JTubeSpeech: 音声認識と話者照合のために YouTube から構築される日本語音声コーパス

高道慎之介[1]，Kürzinger Ludwig[2]，佐伯高明[1]，塩田さやか[3]，渡部 晋治[4]
1 東京大学，2 ミュンヘン工科大学，3 東京都立大学，4 カーネギーメロン大学
shinnosuke_takamichi@ipc.i.u-tokyo.ac.jp

概要

本論文は YouTube 動画からの音声コーパス構築法を提案する．オープンな音声コーパスは音声言語処理の要だが，英語と中国語に比べ多くの言語でその整備が遅れている．そこで本研究は，言語にほぼ依存しないコーパス構築法を提案するとともに，それを日本語に適用し音声認識・話者照合のためのコーパスを構築する．構築スクリプトと動画リストをプロジェクトページにて公開している．

1 はじめに

深層学習の恩恵を受け，音声言語処理（例えば，音声認識や話者照合）[1]–[5] の性能が顕著に向上している．深層学習に基づく音声言語処理は多量の学習データを要するため，主要言語に限らず多くの言語のオープン音声コーパスを整備すべきである．しかしながら，英語や中国語 [6]–[12] に比べ，それ以外の言語のコーパス整備は遅れている．これは日本語においても例外ではない．

関連研究として，動画から音声コーパスを収集する研究がある [9], [13]–[15]．特に YouTube は多様なジャンル，環境，話者，アクセントの動画を有するため，YouTube 動画は有用な音声データと成り得る．Chen ら [16] と Fan ら [12] らはそれぞれ，英語音声認識との中国語話者照合のための音声コーパス構築法を提案している．言語依存の処理や手動の処理を含むこれらの研究と異なり，本研究は，言語非依存で音声コーパスを自動構築する方法論を目指す．この方法論の確立は，英語と中国語に限定されない音声コーパスの構築に有効である．

本稿では，音声認識 (ASR) と話者照合 (ASV) を対象として，（ほぼ）言語非依存でコーパスを自動構築する方法を提案するとともに，日本語音声コーパス JTubeSpeech を構築する．最初に，YouTube をクロールしてコーパスの候補となるテキストと音声の対を取得する．次に，音声認識用コーパスの構築では，end-to-end 音声認識モデルと CTC segmentation [17] に基づいて，テキストと音声の適合度合いを定量化しその適合対を抽出する．話者照合用コーパスの構築では，動画内の話者ベクトル分散に基づいて，単一話者動画を抽出する．これらの処理はほぼ言語非依存で実行される．本研究の貢献は以下のとおりである．

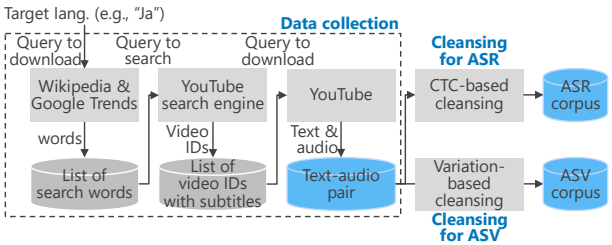

図 1 コーパス構築手順．

- 日本語音声認識・話者照合用の大規模コーパスを構築する．構築に利用した YouTube 動画リストを，プロジェクトページ[1)]で公開している．
- 多くの言語に適用可能なコーパス構築法を提案する．上記プロジェクトページでは，日本語以外にも対応したデータ収集スクリプトを公開している．

2 コーパス構築

図 1 にコーパス構築手順を示す．

2.1 データ収集

検索フレーズの作成： まず，動画検索に使用する検索フレーズを作成する．対象言語の Wikipedia 記事からハイパーリンク付きのフレーズを抽出する．Gigaspeech [16] と異なり，本稿では記事カテゴリを指定せずに全 Wikipedia 記事を使用する．また，Google Trends にて提供される急上昇検索フレーズも

1） https://github.com/sarulab-speech/jtubespeech

使用する．

字幕付き動画の取得： 次に，字幕付き動画の ID を取得する．フレーズ検索で動画 ID を取得し，その動画 ID のうち字幕を有するものをリスト化する．本稿では手動字幕（動画作成者による字幕）動画のみを対象とするが，本処理では同時に自動字幕（動画提供者による音声認識結果）動画も対象とする．最後に，動画をダウンロードして音声と手動字幕の対を作成する．この際，音声のフォーマットを 16kHz サンプリングのモノラルに統一する．

2.2 音声認識のためのデータ洗練

作成した音声と字幕のうち，音声認識に不適なデータ（音声と字幕が適合しない発話）を除去する．具体的には，CTC segmentation [17] を用いて音声と字幕の適合スコアを計算し，そのスコアに基づいて発話を除外する．また，多くの字幕のタイミング（発話開始終了時刻）は不正確であるため，再びアライメントをおこなう．

テキストに対する前処理：事前に最低限の言語依存処理を施す．本稿では，num2words[18] を用いて，数字列を読みの単語列に変換する．

アライメント： 各発話のタイミングの推定（アライメント）は，文献 [17] の CTC segmentation に従う．ただし，原著では発話前区間のスキップ機構を最初の発話のみに適用していたが，本稿では字幕により分割された各発話全てに適用する．アライメントには学習済み音声認識モデルによる推論を利用する．

スコアリングとデータ洗練：字幕のタイミングと音声の間で，CTC スコアを計算する．このスコアは音声と字幕と適合に関する対数確率である．CTC スコアに対し閾値を設け，閾値以下のデータを除去する．

長い音声の処理：1 つの字幕の時間長が数時間を超える場合がある．Transformer に基づく音声認識モデルは，時間長に対し 2 乗の計算量オーダーを要するため，長い音声を一度に処理できない．そのため本稿は recurrent neural network (RNN) に基づく音声認識モデルを利用する．64 GB のメモリ上で最大 500 秒分しか推論できない Transformer に対し，RNN は 2.7 時間分を推論できる．

長い音声を扱う際には，音声を小さなブロックに分割した後，各ブロックについて推論し，CTC スコアを再計算する．ブロックの最大サイズは，音声認識モデルの計算量と利用可能メモリに応じて決定する．音声信号の急さな分割は推論時の歪を生むため，ブロック区間を僅かに重複（本稿では 600 ミリ秒）させて分割する．各時刻の CTC スコアを計算した後に重複区間のスコアを除外して，最終的な CTC スコアとする．

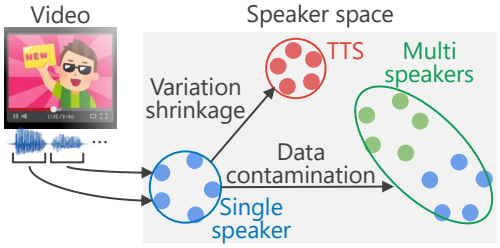

図 2 話者空間における分布の比較．

2.3 話者照合のためのデータ洗練

音声認識と異なり，話者照合は話者ラベルを必要とする．そこで本研究では，独話動画（単一話者動画）を教師なしに抽出する方法を提案する．この方法では，テキスト音声合成（TTS）による合成音声も同時に除去する．これは，合成音声の特徴が人間の音声と顕著に異なるためである．

非音声動画，短時間動画の除去：まず，音声を含まない動画を除外する．音声認識の場合と異なり，音声と字幕の適合を必要としないため，音声区間検出 (VAD) を用いて音声を検出する．字幕に基づいて切り出した発話に対し VAD を適用し，音声区間と判定されたもののみを使用する．また，後述の統計量計算のために，極端に短い動画を除外する．

話者空間における動画内変動の評価：単一話者動画を抽出するため，動画内の話者変動を計算する．図 2 に概念を示す．本研究では，学習済み深層学習モデルを用いて，話者表現ベクトルである *d*-vector[4] を抽出する．*d*-vector は発話ごとに計算され，動画内の話者変動はその分散として計算される．合成音声の発話間変動は極端に小さいため，合成音声の動画の話者変動は，単一話者動画に比べて小さくなると期待される．逆に，複数話者動画の話者変動は大きくなると期待される．以上の仮説より，分散に対する閾値を設けて単一話者動画を抽出する．実装では，t-SNE[19] により次元削減された *d*-vector を使用し，共分散行列の行列式の値を使用する．

YouTube チャンネルによるグループ化：話者照合用コーパスでは，同一話者が異なる話者として扱われることを避けなければならない．そのため本稿で

表 1　データ収集の結果.

Retrieved entity	Value
#search-terms	2.34M terms
#videos found in the search	11.9M videos
#videos with manual subtitles	0.11M videos
(#videos with auto subtitles)	4.96M videos

表 2　既存の日本語コーパス（上半分）および他言語コーパス（下半分）との比較.

Lang.	Task	Corpus	Open-source	Duration
Ja	ASR/ASV	JNAS [20]	No	90
Ja	ASR	CSJ [21]	No	600
Ja	ASR	LaboroTVspeech [22]	Yes	2,000
Ja	ASR	Common Voice [8]	Yes	2
Ja	ASV	Liveness [23]	No	4
Ja	ASR/ASV	**JTubeSpeech (ours)**	Yes	1,300/900
En	ASR	Librispeech [7]	Yes	982
En	ASR	Common Voice [8]	Yes	1,100
En	ASR	SPGISpeech [9]	Yes	5,000
En	ASR	GigaSpeech [16]	Yes	10,000
En	ASV	VoxCeleb [14]	Yes	2,800
Zh	ASR	Common Voice [8]	Yes	12
Zh	ASR	AISHELL-2 [24]	Yes	1,000
Zh	ASV	CN-Celeb [12]	Yes	1,000

は，同じ YouTube チャンネルに属する単一話者動画の話者を，単一の話者として扱う.

3　実験的評価

3.1　データ収集における評価

収集期間は 2021 年 2 月から 4 月である．表 1 より，1) 各検索フレーズから平均 5.09 個の動画 ID を収集できること，2) 収集した動画 ID の 0.92 %に手動字幕が，41.7 %に自動字幕がそれぞれ付与されていることが分かる．この収集により，110,000 個の手動字幕付き動画から約 10,000 時間の音声データを作成した.

表 2 は既存コーパスとの比較である．我々のコーパスのサイズは，次節以降の実験で使用されるものである．我々の音声認識コーパスは，LaboroTVspeech（日本語）[22] や Common Voice（英語）[8] と同程度である．また，我々の話者照合コーパスは，CN-Celeb（中国語）[12] と同程度のサイズであり，初の日本語コーパスである.

3.2　音声認識における評価

データ洗練： 2.2 節に示す方法を用いてデータ洗練を実施した．学習済み音声認識モデルは ESPnet LaboroTVspeech[22] のレシピ [25] を用いて学習した．表 3 に学習データ，テストデータの統計量を示す．本稿では (1) 2.3 節の方法で構築した単一話者動画（“single_speaker” もしくは “ss”）と (2) CTC スコアの高い 15,000 動画（“top_15k” もしくは “15k”）の 2 種類のサブセットを利用する．ただし，top_15k には単一話者動画以外も含まれ，この 2 つのサブセットの一部は重複していることに注意する．以降の節で登場する “train_ss_15k” は，この 2 つのサブセットを結合したものである.

表 3　音声認識における学習データと評価データの統計量．θ は CTC スコアに対する閾値.

	θ	# videos	# utts	hours
dev_easy_jun21	−0.3	110	785	0.7
eval_easy_jun21	−0.3	106	829	0.7
dev_normal_jun21	−1.0	128	1,036	1.1
eval_normal_jun21	−1.0	129	834	0.8
train_single_speaker	−0.3	1,297	14,797	12.7
train_single_speaker	−0.5	1,792	26,209	24.2
train_single_speaker	−1.0	2,906	66,563	71.9
train_single_speaker	−3.0	4,342	285,846	362.0
train_top_15k	−3.0	14,418	1,048,699	1087.1
train_ss_15k	−3.0	**17,761**	**1,270,124**	**1376.9**

評価データのデザイン：“single_speaker” の動画から，(1) −0.3 以上の CTC スコアの発話を有する 1,621 動画を選択，(2) 約 20%の動画をランダム抽出し，3,396 発話を選択，(3) 各発話を聴取し文と正しく対応する 1,614 発話を選択，(4) その発話を開発データ **dev_easy_jun21**（785 発話）と評価データ **eval_easy_jun21**（829 発話）に分割の手順で “easy” セットを作成した.

また本稿では，“normal” セットを作成した[2)]．このセットは，−1.0 以上の CTC スコアを持つ発話である．“easy” と同じ動画から該当発話を抽出し，上記の (3)(4) を同様に実行したのち，開発データ **dev_normal_jun21**（1,036 発話）と評価データ **eval_normal_jun21**（834 発話）を作成した.

音声認識の性能： ESPnet に実装されている，ハイブリッド CTC/アテンション構造 [26] に基づく Conformer [3], [27] を用いて，character error rate (CER) を評価した．設定の詳細は，ESPnet JTubeSpeech レシピ[3)] を参照されたい.

図 3 に結果を示す．“normal” セットの認識性能が “easy” セットのそれより悪いことから，CTC スコアによる認識難易度の付与が妥当と言える．また，閾値 θ を下げると学習データ量の増加とともにノイジーな書き起こしが混入されるが，音声認識性能は依然として改善することがわかる．最終的な CER は 5.2% (eval_easy_jun21) と

2)　“easy” と “normal” の定義は，スコアに基づいて客観的に決定されたものである．後述する音声認識実験において，この定義が妥当であることを示す.

3)　https://github.com/espnet/espnet/pull/3311

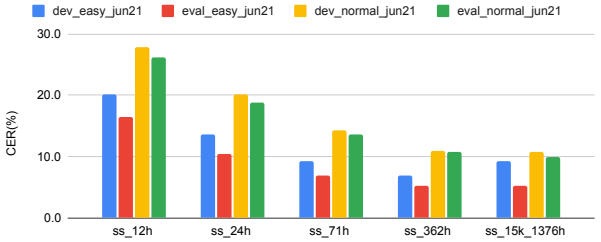

図 3　音声認識性能．閾値 θ を変えて学習データ量を 12, 24, 71, 362, 1, 376 時間に変更している．

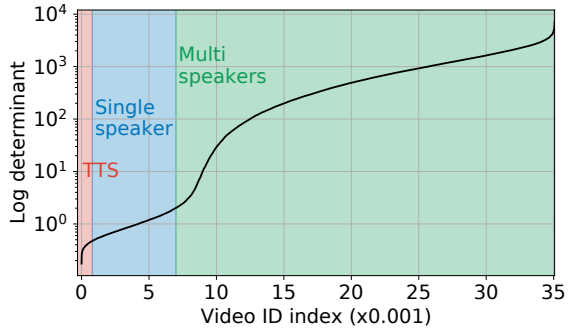

図 4　話者空間における動画内変動．

10.7% (eval_normal_jun21）であり，これらは他の日本語音声認識ベンチマーク (CSJ[21] では 4 – 6%, LaboroTVspeech[22] では 13%) と同程度である．全ての学習データを組み合わせた場合 (ss_15k_1376h) の結果を本図の右端に示している．dev_easy_jun21 以外の全ての性能が向上していることから，より難しいデータの性能を向上させるには学習データの増強が有効であると確認できる．

3.3　話者照合における評価

データ洗練：音声区間検出に py-webrtcvad[4] を，*d*-vector 抽出に学習済みモデル[5] を使用した．分散の計算は，10 発話以上から成る動画のみを使用した．pilot study として，本研究では 35, 000 動画（全体の約 30 %）を使用した．図 4 の結果より，横軸の値が 0 付近および 8 から 9 付近で急さな変化が見られる．図 2 の概念に基づいて 2 つの閾値を設け，TTS, single speaker, multi speakers のクラスラベルを各動画に付与する．

このラベルの品質を定量的に評価するため，クラスあたり 100 動画をランダムに抽出し，真のクラスラベルを手動付与した．表 4 に示す結果より，閾値に基づくラベルはおよそ正確であることが分かる．特に，single speaker のクラスに多数話者動画が含まれていないことから，提案法は単一話者動画を抽出できることがわかる．

4) https://github.com/wiseman/py-webrtcvad
5) https://github.com/yistLin/dvector

表 4　動画の種類に関するクラスタリングとアノテーション．

Classified \Annotated	TTS	single speaker	multi speakers
TTS	20	80	0
single speaker	**5**	**95**	**0**
multi speakers	1	36	63

話者照合の性能：single speaker のクラスに含まれる 1, 795 人の話者を学習と開発に，92 人の話者を登録と照合に用いた．学習と開発のデータ量はそれぞれ 127, 997 と 25, 392 発話であり，照合のデータ量は，276 発話の正例と 25, 116 発話の負例からなる 25, 392 発話である．話者埋め込みモデルは 4 層の畳み込み層，1 層の pooling 層，2 層の全結合層である．音声特徴量は 40 次元のメル周波数ケプストラム係数であり，話者ベクトルの次元は 512 次元とした．話者照合の評価基準は equal error rate (EER) である．確率的線形判別分析（PLDA）とデータ拡張は使用しなかった．

話者照合の性能は 10.9 %だった．単純なモデルを使用しているにも関わらず，Voxceleb1 と複雑なモデルを使用した場合 [14] と同程度の性能を達成している．故に，提案法は話者照合のためのデータを効率的に選択できることがわかる．

4　まとめ

本稿では，言語非依存でコーパスを自動構築する方法を提案し，1, 300 時間の日本語音声認識用コーパス，900 時間の日本語話者照合用コーパスを構築した．今後は，提案法を日本語以外の言語に適用する．

本稿の執筆時点（2022 年 1 月初頭）において，本稿の内容に加え以下の内容をプロジェクトページで公開している．音声認識と話者照合に限らず，多様な音声言語処理タスク，自然言語処理タスクで利用可能である．

- **多言語の動画 ID リスト：**YouTube 動画から取得可能な約 70 言語のうち，より多くの動画 ID を取得できた 30 言語の動画 ID リストを公開している．
- **自動字幕の取得スクリプト：**本稿で扱った手動字幕ではなく，自動字幕を取得し整形するスクリプトを公開している．

謝辞： 有意義な議論をくださった Hiromasa Fujihara 氏，GigaSpeech チーム（特に Guoguo Chen 氏，Shuzhou Chai 氏）に感謝する．本研究の実施にあたり，ジョンスホプキンス大学 HLTCOE クラスタと National Science Foundation grant number ACI-1548562 から支援を受けている, Extreme Science and Engineering Discovery Environment (XSEDE) [28] を使用した．また, NSF award number ACI-1445606 の支援を受けている, Pittsburgh Supercomputing Center (PSC) の Bridges system [29] を使用した．また，本研究は，JSPS 科研費 19K20271, 21H04900, 21H05054（基盤技術開発の内容），JST ムーンショット型研究開発事業 JPMJMS2011（多言語技術の内容），ROIS-DS-JOINT (030RP2021)，セコム財団挑戦的研究助成の助成を受けた．

References

[1]G. E. Dahl, D. Yu, L. Deng, *et al.*, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[2]A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, 2014, pp. 1764–1772.

[3]A. Gulati, J. Qin, C.-C. Chiu, *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[4]E. Variani, X. Lei, E. McDermott, *et al.*, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*, 2014, pp. 4080–4084.

[5]D. Snyder, D. Garcia-Romero, G. Sell, *et al.*, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[6]C. Cieri, D. Miller, and K. Walker, "The fisher corpus: A resource for the next generations of speech-to-text.," in *LREC*, vol. 4, 2004, pp. 69–71.

[7]V. Panayotov, G. Chen, D. Povey, *et al.*, "Librispeech: An asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[8]R. Ardila, M. Branson, K. Davis, *et al.*, "Common Voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[9]P. K. O'Neill, V. Lavrukhin, S. Majumdar, *et al.*, *SPGISpeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition*, 2021.

[10]Y. Liu, P. Fung, Y. Yang, *et al.*, "HKUST/MTS: A very large scale Mandarin telephone speech corpus," in *International Symposium on Chinese Spoken Language Processing*, 2006, pp. 724–735.

[11]J. Du, X. Na, X. Liu, *et al.*, "Aishell-2: Transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.

[12]Y. Fan, J. Kang, L. Li, *et al.*, *CN-CELEB: A challenging Chinese speaker recognition dataset*, 2019.

[13]S. Abu-El-Haija, N. Kothari, J. Lee, *et al.*, *YouTube-8M: A large-scale video classification benchmark*, 2016.

[14]A. Nagrani, J. S. Chung, W. Xie, *et al.*, "Voxceleb: Large-scale speaker verification in the wild," *Computer Science and Language*, 2019.

[15]D. Galvez, G. Diamos, J. M. C. Torres, *et al.*, "The People' s Speech: A large-scale diverse English speech recognition dataset for commercial usage," 2021, https://openreview.net/forum?id=R8CwidgJ0yT.

[16]G. Chen, S. Chai, G. Wang, *et al.*, "GigaSpeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[17]L. Kürzinger, D. Winkelbauer, L. Li, *et al.*, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *Speech and Computer*, 2020, pp. 267–278.

[18]V. Dupras, M. Grigaitis, and T. Ogawa, *Num2words: Modules to convert numbers to words.* https://github.com/savoirfairelinux/num2words, 2021.

[19]L. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, pp. 2579–2605, 2008.

[20]K. Itou, M. Yamamoto, K.Takeda, *et al.*, "JNAS: Japanese speech corpus for large vocabulary continuous speech recognition research," *Journal of the Acoustical Society of Japan (E)*, vol. 20, no. 3, pp. 199–206, 1999.

[21]K. Maekawa, H. Koiso, S. Furui, *et al.*, "Spontaneous speech corpus of Japanese.," in *Proc. LREC*, 2000, pp. 947–952.

[22]S. Ando and H. Fujihara, "Construction of a large-scale Japanese ASR corpus on TV recordings," in *ICASSP*, 2021, pp. 6948–6952.

[23]S. Shiota, F. Villavicencio, J. Yamagishi, *et al.*, "Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification," in *Sixteenth annual conference of the international speech communication association*, 2015.

[24]J. Du, X. Na, X. Liu, *et al.*, "AISHELL-2: Transforming Mandarin ASR Research Into Industrial Scale," *ArXiv*, 2018.

[25]V. authors, *Espnet/egs2/laborotv/asr1 · espnet/espnet*, https://github.com/espnet/espnet/tree/master/egs2/laborotv/asr1, 2021.

[26]S. Watanabe, F. Boyer, X. Chang, *et al.*, "The 2020 ESPNet update: New features, broadened applications, performance improvements, and future plans," *arXiv preprint arXiv:2012.13006*, 2020.

[27]P. Guo, F. Boyer, X. Chang, *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP*, 2021, pp. 5874–5878.

[28]J. Towns, T. Cockerill, M. Dahan, *et al.*, "Xsede: Accelerating scientific discovery computing in science & engineering, 16 (5): 62–74, sep 2014," *URL https://doi. org/10.1109/mcse*, vol. 128, 2014.

[29]N. A. Nystrom, M. J. Levine, R. Z. Roskies, *et al.*, "Bridges: A uniquely flexible hpc resource for new communities and data analytics," in *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, 2015, pp. 1–8.