output with automatic sentence segmentation. In **Proceedings of the Second International Workshop on Spoken Language Translation**, 2005.

[29] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, Vol. 29, pp. 3451–3460, 2021.

[30] Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. **arXiv preprint arXiv:2008.00401**, 2020.

[31] Danni Liu, Gerasimos Spanakis, and Jan Niehues. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In **Proc. Interspeech 2020**, pp. 3620–3624, 2020.

[32] Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. Streaming simultaneous speech translation with augmented memory transformer. In **ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 7523–7527, 2021.

[33] Peter Polák and Ondřej Bojar. Long-form end-to-end speech translation via latent alignment segmentation. **arXiv preprint arXiv:2309.11384**, 2023.

Algorithm 2 Segmentation

1: **Input:** $chunk, context, thr$
2: **Output:** $part\_seg, prefix, is\_end$
3: $INPUT\_SPS \leftarrow 16000$ ▷ sps of audio; 16kHz
4: $OUTPUT\_SPS \leftarrow 49.95$ ▷ sps of seg()'s output
5: $IN\_OUT\_RATIO \leftarrow INPUT\_SPS \,/\, OUTPUT\_SPS$
6: $probs\_context, probs \leftarrow$ seg(concat($context, chunk$))
7: $segments \leftarrow$ thresholding($probs, thr$)
8: $segments \leftarrow$ expand_list($segments, IN\_OUT\_RATIO$)
9: $B_{bos} \leftarrow$ findBos($segments$)
10: $B_{eos} \leftarrow$ findEos($segments$)
11: **if** len($context$) $\neq 0$ **then**
12: **if** empty(B_{bos}) and empty(B_{eos}) **then** ▷ pattern (ii)
13: **return** $chunk$, [], False
14: **else if** B_{eos}[-1] > B_{bos}[-1] **then** ▷ pattern (iv), (viii)
15: **return** $chunk[:B_{eos}[-1]]$, [], True
16: **else** ▷ pattern (vi)
17: **return** $chunk[:B_{eos}[0]]$, $chunk[B_{bos}[0]:]$, True
18: **else**
19: **if** empty(B_{bos}) and empty(B_{eos}) **then** ▷ pattern (i)
20: **return** [], [], True
21: **else if** B_{eos}[-1] > B_{bos}[-1] **then** ▷ pattern (v)
22: **return** $chunk[B_{bos}[0]:B_{eos}[-1]]$, [], True
23: **else** ▷ pattern (iii), (vii)
24: **return** $chunk[B_{bos}[0]:]$, [], False

A 参考情報

A.1 Segmentation アルゴリズム

Algorithm 1 中の Segmentation(·) の擬似コードを Algorithm 2 に示す。処理手順は以下の通りである。

1. 過去の情報 *context* とチャンク *chunk* を連結して音声分割モデル seg(·) に入力し、各フレームがセグメントに含まれている確率を予測する (行 6)。
2. thresholding(·) で、*chunk* に対応する確率の系列 *probs* を閾値処理し $segments = [s_0, s_1, \ldots, s_{\lfloor X \cdot OUTPUT\_SPS \rfloor}](s_i \in \{0,1\})$ を得る (行 7)。
3. expand_list(·) で、*segments* の各要素を *IN_OUT_RATIO* 回繰り返した新しいリスト $segments = [s_0, s_1, \ldots, s_{\lfloor X \cdot INPUT\_SPS \rfloor}]$ を得る (行 8)。
4. findBos(·) と findEos(·) で、*bos* のインデックスのリスト B_{bos} と *eos* のインデックスのリスト B_{eos} をそれぞれ得る (行 9-10)。ここで
   - $B_{bos} = [i | x_{i-1} = 0 \text{ and } x_i = 1 \text{ for all } i \in \{1, 2, \ldots, \lfloor X \cdot INPUT\_SPS \rfloor\}]$
   - $B_{eos} = [i | x_{i-1} = 1 \text{ and } x_i = 0 \text{ for all } i \in \{1, 2, \ldots, \lfloor X \cdot INPUT\_SPS \rfloor\}]$

   である。
5. 図 4 はチャンクの予測を (i) から (viii) の 8 パターンに分類したものである。このうちパターン (iv) と (viii)、パターン (iii) と (vii) を同様に扱い、*context*, B_{bos}, B_{eos} の内容に応じて 6 通りの分岐処理を行う (行 11-24)。*prefix* はパターン (vi) で生じる次のセグメントのプレフィックスを管理する変数である。

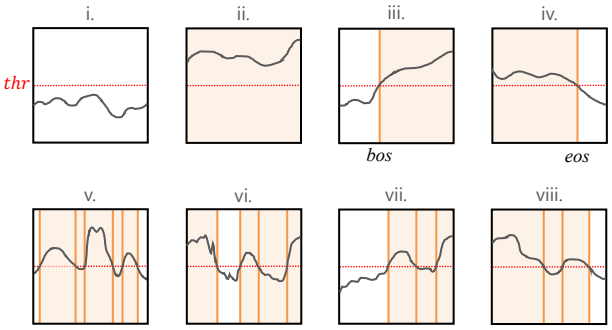


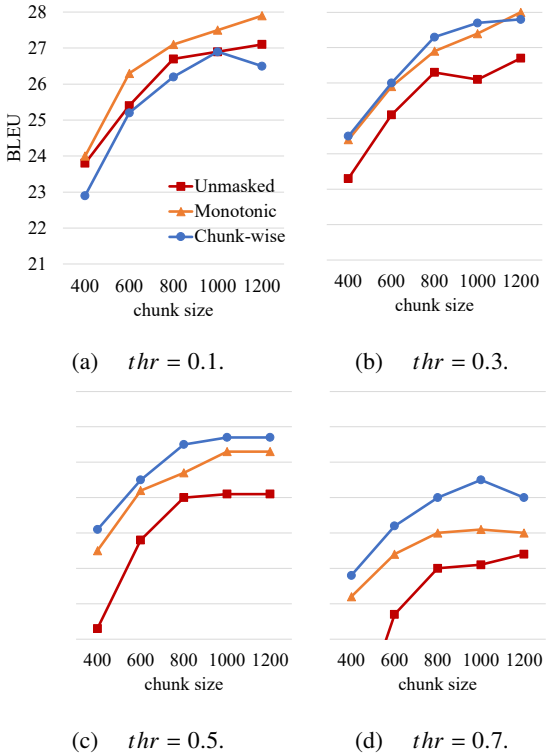
図 4: 音声分割モデルの予測パターン

A.2 各音声分割手法の BLEU スコア

表 1: 音声分割手法の翻訳精度。括弧内はトップラインに対する分割手法の比。

Chunk size	Unmasked	Monotonic	Chunk-wise
400	23.3 (0.943)	24.4 (0.988)	24.5 (0.992)
600	25.1 (0.926)	25.9 (0.956)	26 (0.959)
800	26.3 (0.913)	26.9 (0.934)	27.3 (0.948)
1000	26.1 (0.888)	27.4 (0.932)	27.7 (0.942)
1200	26.7 (0.908)	28 (0.952)	27.8 (0.946)

A.3 異なる閾値における音声分割の比較



(a) $thr = 0.1$. (b) $thr = 0.3$. (c) $thr = 0.5$. (d) $thr = 0.7$.

図 5: 異なる閾値 *thr* における音声分割モデルの比較。

図 5 は、閾値 *thr* が大きく分割が積極的になるほど、音声分割モデルの精度の差が顕著に表れる様子を示す。