音声認識出力の曖昧性に頑健な音声翻訳のための音声認識の精度ごとの性能比較

胡 尤佳[1]　須藤 克仁[1,2]　Sakriani Sakti[1,2,3]　中村 哲[1,2]
[1]奈良先端科学技術大学院大学 [2]理化学研究所 革新知能統合研究センター AIP
[3]北陸先端科学技術大学院大学
{ko.yuka.kp2, sudoh, s-nakamura}@is.naist.jp; ssakti@jaist.ac.jp

概要

音声認識出力の曖昧性は，発音が似ている単語の関係を表していると考えられ，End-to-End 音声翻訳においても，この曖昧性を考慮したモデルが必要となる．従来研究では，Multi-task 学習の End-to-End 音声翻訳において，音声認識出力の分布による Sub-task の学習で精度の向上が見られた．本研究では，音声認識出力の曖昧性に頑健な音声翻訳のためにどのような音声認識モデルが効果的か，また，Main-task と Sub-task の出力の関係について比較し，分析した．

1　背景と関連研究

音声翻訳 (Speech Translation; ST) は，原言語音声を入力として目的言語テキストを出力する技術であり，音声認識 (Automatic Speech Recognition; ASR) と機械翻訳 (Machine Translation; MT) をつなぎ合わせた Cascade モデルと，原言語の音声を直接目的言語のテキストに翻訳する End-to-End モデルが考えられる．近年ではニューラルネットワークを用いた系列変換技術により End-to-End モデルの研究が進んでいる．しかしながら，End-to-End モデルでは学習に原言語音声と目的言語テキストを用いるためデータが限られ，比較的容易にデータを入手できる Cascade モデルと比較して精度が低くなる傾向がある．

このような End-to-End モデルの精度を向上させるアプローチの 1 つとして，原言語音声から目的言語テキストへの翻訳に原言語テキストへの音声認識を Sub-task として追加する Multi-task 学習 [1] が挙げられる．しかし，一般的な Multi-task 学習で使われる cross entropy (CE) loss は，正解トークンとの損失を計算し，発音の似た予測結果と，発音の似ていない予測結果の損失が同じになる可能性があるという問題がある．End-to-End 音声翻訳においてもこのような発音の類似度，聞き間違えを考慮した翻訳が必要となる．

関連研究として，Osamura ら [2] は，Cascade モデルにおいて，One-hot ベクトルの代わりに，音声認識の事後確率分布を用いて機械翻訳をチューニングし，音声認識の曖昧性に対する頑健性を向上させた．また，Chuang ら [3] は，Multi-task End-to-End 音声翻訳における ASR-task で，予測単語と正解単語の埋め込みベクトルのコサイン類似度を損失に利用し，意味の類似度の頑健な学習を実現した．

以上から着想を得た上で，我々は，Multi-task 学習で音声翻訳を学習する際に，音声認識の事後確率分布を用いた学習をする手法を提案し [4]，音声認識出力の曖昧性に対する頑健性の向上に寄与することを示した．音声認識の事後確率分布は，発音が似ている単語が同じようなスコアを持つことが期待され，単語間の発音の類似度の情報を保持する．これを reference として用いることで，音声認識出力の曖昧性に対して頑健な音声翻訳が学習できたと期待できる．しかし，音声翻訳に利用可能な音声認識モデルの性能や，ST-task と ASR-task の出力の性能の関係が確認されていなかった．

本研究では，音声認識出力の曖昧性を考慮した音声翻訳において，どのような音声認識モデルが効果的か，また Main-task と Sub-task の出力の関係を比較し，分析した．

2　Multi-task End-to-End 音声翻訳

$\mathbf{X} = (x_1, ..., x_T)$ を原言語の入力音声に対する音響特徴量の系列，$\mathbf{T} = (t_1, ..., t_N)$ を目的言語テキストのトークン系列，$\mathbf{S} = (s_1, ..., s_M)$ を原言語テキストのトークン系列とする．v を語彙集合 V の元とすると，i 番目の目的言語記号の事後確率は以下の式で

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

表される

$$P_{\mathrm{ST}}(t_i = v) = p(v|\mathbf{X}, t_{<i}). \quad (1)$$

ST の学習時の損失関数 $\mathcal{L}_{\mathrm{ST}}$ は，CE loss を用いて以下の式で表される

$$\mathcal{L}_{\mathrm{ST}} = -\sum_{i=1}^{N}\sum_{v\in V}\delta(v, t_i)\log P_{\mathrm{ST}}(t_i = v). \quad (2)$$

式中の $\delta(v, t_i)$ は，$v = t_i$ のとき 1，そうでなければ 0 とする．

Encoder により隠れベクトルに変換され，その後 ST-task (Main-task) の Decoder と ASR-task (Sub-task) の Decoder の両方を用いて学習される．v を語彙集合 V の元とすると，i 番目の目的言語記号の事後確率は以下の式で表される

$$P_{\mathrm{ASR}}(s_i = v) = p(v|\mathbf{X}, s_{<i}). \quad (3)$$

ASR 学習時の損失関数 $\mathcal{L}_{\mathrm{ASR}}$ は以下の式で表される

$$\mathcal{L}_{\mathrm{ASR}} = -\sum_{i=1}^{M}\sum_{v\in V}\delta(v, s_i)\log P_{\mathrm{ASR}}(s_i = v). \quad (4)$$

ST-task の損失関数を $\mathcal{L}_{\mathrm{ST}}$，ASR-task の損失関数を $\mathcal{L}_{\mathrm{ASR}}$，$\mathcal{L}_{\mathrm{ASR}}$ に対する重みを λ_{ASR} とすると，学習時全体の損失関数 $\mathcal{L}$ は以下の式で表される

$$\mathcal{L} = (1 - \lambda_{\mathrm{ASR}})\mathcal{L}_{\mathrm{ST}} + \lambda_{\mathrm{ASR}}\mathcal{L}_{\mathrm{ASR}}. \quad (5)$$

3 実験に用いる手法

以下の二種類の損失関数を，音声認識出力を reference とした soft label から計算される $\mathcal{L}_{\mathrm{soft}}$ として用い，比較，分析をした．また，本稿では，$\mathcal{L}_{\mathrm{soft}}$ に対して，式 4 における $\mathcal{L}_{\mathrm{ASR}}$ を，hard label (正解系列) による CE loss として $\mathcal{L}_{\mathrm{hard}}$ と表す．

3.1 ASR-PBL: ASR Posterior-based Loss [5]

ASR-task において，事前学習された ASR の事後確率分布のベクトルを reference として用いる．ASR 事後確率分布は，事前学習された ASR を用いて得られた，各トークンに対するスコアを持ったベクトルの softmax を取り，soft label とする．soft label において i 番目のトークン v のスコアを $P_{\mathrm{soft}}(i, v)$ とすると，提案手法よる損失 $\mathcal{L}_{\mathrm{soft}}$ は以下の式で表される

$$\mathcal{L}_{\mathrm{soft}} = -\sum_{i=1}^{M}\sum_{v\in V}P_{\mathrm{soft}}(i, v)\log P_{\mathrm{ASR}}(s_i = v). \quad (6)$$

本実験では，$\mathcal{L}_{\mathrm{ASR}}$ を以下の式として定義し，$\mathcal{L}_{\mathrm{hard}}$ と $\mathcal{L}_{\mathrm{soft}}$ の割合を，重み λ_{soft} で調整できるようにし，

$$\mathcal{L}_{\mathrm{ASR}} = (1 - \lambda_{\mathrm{soft}})\mathcal{L}_{\mathrm{hard}} + \lambda_{\mathrm{soft}}\mathcal{L}_{\mathrm{soft}}. \quad (7)$$

式 7 の損失を ASR Posterior-based Loss といい，本稿では ASR-PBL と表す．

3.2 ASR-SBL: ASR Sequence-based Loss

ASR-PBL では，ASR 事後確率分布を reference として用いていたが，ASR 出力の One-best 系列を reference とした損失も考えることができる．この場合，どの単語とどの単語が間違いやすいかという情報を保持し，音声認識出力の誤りに対して頑健な音声翻訳が期待できる．$\hat{\mathbf{S}} = (\hat{s}_1, ..., \hat{s}_M)$ を，ASR モデルによって予測された One-best の原言語テキストのトークン系列とすると，$\mathcal{L}_{\mathrm{soft}}$ は以下の式で表される

$$\mathcal{L}_{\mathrm{ASR}} = -\sum_{i=1}^{M}\sum_{v\in V}\delta(v, \hat{s}_i)\log P_{\mathrm{ASR}}(\hat{s}_i = v). \quad (8)$$

本実験では，hard loss と式 8 の soft loss を式 7 のように重み付き和で足し合わせた損失を，ASR Sequence-based Loss といい，本稿では ASR-SBL と表す．

4 実験

本実験ではデータセットとして，Fisher Spanish Corpus (Spanish-English) [6] と，MuST-C TED-talks (English-German) [7] を用いた．本稿では，それぞれを Fisher, MuST-C とし，以下の実験をした．

- Fisher: ASR-PBL, ASR-SBL
- MuST-C: ASR-SBL

音響特徴量は，Kaldi [8] により抽出した，3 次元の pitch が付加された 83 次元の Fbank+pitch を用い，train データは，音声のスピードを 0.9 倍と 1.1 倍に調整したものを加えた．テキストは MuST-C の目的言語テキスト以外は，句読点，記号を取り除き小文字化し，音響特徴量はフレーム長 3000，テキストは文字数が 400 より大きいものを取り除いた．Tokenizer は SentencePiece [9] を用い，最大語彙数を Fisher は 1000，MuST-C は 8000 とした．ASR，ST モデルは ESPnet [10] を用い，Transformer [11] により作成した．ST は，Fisher が batch size: 64, accum grad: 4, MuST-C が batch size: 32, accum grad: 8，それ以外の基本的なモデルの設定は ESPnet のデフォルトの値に従った．ST モデルは，epoch 30 で学習した後，dev データの BLEU スコア [12] が高いモデルを 5 つ model averaging し，Fisher では Fisher test, MuST-C では tst-COMMON で評価した．本実験では，式 5, 7 における $\lambda_{\mathrm{ASR}} = 0.5$, Fisher ASR-PBL: $\lambda_{\mathrm{soft}} = \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$,

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

	WER	BLEU			
		ASR-PBL		ASR-SBL (LSM)	
ASR model	Soft-label	soft-0.5	soft-1.0	soft-0.5	soft-1.0
None (Single-task)	-	40.66			
None (CE)	-	43.83			
None (CE-LSM)	-	45.16			
Epoch-6	39.7	**45.34**	41.71	44.68	41.71
Epoch-8	35.3	**45.33**	43.54	**45.63**	42.34
Epoch-10	28.9	**45.76**	42.93	**45.86**	43.79
Attn-Specaug	14.1	**45.29**	43.73	**45.34**	44.34
Attn	9.3	**46.04**	44.53	**45.35**	44.03
Attn-CTC	7.8	44.93	43.98	44.87	44.40
Attn-CTC-Specaug	6.7	44.66	44.82	**45.75**	44.76

表 1 Fisher における soft-0.5, 1.0 での ASR モデルごとの ASR-PBL と ASR-SBL(LSM) の結果.

	WER	BLEU			
		ASR-PBL		ASR-SBL (LSM)	
ASR model	Soft-label	soft-0.5	soft-1.0	soft-0.5	soft-1.0
None (Single-task)	-	17.68			
None (CE)	-	20.40			
None (CE-LSM)	-	21.08			
Attn-Specaug	9.7	-	-	21.07	20.70
Attn	5.2	-	-	21.01	20.87

表 2 MuST-C における soft-0.5, 1.0 での ASR モデルごとの ASR-SBL(LSM) の結果.

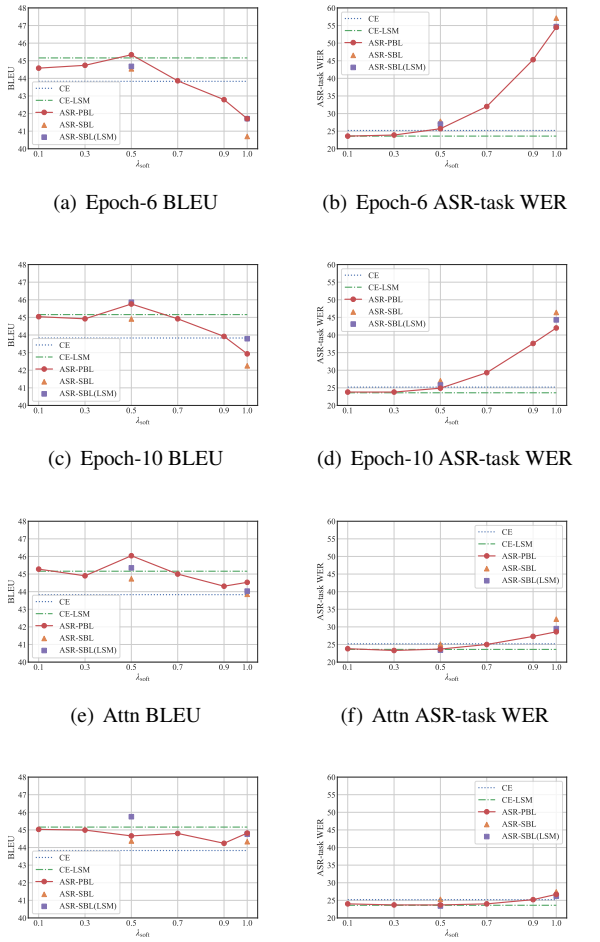
Fisher ASR-SBL: $\lambda_{\text{soft}} = \{0.5, 1.0\}$, MuST-C ASR-SBL: $\lambda_{\text{soft}} = \{0.25, 0.5, 0.75, 1.0\}$ とした．L_{ST} は全て label smoothing weight 0.1 で label smoothing した CE loss(以下，CE-LSM) を用いた．soft label の作成に必要な事前学習された ASR モデルは，実験に用いる soft label の WER が高いものから低いものを用意し，それらから生成された soft label を用いて ASR-PBL, ASR-SBL の学習に用いた (付録 7.1)．ASR-SBL では，CE loss と同様，label smoothing を加えた設定 (ASR-SBL(LSM)) と加えていない設定 (ASR-SBL) で実験した．

5 実験結果と分析

Fisher test における ST の BLEU の結果を表 1 に示す．(soft-0.5 は $\lambda_{\text{soft}} = 0.5$ を意味する．また，baseline を CE, CE-LSM とし，それらより BLEU が向上したものを太字で示す．) また，それぞれの Fisher の ST モデルの λ_{soft} ごとの ST-task BLEU と ASR-task WER の変化を図 1 に示す (全ての図は付録 7.2 を参照)．MuST-C (tst-COMMON) における ST の BLEU の結果を表 2 に示す．また，それぞれの MuST-C モデルの λ_{soft} ごとの ST-task BLEU と ASR-task WER の変化を図 2 に示す．

表 1 から，soft-0.5 の ASR-PBL と ASR-SBL において，ほとんどの場合で CE, CE-LSM と比較して精度の向上が見られ，大体の場合 soft-1.0 を上回る結果となった．よって，ASR-SBL が音声認識誤りに頑健なモデルの作成に効果的であることがわかった．図 1 から，ASR-PBL, ASR-SBL の両方で，soft-0.5 のときに BLEU が一番高い値を取ることが多い結果になった．ASR-SBL に関しては図 1 から，label smoothing を入れた ASR-SBL(LSM) が多くの場合 ASR-SBL を上回ったため，表 1 に ASR-SBL(LSM) のみを記載しているが，label smoothing を入れていない ASR-PBL が ASR-SBL(LSM) と同程度の性能を出すことができていることが分かり，分布レベルの学習によって label smoothing に近い効果も得られていることが予想される．soft-1.0 の結果は，ASR-PBL と ASR-SBL の両方において，Epoch-6 から Attn-CTC-Specaug へと WER が低いモデルになるにつれて，BLEU が高くなっていることが分かり，誤りが大きすぎるモデルを用いると精度の低下がみられることが分かる．

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



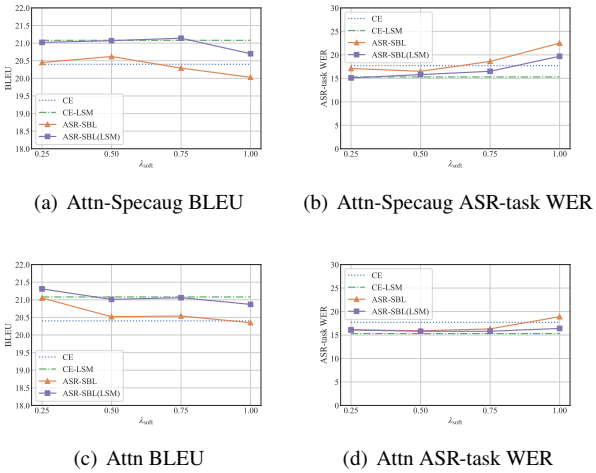
(a) Epoch-6 BLEU (b) Epoch-6 ASR-task WER

(c) Epoch-10 BLEU (d) Epoch-10 ASR-task WER

(e) Attn BLEU (f) Attn ASR-task WER

(g) Attn-CTC-Specaug BLEU (h) Attn-CTC-Specaug ASR-task WER

図 1 Fisher (Fisher test) の BLEU と ASR-task WER.



(a) Attn-Specaug BLEU (b) Attn-Specaug ASR-task WER

(c) Attn BLEU (d) Attn ASR-task WER

図 2 MuST-C (tst-COMMON) の BLEU と ASR-task WER.

図 1 からも，soft-1.0 に注目すると，WER が低いモデルの結果になるにつれて，ASR-task WER が高い状態から下がっていき，BLEU の向上が見られる．

また，Epoch-6, 8, 10 のような WER が高いモデルでの実験に関しても hard loss と混ぜることにより，BLEU の向上が見られ，学習時の label に曖昧性や誤りが perturbation となって加えられることによる効果とみられる．しかし，Attn-CTC や Attn-CTC-Specaug といった，WER が低いモデルに関しては，ASR-PBL の soft-0.5 において BLEU の向上が見られなかった．図 1 からも，Attn-CTC と Attn-CTC-Specaug の λ_{soft} ごとの ASR-task WER の差は小さくなり収束しているが，それに伴った BLEU の向上は λ_{soft} ごとに見ても確認できなかった．このことは，Fisher が電話の日常会話音声を収録したもので，聞き返しが必要となるような聞き間違えが起こりやすいデータと考えられ，信頼度が高く分布が鋭いものより滑らかな分布が Fisher における ASR-PBL では効果的であると考えられる．

表 2 から，MuST-C では，CE の baseline と比較して BLEU の向上は見られるものの，CE-LSM を上回る結果はなく，soft-0.5 が CE-LSM と同じぐらいの結果になったことがわかる．図 2 から，ASR-task WER が滑らかで，BLEU の変動があまり見られず，今回実験に用いた soft label に含まれる誤りが少ないことから，hard loss と比較して大きな変化がなかったとみられる．今後はより誤りが多い soft label で学習した場合や，滑らかな分布で学習した場合の精度の変化を調査する必要がある．

6 まとめと今後の展望

本研究では，音声認識の事後確率分布，誤りを用いて，End-to-End 音声翻訳を学習する方法で，学習に有効な音声認識の性能は，soft loss のみを用いる場合は，誤りが少ないモデルがより効果的だが，ASR-PBL で hard loss と soft loss を混ぜて使う場合は，誤りがとても少ないモデルだと効果が下がることがわかり，鋭い分布よりも滑らかな分布が有効であることがわかった．また，ST-task の BLEU が高い場合，ASR-task の WER も下がる傾向があることが分かった．

今後の課題として，学習の際に適切な分布がどのような特徴を持っているのかの定量的な調査 (分布の鋭さなど) を考えている．また，本研究では train set の数だけラベルを Decode する必要がある点でコストが高いため，生成ラベルを削減して実現できる手法も考えている．

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

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] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. Sequence-to-sequence models can directly translate foreign speech. In Francisco Lacerda, editor, **Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017**, pp. 2625–2629. ISCA, 2017.

[2] Kaho Osamura, Takatomo Kano, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. Using spoken word posterior features in neural machine translation. **Proceedings of the 15th International Workshop on Spoken Language Translation, 181-188, Oct. 2018**.

[3] Shun-Po Chuang, Tzu-Wei Sung, Alexander H. Liu, and Hung-yi Lee. Worse wer, but better bleu? leveraging word embedding as intermediate in multitask end-to-end speech translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020**, pp. 5998–6003. Association for Computational Linguistics, 2020.

[4] 胡尤佳, 須藤克仁, Sakriani Sakti, 中村哲. 音声認識仮説の曖昧性を考慮する Multi-task End-to-End 音声翻訳. 言語処理学会第 27 回年次大会 (NLP2021), 2021.

[5] Yuka Ko, Katsuhito Sudoh, Sakriani Sakti, and Satoshi Nakamura. ASR Posterior-Based Loss for Multi-Task End-to-End Speech Translation. In **Proc. Interspeech 2021**, pp. 2272–2276, 2021.

[6] Christopher Cieri, David Miller, and Kevin Walker. The fisher corpus: a resource for the next generations of speech-to-text. In **Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal**. European Language Resources Association, 2004.

[7] 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.

[8] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In **IEEE 2011 workshop on automatic speech recognition and understanding**, No. CONF. IEEE Signal Processing Society, 2011.

[9] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 66–71, 2018.

[10] 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. **Proc. Interspeech 2018**, pp. 2207–2211, 2018.

[11] 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.

[12] 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, 2002.

[13] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. In Gernot Kubin and Zdravko Kacic, editors, **Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019**, pp. 2613–2617. ISCA, 2019.

[14] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In William W. Cohen and Andrew W. Moore, editors, **Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006**, Vol. 148 of **ACM International Conference Proceeding Series**, pp. 369–376. ACM, 2006.

[15] 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/).

7 付録 (Appendix)

7.1 事前学習 ASR モデルの設定

7.1.1 Fisher

1. Epoch-6, 8, 10
   - Attention で epoch 30 まで学習した際に保存された，epoch 6, 8, 10 のモデル．
2. Attn-Specaug
   - Attention+SpecAugment [13] で epoch 50 まで学習した中で，dev データの accuracy が最も高かったモデル．
3. Attn
   - Attention で epoch 30 まで学習した中で，dev データの accuracy が最も高かったモデル．
4. Attn-CTC
   - Hybrid CTC/Attention [14, 15] で epoch 50 まで学習した中で，dev データの accuracy が最も高かったモデル．Attention に対する CTC の重みは 0.3.
5. Attn-CTC-Specaug
   - Hybrid CTC/Attention + SpecAugment で epoch 50 まで学習した中で，dev データの accuracy が最も高かったモデル．Attention に対する CTC の重みは 0.3.

ASR WER	Soft-label WER
Epoch-6	39.7
Epoch-8	35.3
Epoch-10	28.9
Attn-Specaug	14.1
Attn	9.3
Attn-CTC	7.8
Attn-CTC-Specaug	6.7

表 3 Fisher の soft label における WER.

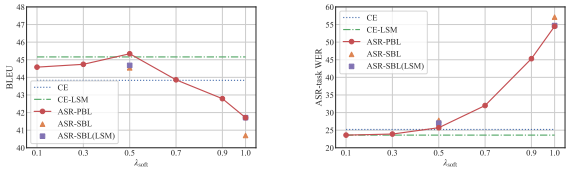
7.1.2 MuST-C

Total epoch: 45 / Batch size: 64 / Accum grad: 4

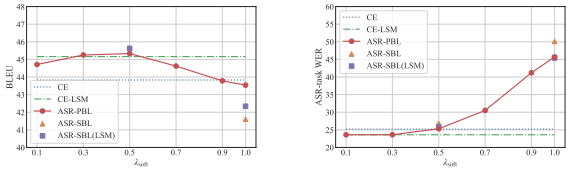
ASR WER	Soft-label WER
Attn-Specaug	9.7
Attn	5.2

表 4 MuST-C の soft label における WER.

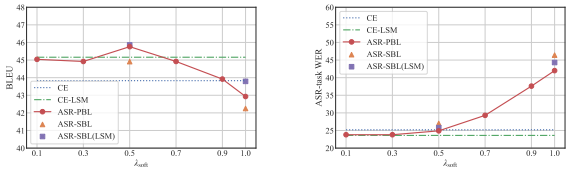
7.2 Fisher の BLEU と ASR-task WER (全ての図)



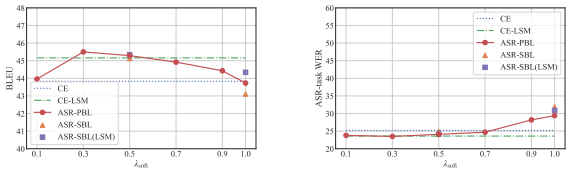
(a) Epoch-6 BLEU (b) Epoch-6 ASR-task WER



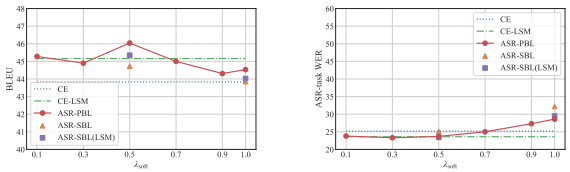
(c) Epoch-8 BLEU (d) Epoch-8 ASR-task WER



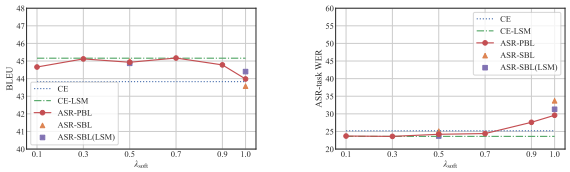
(e) Epoch-10 BLEU (f) Epoch-10 ASR-task WER



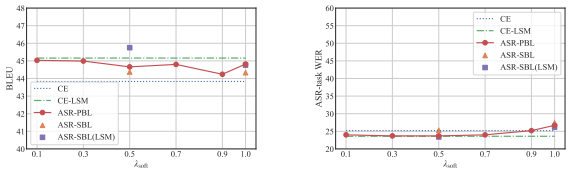
(g) Attn-Specaug BLEU (h) Attn-Specaug ASR-task WER



(i) Attn BLEU (j) Attn ASR-task WER



(k) Attn-CTC BLEU (l) Attn-CTC ASR-task WER



(m) Attn-CTC-Specaug BLEU (n) Attn-CTC-Specaug ASR-task WER

図 3 Fisher (Fisher test) の BLEU と ASR-task WER.

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