END-TO-END REAL TIME TRACKING OF CHILDREN'S READING WITH POINTER NETWORK

Vishal Sunder, Beulah Karrolla, Eric Fosler-Lussier

The Ohio State University

ABSTRACT

In this work, we explore how a real time reading tracker can be built efficiently for children's voices. While previously proposed reading trackers focused on ASR-based cascaded approaches, we propose a fully end-to-end model making it less prone to lags in voice tracking. We employ a pointer network that directly learns to predict positions in the ground truth text conditioned on the streaming speech. To train this pointer network, we generate ground truth training signals by using forced alignment between the read speech and the text being read on the training set. Exploring different forced alignment models, we find a neural attention based model is at least as close in alignment accuracy to the Montreal Forced Aligner, but surprisingly is a better training signal for the pointer network. Our results are reported on one adult speech data (TIMIT) and two children's speech datasets (CMU Kids and Reading Races). Our best model can accurately track adult speech with 87.8% accuracy and the much harder and disfluent children's speech with 77.1% accuracy on CMU Kids data and a 65.3% accuracy on the Reading Races dataset.

***Index Terms*—** Speech tracking, End-to-End models, Reading assessment

1. INTRODUCTION

Tracking read speech finds useful applications in education, when teaching children how to read properly. Building reading tutors has been a popular application of automatic speech recognition (ASR) [1, 2, 3] and tracking is an important part of that. In contrast to offline assessment to score pronunciations and give offline feedback [4, 5, 6], a tracker needs to function in real time. An automated tracker can follow along a student as they are reading and when they are stuck at a difficult to pronounce word, it can prompt the word thus aiding the student. However, automated tracking is not without challenges. Children's reading, when they are learning, is especially difficult to track owing to the disfluencies present. There can be a lot of false starts, word repetitions and word skipping involved.

Traditional modeling of a reading tracker has used a cascade of an ASR model and a rule based tracking algorithm [7]. Li et al. [8] further improve this method by taking into account the real time nature of tracking, but their method is also dependant on an ASR model. For a scenario where data is scarce, training an ASR model can be challenging [9]. An example of this is the Reading Races dataset that we experiment with. In such cases, using an off-the-shelf pretrained ASR model can lead to hallucinations. Another problem is the time delay between the occurrence of acoustic evidence and the prediction [10].

In this work, we build a fully end-to-end (E2E) speech tracker using a pointer network [11]. This formulation is completely ASR free and our tracker learns an attention map over the text being read conditioned on the streaming speech. This attention map is learnt explicitly using ground truth alignments that we obtain from a forced aligner. Using this formulation, the time lag between acoustic evidence and the prediction is reduced to a significant extent as we directly predict a pointer position at each time step without having to predict the actual word. Another advantage of this approach is that we can directly get the alignment by reading the attention maps without needing to run a separate alignment algorithm. This way, we avoid the cascading effects of ASR errors.

We experiment with three forced alignment models to generate the ground truth to train the tracker, attention-based encoder-decoder ASR model (AED) [12], a CTC-based ASR model [13] and the classical GMM-HMM based ASR model [14]. We note that the advantage of using the AED model is that we naturally get soft alignments as training targets for the tracker which can be useful for knowledge distillation.

We provide results on one adult and two children speech datasets. For the adult voice dataset, we use TIMIT data [15]. For the children voice dataset, we use the CMU Kids [16] data and the Reading Races dataset [17]. We observe better tracking accuracy on TIMIT with the best tracking accuracy of 87.8%. On CMU Kids and Reading Races, we report the best tracking accuracy of 77.1% and 65.3% respectively. We also provide qualitative results on the children's dataset showing how some disfluencies are is handled by the pointer network.

2. MODEL OVERVIEW

The overall pipeline for building a real time tracker comprises of two steps. In the first step, we generate forced alignments for the training data using an ASR-based model. These alignments are then used as the ground truth supervision for the second

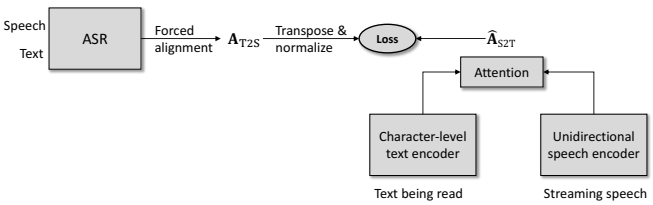

Fig. 1. The ASR is used to generate the alignment between the speech and text, $\mathbf{A}_{T2S} \in \mathbb{R}^{m \times n}$ where m and n are the number of text and speech tokens respectively. The tracker learns an attention over the text encoder output using the streaming speech to predict the alignment $\hat{\mathbf{A}}_{S2T} \in \mathbb{R}^{n \times m}$ which is learnt using $\mathbf{A}_{T2S} \in \mathbb{R}^{m \times n}$ as the supervision.

step which is to train a pointer network based tracker.

2.1. Forced alignment

A forced aligner is an ASR model which predicts the best possible alignment between a text input and the corresponding speech. We denote the alignment generated by the forced aligner as a matrix $\mathbf{A}_{T2S} \in \mathbb{R}^{m \times n}$ where m and n are the number of text and speech tokens respectively. Each row of $\mathbf{A}_{T2S}$ is the alignment of a text token with all the speech frames. Depending on the type of ASR model used, this alignment can be soft (a probability distribution) or hard (a one/multi-hot vector). We explore three ASR architectures for performing forced alignment.

Attention-based encoder decoder (AED): This model is based on the LAS framework [12]. Here, speech is encoded using a bidirectional speech encoder and the decoder is a unidirectional LSTM which decodes the text one character at a time while implicitly learning an alignment between speech and text through the attention layer. The alignment matrix $\mathbf{A}_{T2S}$ is obtained from the attention layer and is a soft alignment, i.e. for each character in the text, we obtain a probability distribution over the sequence of speech frames, denoting the alignment. We can also convert this soft alignment into a hard alignment easily by converting $\mathbf{A}_{T2S}$ into a multi-hot vector based on an alignment-weight threshold.

CTC-based ASR: For this, we train an ASR model with the CTC criterion [13]. Once the model is trained, we follow Kurzinger et al. [18] to obtain the trellis matrix which is the probability of the characters aligned at each time step. Using this trellis, we can estimate the most likely CTC path for the given speech-text pair by backtracking. This gives us the desired alignment, $\mathbf{A}_{T2S}$ which is a hard alignment.

GMM-HMM based ASR: We use the Montreal Forced Aligner (MFA) [14] for this. We train acoustic models using the pronunciation dictionary provided for the domain specific datasets for forced alignment. MFA gives hard alignments by default at the word and phone level. We could also get soft alignments at the phonetic or grapheme level by extracting the γ probabilities in the HMM model. For this work, we limit ourselves to the default hard alignments from MFA.

2.2. Pointer network based tracker

Pointer networks were introduced in Vinyals et al. [11] for tackling various combinatorial problems with deep learning models using an additive attention mechanism. The original formulation of pointer networks is autoregressive, where the decoder points to a certain position in the encoder sequence and this position is then added to the decoder output. Our tracker application does not need the autoregressive formulation as our model is not generative.

Our pointer network consists of a character-level text encoder, a unidirectional LSTM-based speech encoder and an additive attention layer. Let the output of the text encoder be a sequence of character embeddings $(g_1, g_2, ..., g_m)$ and that of the speech encoder be a sequence of speech frames $(h_1, h_2, ..., h_n)$. Given these two sequences, we want to get an alignment estimate, $\hat{\mathbf{A}}_{S2T} \in \mathbb{R}^{n \times m}$ where each row corresponds to a speech frame alignment with all characters in the text. We estimate $\hat{\mathbf{A}}_{S2T}$ using the attention layer as follows,

$$x_j^i = \mathbf{v}^{\mathrm{T}} \tanh(\mathbf{W}_1 g_i + \mathbf{W}_2 h_j)$$
$$\mathbf{a}_j = \text{softmax}(\mathbf{x}_j)$$
$$\hat{\mathbf{A}}_{S2T} = \text{concat}([\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n])$$

Here, $\mathbf{v}$, $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable parameters. The alignment of the j^{th} speech frame with all the characters in the text is denoted by the probability distribution $\mathbf{a}_j$.

At inference, we can compute alignments as we get speech frames, h_j in real time using the unidirectional LSTM.

Training: To train the tracker, we obtain supervision from the alignments, $\mathbf{A}_{T2S}$ generated from the forced aligner. We compute the ground truth, $\mathbf{A}_{S2T}$ as follows,

$$\mathbf{A}_{S2T} = \text{L1-normalize}(\mathbf{A}_{T2S}^{\mathrm{T}})$$

We transpose $\mathbf{A}_{T2S}$ and normalize it row-wise so that we obtain a probability distribution for every speech frame. If $\mathbf{A}_{S2T}$ is a hard alignment, we compute,

$$L_{hard} = \frac{1}{|\mathbb{B}|} \sum_{b \in \mathbb{B}} \frac{1}{N_b} \sum_{i=1}^{N_b} \text{CrossEntropy}(\hat{\mathbf{A}}_{S2T}[i], \mathbf{A}_{S2T}[i])$$

where the cross entropy is computed for the alignment of every speech frame, i.e. every row of the alignment matrices across the batch $\mathbb{B}$. When $\mathbf{A}_{T2S}$ is a soft alignment, we compute,

$$L_{soft} = \frac{1}{|\mathbb{B}|} \sum_{b \in \mathbb{B}} \frac{1}{N_b} \sum_{i=1}^{N_b} \text{KLDivergence}(\hat{\mathbf{A}}_{S2T}[i], \mathbf{A}_{S2T}[i])$$

L_{soft} can also be computed as a cross entropy loss, but we follow previous work by Hinton et al. [19] and use KL-Divergence by treating the soft target as a knowledge distillation target. The model overview is shown in figure 1.

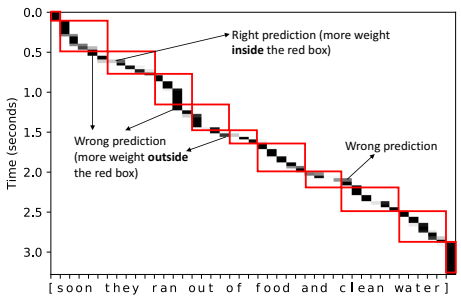

Fig. 2. The red boxes are the ground truth alignments of words against time. The pixels represent the tracker prediction. The height of each pixel is a 40ms frame. For a correct word prediction by a frame, the total weight inside the red box should be greater than that outside. For all frames, we can count the number of correctly/incorrectly predicted words.

3. EXPERIMENTS

We train two deep learning based E2E ASR systems to build the AED and CTC forced aligners. For AED, we follow the design choices of Chan et al. [12]. We use one BiLSTM layer to encode sequences of 80 dimensional log mel-filterbank features, followed by two pyramid BiLSTM layers which downsample the sequence length by a factor of 4. Finally, we add the last BiLSTM layer to produce the final acoustic representations. The decoder is a two layer LSTM and the attention mechanism is content-based and additive.

The CTC model has the same speech encoder architecture as AED. We train the CTC and AED models on Librispeech [20] followed by an adaptation on the downstream datasets.

For the tracker, the text encoder is a two layer BiLSTM and the speech encoder follows the same architecture as the encoder of AED except that it is unidirectional. The tracker is pre-trained on Librispeech and adapted on downstream data.

3.1. Evaluation

Forced alignment: To evaluate the performance of forced aligners, we use precision, recall and jaccard similarity. For a given word, let t_1 and t_2 be the ground truth start and end times and $\hat{t}_1$ and $\hat{t}_2$ be the predicted start and end times respectively. Then, we define the evaluation metrics as follows,

$$\text{intersection} = \max(\min(t_2, \hat{t}_2) - \max(t_1, \hat{t}_1), 0)$$

$$\text{union} = t_2 - t_1 + \hat{t}_2 - \hat{t}_1 - \text{intersection}$$

$$\text{jaccard (\textbf{Ja})} = \frac{\text{intersection}}{\text{union}}$$

$$\text{precision (\textbf{Pr})} = \frac{\text{intersection}}{\hat{t}_2 - \hat{t}_1}$$

$$\text{recall (\textbf{Re})} = \frac{\text{intersection}}{t_2 - t_1}$$

Ground truth	TIMIT Manual			CMUK Sphinx II [21]			READR Manual (10 samples)		
	Pr	Re	Ja	Pr	Re	Ja	Pr	Re	Ja
AED-aligner (ours)	84.57	89.65	76.16	78.87	**82.98**	68.10	**80.91**	**74.94**	**73.15**
CTC-aligner	61.93	69.58	50.29	50.80	58.26	37.49	39.33	61.34	31.45
MFA (flat start)	**91.19**	**90.96**	**83.56**	**84.72**	75.30	**68.48**	21.57	37.00	17.48
MFA (adapted)	48.66	49.48	34.91	71.03	67.51	56.78	16.98	24.17	12.72

Table 1. Forced alignment results on TIMIT, CMUK and READR. The ground truth for TIMIT is manually annotated, for CMUK, we use the provided Sphinx II annotation as ground truth. For READR, we manually annotate 10 random examples from the test set for evaluation.

Tracking: To evaluate the tracking performance of the pointer network, we use the predicted alignment, $\hat{\mathbf{A}}_{S2T}$. Each row of this matrix represents the alignment of a 40ms speech frame with all characters in the text. For each frame, we compute the score for every word in the text by adding the character weights for that word in the corresponding row. The word with the highest score is then predicted. A detailed illustration of this is shown in figure 2. To make the prediction for a speech frame i more deterministic, we make the output distribution sharper using the following operation with $\tau = 0.1$,

$$\texttt{sharp}(\hat{\mathbf{A}}_{S2T}[i]) = \frac{(\hat{\mathbf{A}}_{S2T}[i])^{\frac{1}{\tau}}}{||(\hat{\mathbf{A}}_{S2T}[i])^{\frac{1}{\tau}}||_1}$$

3.2. Datasets

We use three datasets for evaluation.

TIMIT [15]: This is a 5-hour dataset of adult voice recordings. We use the standard train-test split. This data provides time aligned transcriptions which act the ground truth.

CMU Kids (CMUK) [16]: This is a 9-hour corpus of children read speech. The children's age vary between 6 to 11 years. This data provides time aligned transcriptions from Sphinx II [21]. For disfluencies, we are provided with a phoneme level transcription. We convert these into word-level transcription by using dynamic time warping to align orthographic and phonemic transcriptions of words.

Reading Races (READR) [17]: This is a 15-hour corpus of children read speech with each data instance being a minute long. This is a more challenging dataset with participants being in the age group of 5 to 8 years with reading difficulties.

3.3. Results

Forced alignment: The output of forced alignment acts as the training signal for our pointer-network based tracker. We compare forced alignment performance of three ASR models: AED, CTC and GMM-HMM (MFA for montreal forced aligner [14]). The results are shown in table 1.

We note that MFA performs best compared to the other two on TIMIT and CMUK. However, this required training an acoustic model using the provided pronunciation dictionary.

Ground truth (→)	TIMIT Manual		CMUK Sphinx II [21]		READR AED	
Training signal (↓)	Acc	F1	Acc	F1	Acc	F1
AED (L_{hard})	83.30	81.26	73.92	70.82	**65.34**	**67.76**
AED (L_{soft})	87.73	83.81	**77.07**	**77.15**	63.68	63.41
AED ($L_{hard} + L_{soft}$)	**87.82**	**83.85**	77.06	76.13	64.45	67.12
MFA	77.67	72.76	67.11	49.70	7.69	5.84

Table 2. Tracking results. Training signal from 4 different forced aligners are used to train to tracker. The accuracy and F1 scores are measured as in figure 2. For READR, we use the AED forced aligner output as the ground truth.

AED and CTC based forced aligners are fully E2E and do not require a pronunciation dictionary to adapt their acoustic models. The advantage of this is evidenced by the results of MFA on READR which does not have a pronunciation dictionary of its own and thus MFA performs poorly. For READR, we manually time aligned 10 one-minute long random examples from the test set. We note that AED performs much better compared to CTC with an additional advantage that AED provides soft alignments between speech and text which can be used for teacher forcing to train the tracker.

Tracking: Table 2 shows the tracking results. We use the evaluation procedure mentioned in section 3.1 reporting tracking accuracy and F1 score. We report tracking performance with 4 training signals but do not evaluate the CTC based training signal as it's forced alignment performance was not very good.

When using the AED variants for the training signal, $\mathbf{A}_{S2T}$, we perform significantly better compared to the MFA based $\mathbf{A}_{S2T}$ even though MFA gave better forced alignment results on TIMIT and CMUK. This is because MFA gives word-level time alignments while the AED model is trained to give character-level alignments. Correspondingly, the tracker can also be trained at the character level. For a character-level model, even if a character prediction for a frame is out of bounds (outside the red box in figure 2), there is still a chance for recovery through its alignment with other characters in the word (more weight inside the red box in figure 2). There is no such recovery provision for a word-level model.

For the READR data, we note that using the hard training signal (L_{hard}) gives best performance whereas for CMUK and TIMIT, the soft signal (L_{soft}) helps more. We hypothesize that as READR has very long audio inputs (1 minute on average), having a precise alignment as a training signal better facilitates the model in finding the exact location. Also note that the MFA training signal performs poorly on READR due to the MFA forced alignments themselves being very poor in table 1.

For READR, we also evaluate the tracker against the 10 manually aligned examples. We compare this with the performance we got when evaluating against the force aligned ground truth (see table 3). Note that we see better performance when evaluating against the manual annotations which shows that our tracker follows closer with human alignments.

	Manual annotation		AED annotation	
Training signal	Acc	F1	Acc	F1
AED (L_{hard})	**69.34**	**71.99**	64.08	67.25

Table 3. Tracker result on 10 random examples from READR, comparing manual alignments with AED alignments.

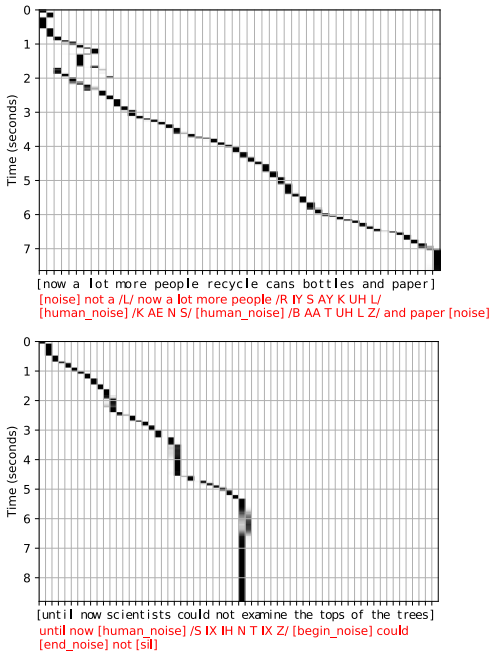

Fig. 3. Pointer network's tracking. X-axis is the sentence being read and the text in red is the transcript. We see that word repetition (top) and skipping (bottom) is effectively captured.

Finally, we show how the tracker behaves in the presence of disfluencies in figure 3. The top plot shows that in the case of repetition or false start, the tracker is able to effectively realign itself and continue the monotonic trajectory. Also, the tracker can detect the stopping point in a partially read sentence and ignore the unread part (bottom of figure 3). Thus, the tracker can give meaningful alignments for disfluent children's speech.

4. CONCLUSION

In this work, we build a real time reading tracker using pointer network. Our proposed method does not require manual annotation and relies on forced alignment to generate the training signal to train the tracker. We explore different forced alignment strategies to generate the training signal and note that AED based forced alignment works best to train the tracker.

5. ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under Grant No. 2008043.

6. REFERENCES

[1] Marilyn Jager Adams, "The promise of automatic speech recognition for fostering literacy growth in children and adults," in *International handbook of literacy and technology*, pp. 109–128. Routledge, 2013.

[2] Jack Mostow, Steven F Roth, Alexander G Hauptmann, and Matthew Kane, "A prototype reading coach that listens," in *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, 1994, pp. 785–792.

[3] Jack Mostow, "Why and how our automated reading tutor listens," in *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training (ISADEPT)*. KTH Stockholm, Sweden, 2012, pp. 43–52.

[4] Matthew Black, Joseph Tepperman, Sungbok Lee, Patti Price, and Shrikanth S Narayanan, "Automatic detection and classification of disfluent reading miscues in young children's speech for the purpose of assessment," in *Interspeech*, 2007.

[5] Minglin Wu, Kun Li, Wai-Kim Leung, and Helen Meng, "Transformer based end-to-end mispronunciation detection and diagnosis.," in *Interspeech*, 2021, pp. 3954–3958.

[6] Lavanya Venkatasubramaniam, Vishal Sunder, and Eric Fosler-Lussier, "End-to-end word-level disfluency detection and classification in children's reading assessment," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[7] Morten Højfeldt Rasmussen, Jack Mostow, Zheng-Hua Tan, Børge Lindberg, and Yuanpeng Li, "Evaluating tracking accuracy of an automatic reading tutor," in *Speech and Language Technology in Education*, 2011.

[8] Yuanpeng Li and Jack Mostow, "Evaluating and improving real-time tracking of children's oral reading.," in *FLAIRS Conference*, 2012.

[9] Lucile Gelin, Morgane Daniel, Julien Pinquier, and Thomas Pellegrini, "End-to-end acoustic modelling for phone recognition of young readers," *Speech Communication*, vol. 134, pp. 71–84, 2021.

[10] Peter Plantinga and Eric Fosler-Lussier, "Towards real-time mispronunciation detection in kids' speech," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 690–696.

[11] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly, "Pointer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[12] William Chan, Navdeep Jaitly, Quoc Le, and Oriol 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.

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[14] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi.," in *Interspeech*, 2017, vol. 2017, pp. 498–502.

[15] John S Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[16] M Eskenazi, J Mostow, and D Graff, "The cmu kids corpus," in *Linguistic Data Consortium, no. 11*. LDC, 1997.

[17] Morris R Council III, Ralph Gardner III, Gwendolyn Cartledge, and Alana O Telesman, "Improving reading within an urban elementary school: computerized intervention and paraprofessional factors," *Preventing School Failure: Alternative Education for Children and Youth*, vol. 63, no. 2, pp. 162–174, 2019.

[18] Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *International Conference on Speech and Computer*. Springer, 2020, pp. 267–278.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[20] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[21] Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee, and Ronald Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech & Language*, vol. 7, no. 2, pp. 137–148, 1993.