Abstract

End-to-end (E2E) systems for automatic speech recognition (ASR), such as RNN Transducer (RNN-T) and Listen-Attend-Spell (LAS) blend the individual components of a traditional hybrid ASR system - acoustic model, language model, pronunciation model - into a single neural network. While this has some nice advantages, it limits the system to be trained using only paired audio and text. Because of this, E2E models tend to have difficulties with correctly recognizing rare words that are not frequently seen during training, such as entity names. In this paper, we propose modifications to the RNN-T model that allow the model to utilize additional metadata text with the objective of improving performance on these named entity words. We evaluate our approach on an in-house dataset sampled from de-identified public social media videos, which represent an open domain ASR task. By using an attention model to leverage the contextual metadata that accompanies a video, we observe a relative improvement of about 12% in Word Error Rate on Named Entities (WER-NE) for videos with related metadata.

Index Terms: RNN-T, Deep Contextualization, Context biasing, E2E ASR.

1. Introduction

Present day ASR models using Deep Neural Networks (DNN) can be broadly classified into two frameworks: hybrid [1] and E2E [2][3][4]. A typical hybrid HMM-DNN system consists of three components trained individually: an acoustic model (AM) that estimates the posterior probabilities of Hidden Markov Model (HMM) states, a language model (LM) that estimates probabilities of word sequences, and a pronunciation model (PM) to map phonemes to words. These models are optimized independently [5] and then combined together using a Weighted Orthogonal Transducer (OTF) rescoring [10, 11, 12] or as an additional input to the DNN along with the audio. The first approach is generally referred to as Shallow Fusion whereas the latter as Deep Contextualization [4]. Our work falls in the latter category. It is most closely related to Contextual Listen, Attend And Spell (CLAS) [7], which also used context words from unpaired text to bias an E2E ASR model. The CLAS model was originally evaluated for closed domain ASR tasks like those used for virtual assistants by using entities such as contact names as context words. Further improvements to CLAS were done in [9] and [8] by using representations that leverage phonetic information as well. In this work, different from CLAS, we look at Deep Contextualization [7].

2. Prior Work

Prior work has leveraged contextual words either by on-the-fly (OTF) rescoring [10] [11] [12] or as an additional input to the DNN along with the audio. The first approach is generally referred to as Shallow Fusion whereas the latter as Deep Contextualization [4]. Our work falls in the latter category. It is most closely related to Contextual Listen, Attend And Spell (CLAS) [7], which also used context words from unpaired text to bias an E2E ASR model. The CLAS model was originally evaluated for closed domain ASR tasks like those used for virtual assistants by using entities such as contact names as context words. Further improvements to CLAS were done in [9] and [8] by using representations that leverage phonetic information as well. In this work, different from CLAS, we look at Deep Contextualization in the setting of an RNN-T ASR model, and evaluate our method on an open domain video ASR task using noisy text metadata from videos as context. In a closed domain use case such as making calls through an assistant, there is strong prior information about where entity names can appear in the utterance, whereas in our case the context words may appear...
anywhere in the conversational speech of the video. Deep contextualization of RNN-T was explored in [13] for keyword spotting use case, where the phrase sequence of the keyword represented as a one-hot vector was used to attend to and recognize the target keyword. An alternate approach for using contextual metadata from videos to improve ASR is explored in [14], where lattices produced by a hybrid ASR system are rescored using metadata.

3. RNN Transducer

The framework of RNN-T ASR system is illustrated in Fig. 1. RNN-T for ASR has three main components: Audio Encoder, Text Predictor and Joiner. The Audio Encoder uses audio frame at $x_t$ to produce audio embedding $h_t^{enc}$ (Equation 1). The Audio Encoder used in this work is a stack of bi-directional LSTM (BLSTM) layers.

$$h_t^{enc} = f^{enc}(x_t)$$  (1)

The Text Predictor uses the last non-blank target unit $y_{t-1}$ to produce embedding $h_u^{pred}$ (Equation 2). The Text Predictor is a stack of LSTM layers in this work. We use sentence pieces as target units.

$$h_u^{pred} = f^{pred}(y_{t-1})$$  (2)

The Joiner takes the output of Audio Encoder and Text Predictor and combines them to produce an embedding $z^{t,u}$:

$$z^{t,u} = \phi(Uh_t^{enc} + Vh_u^{pred} + b)$$  (3)

$U$ and $V$ are matrices that are used to project audio and text embeddings to the same dimensions. $\phi$ is a non-linear function such as ReLU [15] or tanh.

Finally, the joiner’s output, $z^{t,u}$, is passed through a linear transformation followed by a softmax layer to produce a probability distribution over target units ($y$), i.e. sentence pieces plus a special blank symbol:

$$h^{t,u} = W^{t,u}z^{t,u} + b$$  (4a)

$$p(y|t,u) = softmax(h^{t,u})$$  (4b)

By incorporating both audio and text for producing $p(y|t,u)$ (Equation 4b), RNN-T can overcome the conditional independence assumption of CTC models [10]. The emission of blank as output unit results in an update of the audio embedding by moving ahead in time axis $t$ whereas emission of non blank results in a change in the text embedding. This results in various possible alignment paths as shown in the lattice of size $T \times U$ in Figure 1 of [2]. The sum of probabilities of these paths gives the probability of an output sequence, $Y$, given the input sequence, $X$, where $Y$ is the sequence of non blank output target units and $X$ is the input sequence of audio frames.

4. Contextual RNN-T

We modify the base RNN-T model described in Section 3 and add two additional components: an Embedding Extractor (EE) and an Attention Module (AttModule) as shown in Figure 2.

As in [7], each context word, $w_i$, is first represented as a sequence of target sentencepiece units, e.g. the word “Jarred” may be mapped to [Ja, r, re, d]. This sequence is then fed to an BLSTM, and the last state of the BLSTM is used as the embedding of the given context word (shown as $h^{EE}_t$ in Figure 2). In the vanilla RNN-T system described in Section 3, probabilities over target units $p(y|t,u)$ (Equation 4b) are conditionally dependent on the outputs of the Audio Encoder, $h_t^{enc}$, and Text Predictor, $h_u^{pred}$. In contextual RNN-T, we would like to make $p(y|t,u)$ conditionally dependent on contextual metadata words as well. This dependency can be achieved by incorporating the context word embeddings, $h^{EE}_t$, into any of the Audio Encoder, Text Predictor and Joiner components.

In this work, we explore incorporating the context word embeddings into the Text Predictor of the RNN-T. An Attention Module (AttModule) is used to compute attention for each word in the metadata text. AttModule uses the predictor output for non-blank text history up to $u$ ($h_u^{pred}$) and word embedding, $h_t^{EE}$, to compute attention weight, $e_{u,i}$, as shown in Equation (5b). We use location-aware attention that takes into account the attention weights from the previous predictor state, $\alpha_{u-1}$, while computing alignments at the current step [4].

$$F = Q \ast \alpha_{u-1}$$  (5a)

$$e_{u,i} = w^t\tanh(Ah_u^{pred} + Bh_t^{EE} + Cf_i + b)$$  (5b)
The input to the network consists of globally normalized 80-dimensional log Mel filterbank features, extracted with 25ms FFT windows and 10ms frame shifts. Sentence piece encoding of each word, \( w_i \), in the metadata is appended with a special sentence piece unit. We use the Adam optimizer [18], with a learning rate of 0.0004, and SpecAugment [19] with policy LB during training. Both RNN-T models were trained for 30 epochs. A beam size of 10 was used during inference.

5.3. Impact on WER-NE and WER

We measure performance of our models using WER and WER-NE on the two test sets described in Section 5.1. An in-house Entity tagger was used to tag named entities in transcripts and metadata.

As seen in Table 1, the Contextual RNN-T model (row 2) improves on WER-NE by about 12% relative compared to the baseline model (row 1) on the CommonZero evaluation set. As shown in Table 2 both WER and WER-NE for the CommonZero test set does not get significantly impacted by the Contextual RNN-T model when there is no intersection between the metadata words and the reference.

Table 1: WER and WER-NE results on CommonZero test set

| Model                  | WER  | WER-NE |
|------------------------|------|--------|
| Baseline               | 14.69| 21.91  |
| Contextual RNN-T       | 14.22| 19.19  |

Table 2: WER and WER-NE results on CommonZero test set

| Model                  | WER  | WER-NE |
|------------------------|------|--------|
| Baseline               | 22.53| 26.68  |
| Contextual RNN-T       | 22.40| 26.95  |

We also measure robustness of our system using precision and recall of the emission of context words in the model’s hypotheses. A True Positive occurs when a context word from the metadata of the video is correctly output by the model as compared to the reference. A False Positive occurs if the model outputs a context word but it does not appear in the reference. We show aggregated precision and recall over both test sets for triggering of the context words in Table 3. We see an improvement in recall by 4.5% and degradation in precision by 1.2% for the Contextual model compared to the baseline.

Table 3: Precision and Recall for context words across both test sets

| Model                  | Precision | Recall |
|------------------------|-----------|--------|
| Baseline               | 0.916     | 0.856  |
| Contextual RNN-T       | 0.905     | 0.895  |

6. Analysis

To understand better what the Contextual RNN-T model is doing, we visualize attention values for a few test segments where it correctly recognizes named entities that the baseline model makes errors on. These examples are shown in Table 4.

For the example shown in row 1 of Table 4 both the Contextual and baseline models are able to recognize common entities such as Africa. However, the baseline model has difficulties in recognizing entities that are not frequent in training data set, such as Android and PyTorch. Since Android appears in the metadata, the Contextual RNN-T model is able to attend to it.
Table 4: Comparing outputs generated by the baseline and Contextual RNN-T models. Named entities are represented with bold font in these examples.

| Reference Snippet                                      | Baseline Output             | Contextualization Output | Metadata Words (truncated)                  |
|--------------------------------------------------------|-----------------------------|--------------------------|--------------------------------------------|
| from the **Africa Android** Challenge                   | from the **Africa and red** challenge | from the **Africa Android** challenge | Plus, app, tutorial innovative, School, language first, session, Project products, educational, startup Android, problem .. |
| its very intuitive so when you look at **PyTorch** itself | its very intuitive so when you look at **pie towards** itself | its very intuitive so when you look at **PyTorch** itself | experiences, novel, PyTorch path, machine, large updates, Facebook, resources AI, language, research .. |

Figure 3: Visualizing attention weights, $\alpha_{u,n}$ from Equation (5c), for the example in Table 4, row 1. The $x$-axis shows the target units from the hypothesis output by the Contextual RNN-T Model, and the $y$-axis shows the contextual metadata words ($w_i$). Darker colors represent values close to zero while brighter colors represent values closer to 1.

7. Conclusion

In this work, we show that contextual metadata text, even if it is noisy, can be used to improve recognition of named entities for a challenging open domain ASR task such as social media videos within the framework of an E2E RNN-T ASR model. Some directions to explore further as future work could be: i) Using contextual embeddings from other modalities such as images from video, ii) An in-depth study of the impact of augmenting contextual information in the Audio Encoder, Text Predictor and Joiner for different modalities iii) Exploring different architectures for the EmbeddingExtractor(EE), iv) Using semantic embeddings to represent the metadata.

8. Acknowledgement

The authors would like to thank Anuroop Srinivasan, Duc Le, Florian Metze and Geoffrey Zweig for many helpful discussions and suggestions.

9. References

[1] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 2012, vol. 247.

[2] A. Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[4] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” *CoRR*, vol. abs/1506.07503, 2015. [Online]. Available: [http://arxiv.org/abs/1506.07503](http://arxiv.org/abs/1506.07503)

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury et al., “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal processing magazine*, vol. 29, 2012.

[6] M. Mohri, F. Pereira, and M. Riley, “Weighted finite-state transducers in speech recognition,” *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[7] G. Pandek, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, “Deep context: end-to-end contextual speech recognition,” *arXiv preprint arXiv:1808.02480*, 2018.
[8] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer, and C. Fuegen, “Joint grapheme and phoneme embeddings for contextual end-to-end asr,” in INTERSPEECH, 2019.

[9] A. Bruguier, R. Prabhavalkar, G. Pandak, and T. N. Sainath, “Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition,” 2019.

[10] D. Zhao, T. N. Sainath, D. Rybach, D. Bhatia, B. Li, and R. Pang, “Shallow-fusion end-to-end contextual biasing,” Submitted to INTERSPEECH, vol. 2019, 2019.

[11] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, “Contextual speech recognition in end-to-end neural network systems using beam search.” in Interspeech, 2018, pp. 2227–2231.

[12] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, et al., “Streaming end-to-end speech recognition for mobile devices,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 6381–6385.

[13] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw, “Streaming small-footprint keyword spotting using sequence-to-sequence models,” in 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017, pp. 474–481.

[14] D.-R. Liu, C. Liu, F. Zhang, G. Synnaeve, Y. Saraf, and G. Zweig, “Contextualizing asr lattice rescoring with hybrid pointer network language model,” in Submitted to Proc. Interspeech, 2020.

[15] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[16] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369–376.

[17] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” arXiv preprint arXiv:1808.06226, 2018.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[19] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugsment: A simple data augmentation method for automatic speech recognition,” arXiv preprint arXiv:1904.08779, 2019.