End-to-end 音声認識のための 部分文字リカレントニューラルネットワークに基づく仮説修正*

○太刀岡 勇気 (デンソーアイティーラボラトリ)

1 はじめに

従来の複数のモデルを組み合わせて認識を行うハイブリッド手法に代わって、様々な end-to-end(E2E) 型システムが提案されている [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]。E2E システムの利点は簡単なモデリング、速いデコードと辞書が必要とないことである。これは音響特徴量からシンボルへ直接的に変換する。

文字単位あるいは単語単位の E2E システムはより単純で、言語モデルを使った付加的なデコーディングを必要としない [1, 2]。しかしながら、データのスパース性により、E2E は学習データに存在しない語彙外 (OOV) 単語に弱い。文字単位の E2E システムでは OOV を避けることはできるが、言語の制約が単語単位のものに比べて弱いため、ノイズによって誤りやすく、スペル誤りのような誤りが頻発する。

自然言語処理 (NLP) の分野では、神経機械翻訳 (NMT) が広く使われている [12]。音声認識の仮説を入力、正解ラベルを出力として、NMT モデルを文対文の変換として学習すればよい。これでは、単語単位を全体的に変えてしまうが、多くの音声認識誤りは局所的なので、必ずしももこの手法は適切でない。一方、スペル誤りの修正に特化して、同じく NLP の分野で、部分文字リカレントニューラルネットワーク (scRNN) が提案された [13]。これは入れ替え誤りに焦点を当て、単語単位で局所的な誤りを修正できる。

scRNN は単語対単語の変換のため置換誤りしか扱えないが、音声認識の仮説修正に使うためには、1 対 1 の単語対応が得られない挿入・削除誤りに対応する必要がある。scRNN を拡張してこの問題に対処するため、ここでは空白単語記号と単語結合を導入する。

2 節では今回ベースラインに採用したシステムについて概説する。3 節で部分文字列リカレントニューラルネットワークについて紹介したのち、音声認識に適用するために必要な拡張について 4 節で提案する。5 節では騒音下音声認識タスクと大語彙連続音声認識タスクにより提案法の有効性を検証する。

2 End-to-end 音声認識

ここでは、最新の文字単位の E2E システムをベースラインとして採用する。E2E システムは connectionist temporal classification (CTC) [1, 2] と注意機構 (ATT) [3] の 2 種類に分類できる。図 1 の下 2 段は、共有されたエンコーダ・デコーダである。ここで使った E2E システムは CTC と ATT の両方を結合して使う [10]。

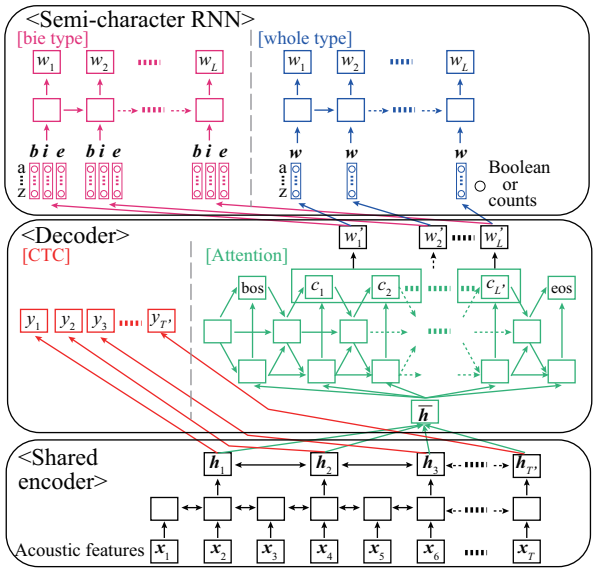

Fig. 1 Proposed end-to-end system with scRNN based correction on top of ASR system.

エンコーダは入力音響特徴量 $\boldsymbol{x}_{1:T}$ を受け取り、T' 長の内部表現 $\boldsymbol{h}_{1:T'}$ に変換する。間引きの影響で、T' は必ずしも T とは一致しない。CTC の出力は、T' 長のラベル系列 $y_{1:T'}$ である。CTC は、文字長とラベル長 T' の差異を補完するため、空白記号やラベル中の繰り返し文字を許容する。文字系列 $\boldsymbol{c}$ に対する CTC の出力確率は

$$P_{CTC}(\boldsymbol{c}|\boldsymbol{x}) = \sum_{\boldsymbol{y}\in\Psi(\boldsymbol{c})} \prod_{t'=1}^{T'} \sigma_{\boldsymbol{h}_{t'}(\boldsymbol{x})}(y_{t'}), \qquad (1)$$

のように、ラベル間 y で独立であると仮定される。ここで、Ψ は、文字系列 $\boldsymbol{c}$ を実現する T' 長の全ラベル系列 $\boldsymbol{y} = y_1, ..., y_{t'}, ..., y_{T'}$ の集合である。$\sigma_{\boldsymbol{h}_{t'}(\boldsymbol{x})}$ は、エンコーダの出力 $\boldsymbol{h}_{t'}(\boldsymbol{x})$ で条件付けされた、ラベル $y_{t'}$ に対するソフトマックス出力である。最終的に、CTC は空白記号と繰り返し文字を削除したのち、文字列を出力する。

ATT の出力は、L' 長の文字系列 $c_1, ..., c_{L'}$ であり、文の始端記号 (bos) と文の終端記号 (eos) を含む。ATT

*Hypothesis Correction Based on Semi-character Recurrent Neural Network for End-to-end Speech Recognition. by TACHIOKA, Yuuki (Denso IT Laboratory)

の出力確率は、$c_0 = \text{bos}$ から始めて、

$$P_{ATT}(\boldsymbol{c}|\boldsymbol{x}) = \prod_{l'=1}^{L'} P(c_{l'}|\bar{\boldsymbol{h}}(\boldsymbol{x}), c_{0:(l'-1)}), \quad (2)$$

のように再帰形で表現される。ここで、$\bar{\boldsymbol{h}}$ は、エンコーダ出力 $\boldsymbol{h}_1, ..., \boldsymbol{h}_{T'}$ を束ねたものである。

これらの出力文字は空白で分離されているので、L 長の単語系列 $w'_1, ..., w'_L$ に変換できる。このシステムでは、明示的な言語制約を使っていないので、出力単語は必ずしも言語的に正しくない。実際、E2E システムの性能は、よく単語誤り率 (word error rate; WER) ではなく、文字誤り率 (character error rate; CER) の観点で評価される。

CTC と ATT 両者の利点を活かすため、これらの統合が提案されている [10]。参照文字系列 $\boldsymbol{c}^*$ に対するマルチタスク損失

$$\mathcal{L} = -\lambda \ln P_{CTC}(\boldsymbol{c}^*|\boldsymbol{x}) - (1-\lambda) \ln P_{ATT}(\boldsymbol{c}^*|\boldsymbol{x}) \quad (3)$$

を最小化する。CTC モデルの学習を補助タスクとして使うことで、CTC による整列がマルチタスク学習の収束を速めることができる。

3 部分文字リカレントニューラルネットワーク (scRNN)

NLP の分野では、スペル誤りを修正するため、scRNN が主にジャンブル誤りに焦点を当てて提案された [13]。ジャンブル誤りは、単語の始めと終わりの文字が一定で、中間の文字のみが置換誤りを起こしているような誤りである。例えば、「characters」が「chraatcres」のような誤りである。これをモデル化するため、文字をはじめ、終わり、中間の 3 種に分けてそれぞれ別々に扱う。

図 1 の上段は scRNN を示す。これは音声認識の仮説の単語列 $w'_{1:L}$ から変換された単語中の文字数を受け取り、修正された単語系列 $w_1, ..., w_L$ を出力する。入力ベクトルの次元はアルファベットと記号を含む文字数の 3 倍である。入力 $\boldsymbol{b}$ と $\boldsymbol{e}$ は、ワンホットベクトルであり、$\boldsymbol{i}$ は疎なベクトルである。例えば、入力単語が「speech」の場合、それぞれのベクトルは $\boldsymbol{b} = \{s = 1\}$、$\boldsymbol{i} = \{c = 1, e = 2, p = 1\}$、$\boldsymbol{e} = \{h = 1\}$ のようになる。scRNN への入力はこれらの結合ベクトル $[\boldsymbol{b}; \boldsymbol{i}; \boldsymbol{e}]$ である。

3 種類別々にモデリングした場合に加えて、単語全体の文字数をモデリングした場合も比較する。例えば、上記の例では、入力ベクトルを $\boldsymbol{w} = \{c = 1, e = 2, h = 1, p = 1, s = 1\}$ のようにする。

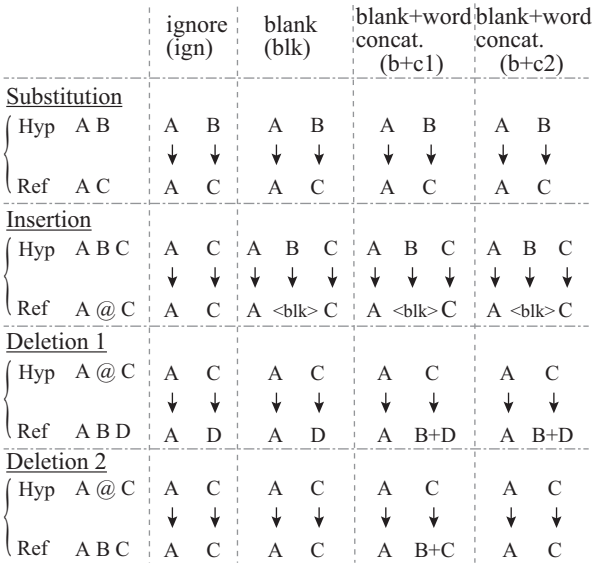

Fig. 2 Four types of blank word symbols (blk) and word concatenation in training stage, where hypotheses (Hyp) and references (Ref) are aligned and @ is null symbol. Input and output to scRNN are Hyp and Ref, respectively.

4 音声認識の仮説修正のための空白単語記号と単語結合の導入

図 2 では、置換・挿入と 2 種類の脱落の 4 種の誤りを示している。図 2 (substitution) の置換誤りに関しては、scRNN が直接的に使える。入力は音声認識仮説であり、出力は正解である。挿入と脱落誤りに関しては、scRNN では 1 対 1 の単語対応が必要なため、仮説と正解の間の単語長の差異を吸収する必要がある。最も簡単な対処法は挿入と削除誤りを無視する (ign) ことだが、単語の文脈を中断してしまう。

そこで、CTC における空白記号に似た空白単語記号 (blk) を導入する。挿入・削除誤りにおいて 1 対 1 で単語対応させるためにこの記号を使う。図 2 (insertion) の挿入誤りは削除誤りよりも扱いやすい。正解における空白単語記号を、音声認識仮説中の入力単語と関連付ける。図中ブランク (blk) は、挿入誤りのみに対応しており、削除誤りは無視している。

削除誤りに対しては、空白単語記号はつかえない。テスト時に仮説中に削除誤りが起こっていることを検出することは難しいためである。正解テキストと音声認識仮説の間の 1 対 1 の単語対応を取るため、複数の単語を結合し、それらを 1 単語として扱う単語結合を行う。ここでは、2 種類の単語結合「b+c1」と「b+c2」を提案する。図 2 (deletion 1) では、削除の直後の単語が仮説と正解では一致していない。仮説中の “C” を、正解文中の “B D” に関連付けられる。これは「b+c1」「b+c2」の両者に共通している。文

中に多量の削除誤りがある場合、結合単語の長さが長くなり、結合単語がほぼコーパス中に現れないことから性能が低下する。本報では、結合単語長の最大値を2とした。図2 (deletion 2) では、削除の直後の単語が一致している。“B” は脱落している可能性があり、認識しがたい。ゆえに、“B” を無視することがよりよい選択肢になる可能性がある。これが「b+c2」である。この手法は文字単位の E2E と結合した単語単位の E2E[14] にも適応しうる。別の見方では、本手法により音声認識誤りを修正することができるので、識別学習的な効果を持つともいうことができる [15]。

5 End-to-end 音声認識実験

5.1 実験条件

2つのコーパスで提案法の有効性を検証した。1つめのコーパスは、第4回 CHiME チャレンジ (CHiME 4) の 1ch トラックである [16]。E2E システムは騒音があるデータに影響を受けやすいので、騒音下音声認識において提案法はより効果的であると考えられる。2つめのコーパスは、TED-LIUM コーパスであり、これは大語彙連続音声認識タスクである [17]。

espnet[1]を利用して、音声認識仮説を得た [10, 11]。両コーパスに対して、付属のスクリプトを利用した。この E2E システムはいかなる言語モデルや辞書も利用していない[2]。位置に基づく ATT を利用し、ユニット数を 320、λ を 0.5 とした。デコーディング時のビームサイズは 20 とした。共有されたデコーダーでは、上2層は長期短期記憶 (LSTM) とし、下層の出力を2フレームに1回入力した (すなわち $T' = T/4$)。

scRNN の語彙は、学習セットに現れた単語で構築したため、開発・評価セットには、OOV がある条件である。特別な OOV 記号 (unk) を scRNN に追加した。scRNN は学習データにより、ミニバッチサイズ 256 で、650 ユニットの LSTM モデルをドロップアウト率 0.01 で学習した。エポック数は 15 とし、Adam [18] を使って、学習率を調整した。ここでは、[13] の著者らにより公開されているスクリプト[3]を修正して用いた。入力文字タイプの次元は 50 であり、アルファベット (a–z) と記号 (ハイフン、コンマ、ピリオド等) から構成される。3 つの異なるタイプの文字数 (bie) を入力とする scRNN と単語全体の数 (w) を入力にする scRNN を比較した。図2のように、空白単語記号と単語結合を 4 種の方法で導入し、“unk” と空白単語記号は削除してから評価した。

Table 1 WER[%] on CHiME 4 challenge. Inputs were whole count (w) or separate count (bie). Input vector type was Boolean (b) or counts (c). scRNN has three types of inputs and outputs in Fig. 2. CER of baseline on development set was 33.5% (dev,real), 33.7% (dev,simu), 44.1% (eva,real), and 41.7% (eva,simu).

	dev set		eva set	
Env.	real	simu	real	simu
baseline	64.7	64.0	78.1	75.2
ign(w)	63.3	62.5	77.2	74.1
ign(bie)	62.8	62.3	76.7	73.6
blk(w)	62.6	61.8	76.2	73.0
blk(bie)	**62.0**	**61.3**	**75.3**	**72.4**
b+c1(bie)	62.4	61.8	76.2	73.2
b+c2(bie)	62.2	61.6	76.0	72.9
NMT(acc)	96.5	96.3	99.3	99.4
NMT(bleu)	97.2	97.2	100.0	100.0

これに加えて、seq2seq[4]を使った NMT[19] と提案法を比較した。このモデルはあたかも音声認識仮説を原言語、正解文章を目的言語とする翻訳のように学習する。NMT は正解率基準 (acc) と bleu [20] 基準 (bleu) の 2 つの基準により学習した。

5.2 CHiME 4 チャレンジ (1ch track)

表1は、CHiME 4 に対する、単語誤り率 (E)[%] である。この場合、NMT が最も性能が低い。正解率基準のほうが bleu 基準よりも若干良好であった。NMT は元の意味を保持していないようなありそうな文章を創り出してしまう。scRNN (表**??**の 2–5 段目) は、ベースラインよりも良い性能を示している。空白単語記号を導入した scRNN(blk) は、削除・挿入誤りを無視した scRNN(ign) を上回る性能を示した。単語結合 (b+c1) が最も良い単語正解率を得たが、全体的には blk が最良の WER を得た。残念ながら b+c2 は blk に劣った。相対誤り低減率は、挿入誤りに対して 15.7%、置換誤りに対して 9.3%、単語誤り率に対して 4.4%であった。

評価セットに対しても傾向は同様である。NMT が最も悪く、scRNN は効果的で、空白単語記号の導入は有効である。blk は相対値で単語誤り率を 3.7–3.9%改善した。

[1] https://github.com/espnet/espnet から利用可能
[2] 詳細な設定は [10] を参照されたい。
[3] https://github.com/keisks/robsut-wrod-reocginiton から利用可能
[4] https://github.com/google/seq2seq から利用可能

Table 2 WER [%] on TED-LIUM development (dev) and test set. CER of baseline was 12.8% (dev) and 12.4% (test).

Set	dev	test
baseline	25.8	24.5
ign(bie)	25.7	24.3
blk(w)	26.4	25.2
blk(bie)	**25.4**	**24.2**
b+c1(bie)	25.5	24.4
b+c2(bie)	25.6	24.4
NMT(acc)	77.0	70.4
NMT(bleu)	77.9	71.4

5.3 TED-LIUM

表 2 は、TED-LIUM コーパスに対する同種の評価を示している。NMT の性能はベースラインに比べてとても悪い。

それ以外の傾向は、CHiME コーパスで観測されたものと類似である。scRNN は WER を相対値で 1.2–1.5%改善した。TED-LIUM は CHiME コーパスに比べて単語数が多く、scRNN の OOV 単語数が多いため、scRNN にとってより難しいタスクであるが、依然として scRNN による仮説修正は効果的である。

6 まとめ

E2E 音声認識システムは騒音の影響を受けやすい。とりわけ、文字単位の E2E は明示的な言語制約を使っていないため、スペル誤りのような誤りを出力することがある。これらの誤りを修正するため、スペル誤りの修正を目的とした scRNN を音声認識の問題に適用した。scRNN は置換誤りにのみ焦点を当てているため、直接的な適用は困難である。削除・挿入誤りを扱うため、空白単語記号と単語結合を導入した。2 種の音声認識タスクの実験により、両タスクに対して、我々の拡張を入れた scRNN はベースラインの性能を改善した。とりわけ、騒音下音声認識タスクにおいて、提案の scRNN は WER を相対値で 4%改善した。

参考文献

[1] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” Proceedings of the 31st International Conference on Machine Learning, pp.1764–1772 (2014).

[2] Y. Miao, M. Gowayyed, and F. Metze, “EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” Proceedings of ASRU, pp.167–174 (2015).

[3] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” Proceedings of NIPS, pp.577–585 (2015).

[4] A. Senior, H. Sak, F. de Chaumont Quitry, T. Sainath, and K. Rao, “Acoustic modeling with CD-CTC-SMBR LSTM RNNs,” Proceedings of ASRU, pp.604–609 2015.

[5] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” Proceedings of ICASSP, pp.4945–4949 (2016).

[6] L. Lu, X. Zhang, and S. Renals, “On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition,” Proceedings of ICASSP, pp.5060–5064 (2016).

[7] R. Prabhavalkar, T. Sainath, B. Li, K. Rao, and N. Jaitly, “An analysis of “attention” in sequence-to-sequence models,” Proceedings of INTERSPEECH, pp.3702–3706 (2017).

[8] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, “Direct acoustics-to-word models for English conversational speech recognition,” Proceedings of INTERSPEECH, pp.959–963 (2017).

[9] H. Soltau, H. Liao, and H. Sak, “Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition,” Proceedings of INTERSPEECH, pp.3707–3711 (2017).

[10] S. Kim, T. Hori, and S. Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” Proceedings of ICASSP, pp.4835–4839 (2017).

[11] S. Watanabe, T. Hori, S. Kim, J.R. Hershey, and T. Hayashi, “Hybrid CTC/attention architecture for end-to-end speech recognition,” IEEE Journal of Selected Topics in Signal Processing, **11**, 1240–1253 (2017).

[12] M.-T. Luong, H. Pham, and C.D. Manning, “Effective approaches to attention-based neural machine translation,” Proceedings of EMNLP, pp.1412–1421 (2015).

[13] K. Sakaguchi, K. Duh, M. Post, and B. Van Durme, “Robsut wrod reocginiton via semi-character recurrent neural network (authors intentionally jumbled the title),” Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp.3281–3287 (2017).

[14] J. Li, G. Ye, R. Zhao, J. Droppo, and Y. Gong, “Acoustic-to-word model without OOV,” Proceedings of ASRU, pp.111–117 (2017).

[15] Y. Tachioka and S. Watanabe, “Discriminative method for recurrent neural network language models,” Proceedings of ICASSP, pp.5386–5390 (2015).

[16] E. Vincent, S. Watanabe, A.A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” Computer Speech and Language, **46**, 535–557 (2016).

[17] A. Rousseau, P. Deléglise, and Y. Estève, “Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks,” Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14) (2014).

[18] D. Kingma and L. Ba, “Adam: A method for stochastic optimization,” Proceedings of ICLR (2015).

[19] D. Britz, A. Goldie, T. Luong, and Q. Le, “Massive exploration of neural machine translation architectures,” Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp.1442–1451 (2017).

[20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” Proceedings of the 40th Annual Meeting of Association for Computational Linguistics (ACL), pp.311–318 (2002).