音声機械翻訳のための音声翻訳コーパスに基づく発話分割

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

音声機械翻訳（Speech Translation; ST）において, 適切な翻訳単位（セグメント）への発話の分割は重要な課題である. 従来研究では, 音声区間検出（Voice Activity Detection; VAD）等を用いた無音区間を基準とした分割が多く行われてきた. しかし, 自然発話において無音区間は必ずしも意味的な境界を意味しないため, しばしば翻訳に不適切な境界で分割される. 本研究では, 音声翻訳コーパスの分割位置を教師とした, 無音区間に基づかない発話分割予測モデルの学習を提案する. TED talks の音声翻訳コーパスを用いた実験では, 提案手法が既存手法に対し最大 3pt の BLEU スコア向上を示した.

1　はじめに

機械翻訳（Machine Translation; MT）の中でも音声を扱う ST に特有の課題として, 連続音声の分割が挙げられる. テキストを入力とする通常の MT では, 句点や終止符を境界とした文単位への分割が一般的に行われる. 一方で, ST の入力である連続音声には明示的な境界記号が存在せず, セグメントの境界は自明ではない. モデルの学習時には, 音声翻訳コーパスに含まれる予め分割されたセグメントを使用できるが, 実行時には自動的な発話分割処理が必要である.

VAD 等を用いた, 無音区間に基づく分割 [1] は一般的な発話分割手法として知られる. オープンソースのツール（WebRTC VAD[1], pyannote.audio[2] [2]）を用いて容易に実行できるため音声認識 (Automatic Speech Recognition; ASR) や ST で広く用いられているが, 無音は必ずしも文境界と一致しない. 無音区間によって文を断片化する過剰分割 (over-segmentation）や, 短い無音区間を無視して 1 セグメント に複数文を含める過少分割（under-segmentation）により, ASR や ST の精度を低下させる問題が指摘されている [3].

本研究では, 音声翻訳コーパスのセグメント境界に基づく発話分割手法を提案する. 音声翻訳コーパスは, テキストに含まれる句点や終止符を基準として発話分割が行われているため, 無音より翻訳に適した分割境界を持つ. 音声翻訳コーパスのセグメント境界を, 入力音声から直接予測するモデルを学習することで, 連続音声を高精度に翻訳できると考えた. 提案手法には Transformer Encoder [4] を用い, 音声入力に対するフレームレベルの系列ラベリング問題としてセグメント境界の予測を学習した. 実験は ASR モデルと MT モデルからなる Cascade 方式の ST システム（Cascade ST）と, 単一のモデルで音声を直接テキストに翻訳する End-to-end 方式の ST システム (End-to-end ST）で行い, 両条件下で提案手法の有効性を示した.

2　関連研究

ST における自動発話分割の取り組みとして, マルコフ過程 [5, 6, 7] や条件付き確率場 [8, 9] を用いたモデル化が検討されてきた. また SVM を使用して, 言語モデルの確率と品詞タグ [10], 無音の長さ [11, 12] 等を手がかりに文境界を予測する手法が提案されている. これらの手法の多くは音声から得られる音響特徴量と, テキストから得られる言語特徴量の両方を利用する. そのため, 書き起こしを介さず直接音声を翻訳する End-to-end ST に適用することが難しい.

より近年の研究では, VAD を基にした分割手法が多く提案されている. Gaido ら [13], Inaguma ら [14] は, VAD の過剰分割に対応するため, 一定の長さまで VAD によるセグメントを連結して翻訳するヒューリスティックな手法を用いた. Gállego ら [15] は, 事前学習済みの ASR モデル wav2Vec 2.0 [16] による無音検出を行った. Yoshimura ら [17] は, RNN に基づく ASR モデルを用いて, CTC 系列のブランク（"_"）の連続を無音区間とみなし分割の基準に用いた. ASR に基づく音声分割は, 無音区間とみなす非文字記号

1） https://github.com/wiseman/py-webrtcvad
2） https://github.com/pyannote/pyannote-audio

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

の連続数の閾値をハイパーパラメータとして調節できるため, 従来の VAD より直感的に制御しやすい利点がある. しかし, これらの手法はいずれも無音区間のみを基準に分割するため, しばしば翻訳に不適切な境界で分割される.

また, 句読点復元モデル [18, 19, 20] や言語モデル [21, 22] を用いて ASR が出力した書き起こしテキストを文単位に再分割することで, MT の翻訳精度が向上することが知られており, Cascade ST の枠組みで広く利用されている. 書き起こしを介して再分割を行うこれらの手法は, End-to-end ST への適用が難しい他, 不適切な分割による ASR の認識誤りを防げない欠点がある. VAD を用いた発話分割による ASR の精度低下については4.2節でも言及する.

最後に本研究とより関連が強い, コーパスに基づく分割学習手法を紹介する. Wan ら [3] は, ASR 出力のセグメント境界を修正するモデルを, 映画やテレビの字幕コーパスを用いて学習した. Wang ら [23], Iranzo-Sánchez ら [24] は, 音声翻訳コーパスのセグメント境界を用いて RNN に基づく発話分割モデルを学習した. 言語特徴量を必要とするこれらの手法と異なり, 本研究の提案手法は音響特徴量のみを用いて発話分割を行う. そのため End-to-end ST への適用が容易である. また, 現在主流となっている Transformer に基づく ST への, 将来的な発話分割機能の統合を期待し, 分割モデルとして Transformer Encoder を採用した.

3 提案手法

本章では, 音声翻訳コーパスに基づく発話分割タスクの学習データ作成手順（3.1）, モデルの構造と動作（3.2）について説明を行う.

3.1 学習データ作成

本研究では, TED talks の音声翻訳コーパス MuST-C [25] を実験に用いる. MuST-C の音声セグメントは文単位のテキストを基準に分割されているため, 文単位の発話分割の学習データとして用いることができる.

具体的な学習データ作成例を図 1に示す. 音声入力に対する系列ラベリング問題としてセグメント境界の予測を学習するため, 連続する 2 つのセグメントを連結して音響特徴量の各フレームに対応するラベル $x \in \{0, 1\}$ を付与した. ラベル 0 と 1 はそれぞれ発話内・外に対応し, 開始・終了の時刻情報から作成する.

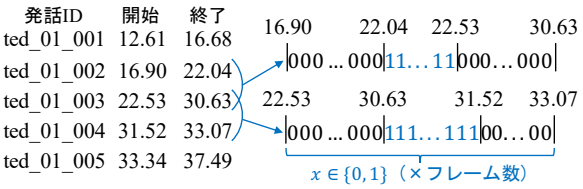

図 1 音声翻訳コーパスを用いたデータ作成の例

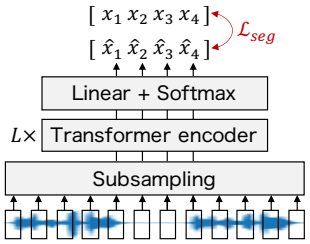

図 2 Transformer Encoder を用いた発話分割モデル

3.2 発話分割モデル

発話分割モデルの構成を図 2に示す. モデルは, Subsampling 層, L 層の Transformer Encoder, 出力のための変換層（Linear+Softmax）で構成され, 入力音声の各フレームのラベル $\hat{x}$ を出力する. Subsampling 層では系列長を 4 分の 1 にサブサンプリングする. 教師ラベル x にも同様にサブサンプリングを適用し長さを揃えた.

3.2.1 学習

3.1で作成したデータを用いてモデルの学習を行う. モデルは, 予測 $\hat{x}$ とラベル x 間のクロスエントロピー損失 $\mathcal{L}_{seg}(\hat{x}, x)$ の最小化を学習する（式 1）.

$$\mathcal{L}_{seg}(\hat{x}, x) = -\sum_{n=1}^{N} \left\{ w_s \log \frac{\exp(\hat{x}_{n,1})}{\exp(\hat{x}_{n,0} + \hat{x}_{n,1})} x_{n,1} + (1 - w_s) \log \frac{\exp(\hat{x}_{n,0})}{\exp(\hat{x}_{n,0} + \hat{x}_{n,1})} x_{n,0} \right\} \quad (1)$$

ここで, w_s は不均衡なラベルの重みを調節するハイパーパラメータである. ラベルの殆どは 0（発話内）であるため, ラベル 1（発話外）の損失にかかる重みを大きくして学習を行う. 予備実験により調整を行い $w_s = 0.9$ に設定した.

3.2.2 推論

推論の際は, 連続音声を固定長 T で区切ってモデルに入力し, 逐次的にラベルを予測する. その後, 予測したラベルにより連続音声を再分割し ST に渡して翻訳する.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

4 実験

提案手法の有効性を検証するために音声翻訳の実験を行った（4.1.1）．まず, ASR モデルと MT モデルからなる Cascade ST と, 単一の ST モデルからなる End-to-End ST の 2 方式の ST システムを構築した（4.1.2）．その後, 3 種類の発話分割手法（4.1.3）を ST システムの精度によって比較した（4.2）．

4.1 実験設定

4.1.1 タスク

音声翻訳コーパス MuST-C に含まれる, 約 408 時間の英語の講演音声と, 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 229,696 個, 1,423 個, 2,641 個である. 音響特徴量として, Kaldi [3] により抽出した 80 次元の対数メルフィルタバンク（FBANK）に 3 次元のピッチ情報を加えた 83 次元のベクトルを用いた. テキストデータは, SentencePiece [26] を用いて Byte Pair Encoding（BPE）によるサブワード分割を行った. サブワード辞書の最大語彙数は 8,000 とし, 英独の学習データを結合して辞書作成に用いた.

評価の際, 各手法（4.1.3）により自動分割した音声に対するモデルの出力を, 編集距離に基づくテキスト整列アルゴリズム [27] で評価データのセグメントと対応付けを行った後, WER や BLEU を測定した.

4.1.2 ST システム

各モデルの作成には ESPnet [4] [28] の Transformer の実装を用いた. Cascade ST の ASR, MT モデル及び End-to-end ST の ST モデルの設定を付録A.1に示す. ASR は, Transformer に CTC モデルを組み込んだ Hybrid CTC/attention [29] で学習を行った. CTC 損失にかかる重みは 0.3 とした. ST は学習済みの ASR の Encoder と MT の Decoder を用いてパラメータを初期化した. モデルの学習には学習データを用い, 重みをエポック毎に保存した. 最大エポック数まで学習後, 開発データで測定したスコア（ASR モデルに対して Accuracy, MT と ST モデルに対して BLEU）の高い 5 つの epoch の重みを平均して評価データで最終的なスコア（ASR モデルに対して WER, MT と ST モデルに対して BLEU）を測定した.

3） https://github.com/kaldi-asr/kaldi
4） https://github.com/espnet/espnet

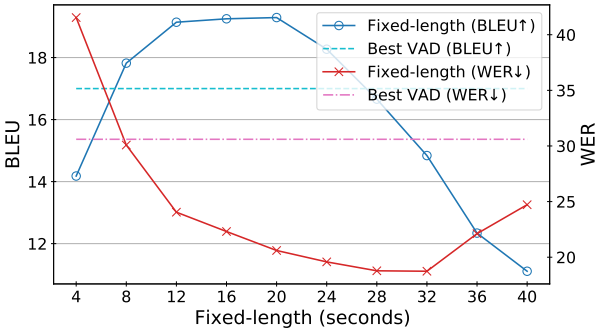

図 3 ベースライン（VAD, Fixed-length）の比較. 縦軸は ASR, MT モデルの BLEU と WER のスコアを示す.

4.1.3 発話分割手法

ベースライン 1: VAD WebRTC VAD を用いる. パラメータは Frame size={10, 20, 30ms}と Agressiveness={1, 2, 3}の範囲で 9 つの組み合わせを試行した.

ベースライン 2: Fixed-length Fixed-length は事前に設定した固定の長さで発話を区切る単純な方法である [7]. 音響や言語を考慮しないため不自然な分割が起こりやすいが, セグメント長が一定に保たれる利点がある. パラメータは, 4 から 40 秒まで 4 秒間隔でとった 10 通りの固定長を試行した.

提案手法 提案手法は3.2節で述べた発話分割モデルである. 学習時は3.1節で述べたように連続する 2 つのセグメントを連結したものを用いる. モデルの設定は付録A.1に示した ASR の Encoder と同一である. ただし, セグメントの連結により入力の平均長さがおよそ 2 倍になることを考慮し, ミニバッチ数を半分の 32, 勾配蓄積の数を 2 倍の 4 に設定した. 推論時の入力の固定長 T は予備実験により調整を行い 20 秒に設定した.

4.2 実験結果

4.2.1 ベースライン

図3に, Fixed-length の各長さにおける Cascade ST の ASR, MT モデルのスコアを示す. また比較のため, VAD で最良の BLEU を得た設定におけるスコアを直線で描画した（Best VAD）. VAD の各設定におけるスコアは付録A.2に示す. Fixed-length では, WER, BLEU ともに入力長に比例してスコアが向上していき, ある長さまで達するとそれ以降は低下した. この

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1 BLEU による各分割手法の翻訳精度の比較.

	Cascade ST	End-to-end ST
Oracle	23.6	19.0
Best VAD	17.0	13.7
Best Fixed-length	19.3	14.9
提案手法	**20.0**	**15.5**

表 2 発話分割のスコア.

	Precision	Recall	F1
提案手法	0.33	0.68	0.44

結果から, セグメントが長いほど自動分割による認識・翻訳精度の低下を防ぐことができると考えられる. 一方で, メモリ上の制約や学習データのセグメント長の分布などに依存した, 精度が頭打ちになる長さ上限が存在することも確認できる. そのため精度の低下を最低限に抑える適切な自動分割が重要である. また Best VAD は WER, BLEU ともに最良の Fixed-length の結果（Best Fixed-length）を下回った[5]. 無音区間に基づく分割により生じる過剰分割や過少分割が, ASR と ST の大きな精度低下に繋がったと考えられる.

4.2.2 提案手法

表 1に, 各手法によって分割した音声に対する翻訳テキストの BLEU スコアを示す. Oracle はコーパスのセグメント境界を用いた場合の BLEU スコアである. 提案手法は Cascade ST と End-to-end ST の両条件下で, 最良の VAD と Fixed-length の結果（Best VAD, Best Fixed-length）を上回った. Best VAD に対しては Cascade ST で 3pt, End-to-end ST で 1.8pt の向上が見られたことから, 音声翻訳コーパスを用いることでより文単位に近い適切な発話分割が行えたと考えられる. VAD と提案手法で分割した発話に対する ST システムの出力例を付録A.3に示す. 一方で, 提案手法は Oracle のスコアと比べて 3pt 以上の低下があり, 改善の余地が大きい. 表 2は, 提案手法の発話分割モデルに対する Precision, Recall, F1 スコアを示している. Precision が低く, Recall が高いことから, 提案手法では過剰にラベル 1（発話外）を予測し, 過剰分割が行われたと考えられる. 実際に, 評価データに含まれる Oracle のセグメント数が 2,641 に対して提案手法は 3,821 と, 頻繁にセグメントを分割していることが分かった. この結果を含む各手法のセグメントのデータ統計を付録A.4に示す. この問題は, セグメント境界とみなすラベル 1 の連続数に閾値を設けるなど, 再分割のアルゴリズムの工夫により改善できる可能性があるため, 今後も調査を続けたい.

5) 同様の傾向が先行研究 [13] でも示されている.

5 おわりに

本研究では, 音声翻訳コーパスを用いた発話分割手法を提案した. また Cascade ST と End-to-end ST の両条件下で実験を行い, 無音区間に基づく分割や固定長による分割と比較し, 提案手法の有効性を確認した. 今後は手法改善のための定性的な分析や, 既存手法との併用についての検討を進める. 自動分割を用いた翻訳に対するより信頼のおける評価手法の確立も重要な課題である. また, システムの単一化による高速化と誤りの伝播の低減を目指し, End-to-end ST への発話分割機能の統合を検討したい.

謝辞

本研究の一部は JSPS 科研費 JP21H05054 の助成を受けたものである.

参考文献

[1] 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.

[2] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. Pyannote. audio: neural building blocks for speaker diarization. In **ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 7124–7128. IEEE, 2020.

[3] 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.

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

[5] Andreas Stolcke, Elizabeth Shriberg, Rebecca A Bates, Mari Ostendorf, Dilek Zeynep Hakkani, Madelaine Plauche, Gökhan Tür, and Yu Lu. Automatic detection of sentence boundaries and disfluencies based on recognized words. In **5th International Conference on Spoken Language Processing (ICSLP)**, Vol. 2, pp. 2247–2250. Citeseer, 1998.

[6] Saab Mansour. Morphtagger: Hmm-based arabic segmentation for statistical machine translation. In **Proceedings**

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

of the 7th International Workshop on Spoken Language Translation: Papers, 2010.

[7] Mark Sinclair, Peter Bell, Alexandra Birch, and Fergus McInnes. A semi-markov model for speech segmentation with an utterance-break prior. In **Fifteenth Annual Conference of the International Speech Communication Association**, 2014.

[8] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[9] Wei Lu and Hwee Tou Ng. Better punctuation prediction with dynamic conditional random fields. In **Proceedings of the 2010 conference on empirical methods in natural language processing**, pp. 177–186, 2010.

[10] 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, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[11] Evgeny Matusov, Dustin Hillard, Mathew Magimai-Doss, Dilek Hakkani-Tur, Mari Ostendorf, and Hermann Ney. Improving speech translation with automatic boundary prediction. In **Proceedings of Interspeech 2007**, pp. 2449–2452, 2007.

[12] Sharath Rao, Ian Lane, and Tanja Schultz. Optimizing sentence segmentation for spoken language translation. In **Eighth Annual Conference of the International Speech Communication Association**. Citeseer, 2007.

[13] 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.

[14] 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, Bangkok, Thailand (online), August 2021. Association for Computational Linguistics.

[15] Gerard I Gállego, Ioannis Tsiamas, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà. End-to-end speech translation with pre-trained models and adapters: Upc at iwslt 2021. In **Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)**, pp. 110–119, 2021.

[16] 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.

[17] Takenori Yoshimura, Tomoki Hayashi, Kazuya Takeda, and Shinji Watanabe. End-to-end automatic speech recognition integrated with ctc-based voice activity detection. In **ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, pp. 6999–7003. IEEE, 2020.

[18] Eunah Cho, Jan Niehues, Kevin Kilgour, and Alex Waibel. Punctuation insertion for real-time spoken language translation. In **Proceedings of the Eleventh International Workshop on Spoken Language Translation**, 2015.

[19] Thanh-Le Ha, Jan Niehues, Eunah Cho, Mohammed Mediani, and Alex Waibel. **The KIT translation systems for IWSLT 2015**. 2015.

[20] Eunah Cho, Jan Niehues, and Alex Waibel. NMT-Based Segmentation and Punctuation Insertion for Real-Time Spoken Language Translation. In **Proceedings of Interspeech 2017**, pp. 2645–2649, 2017.

[21] Andreas Stolcke and Elizabeth Shriberg. Automatic linguistic segmentation of conversational speech. In **Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96**, Vol. 2, pp. 1005–1008. IEEE, 1996.

[22] Xiaolin Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. An efficient and effective online sentence segmenter for simultaneous interpretation. In **Proceedings of the 3rd Workshop on Asian Translation (WAT2016)**, pp. 139–148, 2016.

[23] Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. Online sentence segmentation for simultaneous interpretation using multi-shifted recurrent neural network. In **Proceedings of Machine Translation Summit XVII Volume 1: Research Track**, pp. 1–11, 2019.

[24] Javier Iranzo-Sánchez, Adrià Giménez Pastor, Joan Albert Silvestre-Cerdà, Pau Baquero-Arnal, Jorge Civera Saiz, and Alfons Juan. Direct segmentation models for streaming speech translation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 2599–2611, Online, November 2020. Association for Computational Linguistics.

[25] Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. Must-c: a multilingual speech translation corpus. In **2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 2012–2017. Association for Computational Linguistics, 2019.

[26] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In **EMNLP (Demonstration)**, 2018.

[27] 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.

[28] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. In **Proceedings of Interspeech 2018**, pp. 2207–2211, 2018.

[29] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. **IEEE Journal of Selected Topics in Signal Processing**, Vol. 11, No. 8, pp. 1240–1253, 2017.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

A 付録

A.1 Transformer の設定

ASR, MT, ST モデルの設定を表3に示す.

表 3 Transformer の設定. † バージョン 0.10.3.

設定（ESPnet† の変数名）	ASR	ST	MT
エポック数（epochs）	45	100	
Encoder 層の数（elayers）	12		6
Decoder 層の数（elayers）	6		
FNN の次元数（eunits, dunits）	2048		
Attention の次元数（adim）	256		
Attention のヘッド数（aheads）	4		
ミニバッチ数（batch-size）	64		96
勾配蓄積（accum-grad）	2		1
勾配クリッピング（grad-clip）	5		
学習率（transformer-lr）	5	2.5	1
ウォームアップ（transformer-warmup-steps）	25000		
ラベル平滑化（lsm-weight）	0.1		
ドロップアウト率（dropout-rate）	0.1		

A.2 ベースライン 1 のスコア

VAD の各設定における WER と BLEU スコアを表4に示す. 本文中の Best VAD は (2, 20ms) の設定を用いた結果である.

表 4 VAD の各設定における WER と BLEU スコア. †（Aggressiveness, Frame size）.

	Cascade ST		End-to-end ST
Segmentation	ASR	BLEU	BLEU
Oracle	12.6	23.6	18.97
WebRTC†			
(1, 30ms)	42.6	13.3	10.77
(1, 20ms)	40.8	14.0	11.23
(1, 10ms)	40.2	13.9	11.24
(2, 30ms)	33.5	16.4	12.98
(2, 20ms)	30.6	**17.0**	**13.65**
(2, 10ms)	**30.5**	16.9	13.60
(3, 30ms)	45.6	12.4	9.68
(3, 20ms)	45.9	12.1	9.62
(3, 10ms)	57.0	8.24	6.50

A.3 事例分析

Oracle のセグメント, 及び VAD と提案手法で自動分割した発話に対する ST システムの出力例を表5に示す.

表 5 Cascade ST における, 各分割手法ごとの ASR モデルと MT モデルの出力例. “■” はセグメント境界を示す.

Oracle (ASR): *bonobos are together with chimpanzees you aposre living closest relative* ■ **Oracle (MT)**: *Bonobos sind zusammen mit Schimpansen, Sie leben am nächsten Verwandten.* ■
Best VAD (ASR): *bonobos are* ■ *together with chimpanzees you aposre living closest relative that ...* **Best VAD (MT)**: *Bonobos sind es.* ■ *Zusammen mit Schimpansen leben Sie im Verhältnis zum ...*
提案手法 (ASR): *bonobos are together with chimpanzees you aposre living closest relative* ■ **提案手法 (MT)**: *Bonobos sind zusammen mit Schimpansen, Sie leben am nächsten Verwandten.* ■

表5の例で, VAD による分割では, 過剰分割と過少分割が生じ, 翻訳結果に Oracle のセグメントとの差異が生じている. 一方で提案手法では Oracle と近い境界で発話が分割され, Oracle のセグメントと同一の翻訳結果が得られた.

A.4 データ統計

評価データを各手法で分割したセグメントのデータ統計を表6に示す.

表 6 各手法により分割したセグメントのデータ統計. “% Filtered” は連続音声中の発話外の音声の割合を示す.

Segmentation	Oracle	WebRTC (2, 20ms)	提案手法
% Filtered	14.89	23.32	17.76
Num segm.	2641	2799	3823
Max len (s)	51.97	19.96	25.88
Min len (s)	0.19	0.58	0.05
Avg len (s)	5.66	4.81	3.87
Variance (s)	21.41	13.28	18.74

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).