SUPER-HUMAN PERFORMANCE IN ONLINE LOW-LATENCY RECOGNITION OF CONVERSATIONAL SPEECH

Thai-Son Nguyen, Sebastian Stüker, Alex Waibel

Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology

ABSTRACT

Achieving super-human performance in recognizing human speech has been a goal for several decades, as researchers have worked on increasingly challenging tasks. In the 1990's it was discovered, that conversational speech between two humans turns out to be considerably more difficult than read speech as hesitations, disfluencies, false starts and sloppy articulation complicate acoustic processing and require robust handling of acoustic, lexical and language context, jointly. Early attempts with statistical models could only reach error rates over 50% and far from human performance (WER of around 5.5%). Neural hybrid models and recent attention-based encoder-decoder models have considerably improved performance as such contexts can now be learned in an integral fashion. However, processing such contexts requires an entire utterance presentation and thus introduces unwanted delays before a recognition result can be output. In this paper, we address performance *as well as* latency. We present results for a system that can achieve super-human performance (at a WER of 5.0%, over the Switchboard conversational benchmark) at a word based latency of only 1 second behind a speaker's speech. The system uses multiple attention-based encoder-decoder networks integrated within a novel low latency incremental inference approach.

Index Terms— ASR, Sequence-to-sequence, Online, Streaming, Low Latency, Human Performance

1. INTRODUCTION

Sequence-to-sequence (S2S) attention-based models [1, 2] are a very promising approach to end-to-end automatic speech recognition (ASR). A lot of research has already been dedicated to boost the performance of S2S models. Several works [3, 4, 5, 6, 7] have successfully pushed up the state-of-the-art performance records on different speech recognition benchmarks and proved the superior performance of S2S models over conventional speech recognition models in an offline setting. As so, the next research trend is to apply S2S speech recognition in practice. In many practial applications, adapting ASR models to work under the online low-latency condition is the most desirable need [8].

Early studies [9, 10, 11] pointed out that the disadvantage of a S2S model used in online condition lies in its attention mechanism, which must perform a pass over the entire input sequence for every output element. [10, 11] have dealt with this disadvantage by proposing a so-called monotonic attention mechanism that enforces a monotonic alignment between the input and output sequence. Later on, [12, 13, 14] have additionally resolved the latency issue of bidirectional encoders by using efficient chunk-based architectures. More recent works [15, 16, 17] have addressed these latency issues for different S2S architectures.

While most of the studies focus on model modifications to make S2S models capable of online processing with minimal accuracy reduction, they lack thoughtful research on the latency aspect. In this work, we analyze the latency that the users suffer while interacting with an online speech recognition system, and propose to measure it with two separate terms *computation latency* and *confidence latency*. While computation latency reflects the common real-time factor (RTF), confidence latency corresponds to the delay an online recognizer needs to confidently decide its output. We show that with the support of new computing hardware (such as GPU), the computation latency of S2S models is relatively small (even for big models), and the confidence latency is a more critical criterion and has not been addressed thoroughly.

Optimizing for the confidence latency, we consider the online processing of S2S models as the incremental speech recognition problem. We propose an *incremental inference* approach with two stability detection methods to turn a S2S model to be used in online speech recognition and allow the possibility to trade-off between latency and accuracy. Our experimental results show that it is possible to use a popular Long Short-Term Memory (LSTM) or self-attention based S2S model in online condition without any model modification. With a delay of 1.8 seconds in all output elements, all the experimental models retain their ideal performance as in offline inference. Our best online system, which successfully employs three S2S models in low-latency manner, achieved a word-error-rate (WER) of 5.0% on the Switchboard benchmark. To the best of our knowledge, this online accuracy is on par with the state-of-the-art offline performance. We also demonstrate that is is possible to achieve human performance as reported in [18, 19] while producing output at very low latency.

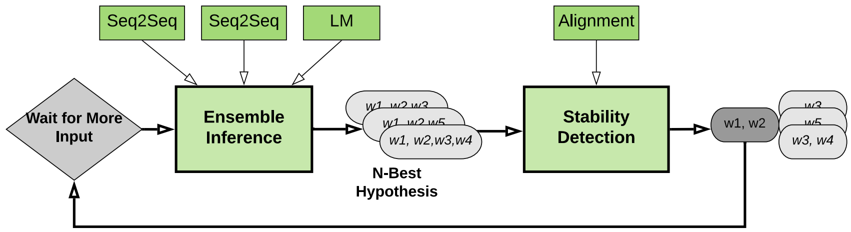

Fig. 1. Incremental inference for low-latency S2S ASR

2. SEQUENCE-TO-SEQUENCE BASED LOW-LATENCY ASR

In this section, we first describe different S2S architectures investigated in the paper. We then present the proposed incremental inference with two stability detection methods.

2.1. Models

To date, there have been two efficient approaches for making S2S ASR systems. The first approach employs LSTM layers in both encoder and decoder networks, while the second follows the Transformer architecture [20] which uses solely self-attention modules to construct the whole S2S network. In this work, we investigate both of the S2S architectures for the online low-latency setting.

Our LSTM-based S2S model employs two convolutional layers with the total time stride of four for down-sampling followed by several bidirectional LSTM layer to encode input spectrogram. In the decoder, we adopt two layers of unidirectional LSTMs for modeling the sequence of sub-word labels and the multi-head *soft-attention* function proposed in [20] to generate attentional context vectors.

In the Transformer model, the down-sampling is handled by a linear projection layer on a frame-stacking of four consecutive feature vectors. We use many stochastic self-attention layers in both encoder and decoder to form a deep architecture as proposed in [6].

For more details of the model architectures and offline evaluations, we would refer the readers to [7] and [6].

2.2. Incremental Inference

Figure 1 illustrates our proposed architecture that allows S2S models to produce incremental transcriptions on a speech stream. In it we handle the two tasks of inference and stability detection by two separate components in a processing pipeline. The first step in the pipeline is to wait for a chunk of acoustic frames with a predefined size (i.e., 200ms), which is then sent to the inference component. The inference component needs to accumulate all the chunks received so far and extend the current *stable* hypothesis to produce a set of new *unstable* hypotheses. This unstable set is then provided to the *stability detection* component for detecting a longer stable hypothesis.

As the stability detection is handled separately, we are able to involve multiple models for the inference to improve recognition accuracy. The involved models can be S2S models with different architectures or language models trained on different text data. All of these models can be uniformly combined via the ensemble technique.

2.3. Stability Detection

Stability detection is the key to make the system work in the incremental manner and to produce low latency output. For an HMM based speech recognition system, stability conditions can be determined incrementally during the time-synchronous Viterbi search [21, 22, 23]. Due to lack of time alignment information and unstable internal hidden states (e.g., of a bidirectional encoder), it is not straightforward to apply the same idea to S2S models. In this work, we investigate a combination of two stability detection conditions for incremental S2S speech recognition:

- **Shared prefix in all hypotheses**: Similar to the *immortal prefix* [21, 23] in HMM ASR, this condition happens when all the active hypotheses in the beam-search share the same prefix. However, different from HMM ASR, this condition may not strongly lead to an *immortal* partial hypothesis due to the unstable search network states in S2S beam-search.

- **Best-ranked prefix with reliable endpoint**: Since it may require a long delay for a *shared prefix* to happen, we also consider a different approach to improve the latency. We make use of the observation from [22] for HMM ASR, that the longer a prefix remains to be part of the most likely hypothesis, the more stable it is. Applied to S2S models, we need a method to align a prefix with audio frames, and so be able to find its endpoint in time. We follow the approach in [17] for the extraction of a prefix endpoint. First, this approach requires to train a single-head attention LSTM-based S2S model with the attention-based constraint loss [17]. Then, the endpoint of a prefix C is determined during incremental inference by finding a time t_c such that the sum of all attention scores from the covering window $[0, t_c]$ is at least 0.95.

3. MEASURE OF LATENCY

Latency is one of the most important factors that decide the usability of an user-based online ASR system. A latency measure needs to reflect the actual delay that the users perceive so that the improvement of latency can lead to better usability. Strictly, the latency observed by a user for a single word is the time difference between when the word was uttered and when its transcript appeared to the user and will never be changed again. We formulate this complete latency as follows.

Let's assume a word w has been uttered, i.e., completely pronounced, at time U_w. Let C_w be the time that the ASR system can start to process the audio of w and that the ASR system can *confidently* infer w after a *delay* of D_w, the time needed to perform the inference. The user-perceived latency with regard to w is then:

$$Latency_w = C_w + D_w + T_w - U_w$$

where T_w presents the transmitting time for audio and text data. T_w is usually small and can be omitted.

For a speech utterance S consisting of N words w_1, w_2,.. w_N, we are interested in the average latency:

$$\begin{aligned} Latency_S &= \sum_i^N (D_{w_i} + C_{w_i} - U_{w_i})/N \\ &= \sum_i^N D_{w_i}/N + \sum_i^N C_{w_i}/N - \sum_i^N U_{w_i}/N \\ &= \sum_i^N D_{w_i}/N + \sum_i^N C_{w_i}/N - \sum_i^N (U_{w_i} - \Delta)/N + \Delta \\ &= D_{avg} + C_{avg} - U_{avg-\Delta} + \Delta \end{aligned}$$

In the final equation, the first term represents the computational delay. If we normalize this term by length of the utterance, then we obtain the real-time factor of the ASR system. The second term indicates how much acoustic evidence the model needs to confidently decide on its output. This latency term makes the difference in calculating the latency for online vs. offline processing. For offline processing, it is always a constant for a specific test set, since all the offline transcripts are output at the end of the test set.

To estimate the third term, we usually need to use an external time alignment system, e.g. a Viterbi alignment using an HMM based acoustic model. It is inconvenient to re-run the time alignment for every new transcript. To cope with this issue, [17] introduced a fixed delay Δ for all the outputs, and proposed to pre-compute a set of $U_{avg-\Delta}$ for different Δ. Later on, only the calculation of C_{avg} is required as the average delay can be found by comparing C_{avg} with the pre-computed set.

The latency improvement requires the optimization of both terms D_{avg} and C_{avg} which we refer to as *computation latency* and *confidence latency*. While computation latency

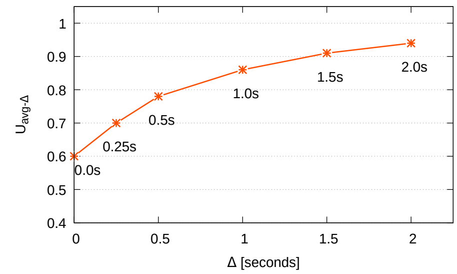

Fig. 2. Confidence latency conversion.

can be improved by faster hardware or improved implementations for the search, confidence latency mainly depends on the recognition model.

4. EXPERIMENTAL SETUP

Our experiments were conducted on the Fisher+Switchboard corpus consisting of 2,000 hours of telephone conversation speech. The Hub5'00 evaluation data was used as the test set, reporting separate performance numbers for the Switchboard (SWB) and CallHome (CH) portions. All our models use the same input features of 40 dimensional log-mel filterbanks to predict 4,000 byte-pair-encoded (BPE) sub-word units. During training, we employ the combination of two data augmentation methods *Dynamic Time Stretching* and *SpecAugment* [7] to reduce model overfitting. Adam with an adaptive learning rate schedule is used to perform 12,000 updates. The model parameters of the 5 best epochs according to the perplexity on the cross-validation set are averaged to produce the final model.

4.1. Latency Evaluation

We evaluate our systems with the decomposed latency terms from Section 3. *Computation latency* is measured every time when incremental inference is performed, while for *confidence latency* we follow the approach in [17] to calculate the terms C_{avg} and U_{avg}. First, $U_{avg-\Delta}$ is computed for different Δ to generate the conversion chart in Figure 2. Later on, C_{avg} is computed the same way for the systems and the corresponding delay is extracted from the conversion chart.

5. RESULTS

5.1. Models and Offline Accuracy

We constructed two LSTM-based models with different model sizes. The smaller one uses 1-head attention and was trained with the attention-based constraint loss proposed in [17] to prevent the attention function from using future context, while the bigger uses 8-head attention and produces better accuracy. The smaller model *S1* can be used either for inference or to extract the endpoint of a hypothesis prefix following [17]. Additionally, we experiment with a transformer model which has 24 self-attention encoder layers and 8 decoder layers.

ID	Model Type	#Params	SWB	CH
S1	6x2 LSTM-1024	162M	5.8	11.8
S2	6x2 LSTM-1536	258M	5.3	11.5
T1	24x8 Transformer	111M	5.8	11.9
E1	S1 + S2	420M	5.3	10.9
E2	S1 + S2 + T1	531M	5.0	10.1

Table 1. Experimental systems and their offline accuracy. The optimal beam size of 8 was found for all the systems.

Table 1 shows the offline performance of all the investigated S2S models in this work. The big LSTM model achieved the best WER performance while the transformer performs worse. However, the transformer is very efficient to supplement the LSTM models in the combination. The ensemble of 3 models (labeled as *E3*) results in a single system that achieved a 5.0% WER on the SWB test set, which is on par with the state-of-the-art performance on this benchmark.

5.2. Latency with Shared Prefix

We use an audio chunk size of 300ms to perform incremental inference with the systems in Table 1. All inferences were performed on a single Nvidia Titan RTX GPU. Table 2 shows the WERs for SWB, *computation latency* and *confidence latency* (see Section 3) for different beam sizes when only using the *share prefix* strategy for stability detection.

As can be seen, the confidence latency is much larger than the computation latency in all the experiments and shown to be a more critical factor for final latency improvement. The systems involving multiple S2S models require more computational power, however, they obtain better confidence latency and accuracy due to the reduction of model uncertainty.

When using a high beam size (e.g., 8), all the experimental systems can achieve their offline accuracy. This result reveals interesting observations for making online S2S ASR systems. First, as this condition is reliable among different S2S architectures, it shows that all S2S ASR models may share the same characteristic in which they tend not to use further context for the inference of a given prefix at a particular time. This observation is consistent with the finding in [17] for the LSTM-based S2S model. Secondly, it proves that the use of bidirectional encoders in online conditions is possible and even results in the same optimal accuracy as in offline inference. Lastly, it reveals a unified approach to build online ASR for different S2S architectures. As an attractive advantage, this approach does not require model modifications.

The best system using the shared prefix condition achieved a WER of 5.0% and suffered an average delay of 1.79 seconds which is slightly slower than the one with lowest latency.

5.3. Trade-off for Better Latency

To further improve the latency, we use both the stability detection strategies from Section 2.3. We do the combination via a logical OR which means the stability is detected as soon as one of the conditions applies. At the end, we can trade-off latency against accuracy as the function of the term Δ – the delay time needed to finalize the endpoint of a prefix. Figure 3 presents the trade-off curves for two systems, *S1* and *E2*. In both systems, the model *S1* is used for detecting the *best-ranked prefix* condition.

Model	Beam Size	Comp.	Conf.	SWB
S1	8	0.10	1.50	5.8
S2	8	0.13	1.55	5.6
T1	8	0.19	1.50	5.8
T1	6	0.16	1.35	5.9
T1	4	0.12	0.70	6.6
E1	8	0.18	1.55	5.3
E2	8	0.29	1.50	5.0
E2	6	0.25	1.30	5.1
E2	4	0.20	0.80	5.7

Table 2. Computation and confidence latency when using shared prefix condition.

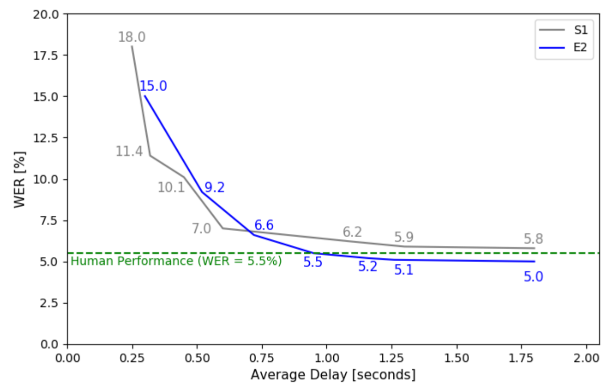

Fig. 3. Trade-off between latency and accuracy. Beam size of 8 is used for both systems.

As can be seen, both systems can achieve much better latency (of only 1.30 seconds) with only a slight increas in WER (e.g., 0.1% abs.). The ensemble system *E2* achieves a latency of 0.85 seconds while yielding the same accuracy as *S1*. Human performance (5.5%) can be reached with an average delay of only 1 second. Note that, the WER for human performance was extracted as the average of the two studies [18] and [19].

6. CONCLUSION

We have shown a unified approach to construct online and low-latency ASR systems for different S2S architectures. The proposed online system employing three S2S models works either in an accuracy-optimized fashion that achieves state-of-the-art performance on telephone conversation speech or in a very low-latency manner while still producing the same or better accuracy as the reported human performance.

7. REFERENCES

[1] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015.

[2] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP 2016*. IEEE, 2016, pp. 4960–4964.

[3] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP 2018*.

[4] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *Proc. Interspeech 2018*.

[5] Chao Weng, Jia Cui, Guangsen Wang, Jun Wang, Chengzhu Yu, Dan Su, and Dong Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," *Proc. Interspeech 2018*.

[6] Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus Muller, and Alex Waibel, "Very deep self-attention networks for end-to-end speech recognition," *Proc. of Interspeech 2019*.

[7] Thai-Son Nguyen, Sebastian Stüker, Jan Niehues, and Alex Waibel, "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation," in *ICASSP 2020*. IEEE, 2020, pp. 7689–7693.

[8] Thai Son Nguyen, Jan Niehues, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Muller, Matthias Sperber, Sebastian Stueker, and Alex Waibel, "Low latency asr for simultaneous speech translation," *arXiv preprint arXiv:2003.09891*, 2020.

[9] Navdeep Jaitly, Quoc V Le, Oriol Vinyals, Ilya Sutskever, David Sussillo, and Samy Bengio, "An online sequence-to-sequence model using partial conditioning," in *Advances in Neural Information Processing Systems*, 2016, pp. 5067–5075.

[10] Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck, "Online and linear-time attention by enforcing monotonic alignments," in *Proceedings of the 34th ICML*, 2017, pp. 2837–2846.

[11] Chung-Cheng Chiu and Colin Raffel, "Monotonic chunkwise attention," *arXiv preprint arXiv:1712.05382*, 2017.

[12] Ruchao Fan, Pan Zhou, Wei Chen, Jia Jia, and Gang Liu, "An online attention-based model for speech recognition," *Proc. Interspeech 2019*, pp. 4390–4394, 2019.

[13] Haoran Miao, Gaofeng Cheng, Pengyuan Zhang, Ta Li, and Yonghong Yan, "Online hybrid ctc/attention architecture for end-to-end speech recognition," *Proc. of Interspeech 2019*, pp. 2623–2627, 2019.

[14] Emiru Tsunoo, Yosuke Kashiwagi, Toshiyuki Kumakura, and Shinji Watanabe, "Towards online end-to-end transformer automatic speech recognition," *arXiv preprint arXiv:1910.11871*, 2019.

[15] Haoran Miao, Gaofeng Cheng, Changfeng Gao, Pengyuan Zhang, and Yonghong Yan, "Transformer-based online ctc/attention end-to-end speech recognition architecture," in *ICASSP 2020*, pp. 6084–6088.

[16] Niko Moritz, Takaaki Hori, and Jonathan Le, "Streaming automatic speech recognition with the transformer model," in *ICASSP 2020*. IEEE, 2020, pp. 6074–6078.

[17] Thai-Son Nguyen, Ngoc-Quan Pham, Sebastian Stueker, and Alex Waibel, "High performance sequence-to-sequence model for streaming speech recognition," *Proc. of Interspeech 2020*, 2020.

[18] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[19] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, et al., "English conversational telephone speech recognition by humans and machines," *arXiv preprint arXiv:1703.02136*, 2017.

[20] 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*, 2017.

[21] P Brown, J Spohrer, P Hochschild, and J Baker, "Partial traceback and dynamic programming," in *ICASSP 82*. IEEE, 1982, vol. 7, pp. 1629–1632.

[22] Sven Wachsmuth, Gernot A Fink, and Gerhard Sagerer, "Integration of parsing and incremental speech recognition," in *9th of the EUSIPCO 1998*, pp. 1–4.

[23] Ethan O Selfridge, Iker Arizmendi, Peter A Heeman, and Jason D Williams, "Stability and accuracy in incremental speech recognition," in *Proceedings of the SIGDIAL 2011 Conference*, 2011, pp. 110–119.