Reading Ahead: Pace-Based Prediction for Real-Time Tracking of Children's Oral Reading in the Browser

Bohdan Podziubanchuk and **Einar Meister** and **Tanel Alumäe**
Department of Software Science
Tallinn University of Technology
Estonia
{bohdan.podziubanchuk,einar.meister,tanel.alumae}@taltech.ee

Abstract

Interactive reading tutors can support children by highlighting the word currently being read. However, when reading position is determined with the help of a streaming automatic speech recognition (ASR) system, the highlight necessarily lags behind the reader because a word can only be confirmed after it has been emitted by the recogniser. We present a browser-based Estonian reading tutor that reduces this delay by predicting the reader's current position between ASR updates. The proposed prediction layer estimates the reader's pace in voiced seconds per letter and uses the voiced time elapsed since the last confirmed word to extrapolate over the expected durations of subsequent words. Voice activity detection prevents pauses from advancing the predicted position. Compared to ASR-based alignment alone, this approach increases exact word-level tracking accuracy from 49.7% to 57.7% and reduces the median word-transition delay from 264 ms to 87 ms. Both ASR and voice activity detection run locally in the browser using WebAssembly, ensuring that no audio data leave the user's device.

1 Introduction

Systems that listen to a child read aloud need to continuously track which word the child is currently reading. This position determines the visual feedback shown by the system, typically by highlighting the current word to help the child keep track of their place in the text. For this feedback to be useful, the highlighted position should closely match the child's actual reading position. Highlighting the wrong word may be confusing or distracting and can even disrupt reading by causing the child to lose their place. Similarly, if the display does not advance after a word has been read, the child may interpret this as the system failing to recognize or accept the word.

Tracking reading position from streaming automatic speech recognition (ASR) is subject to inherent delay. A recogniser cannot identify a word before sufficient acoustic evidence has been observed, while architectural factors such as audio buffering and model lookahead introduce additional latency. In our system, the ASR model begins emitting the tokens of a word approximately 640 ms after the child starts pronouncing it. Delays of this size are perceptible: response times above about 100 ms are generally noticed by users of interactive systems (Miller, 1968; Card et al., 1983). Any matcher that waits for these tokens is therefore delayed by design, causing the highlighted position to trail the child's actual reading position.

Li and Mostow (2012) measured this effect for an English reading tutor and showed that tracking accuracy improves substantially if the tutor is allowed to lag behind the recogniser's frontier: exact accuracy rose from 58% at zero lag to 70% at 200 ms and 76% when the final hypothesis was used. Accuracy and timeliness were traded against each other.

In this paper, we aim to reduce this delay rather than trade it for accuracy. Instead of waiting for further recogniser evidence, we predict the reader's current position. Oral reading proceeds at a pace that can be estimated from the words the recogniser has already confirmed. Our system estimates this pace in voiced seconds per letter and uses it to carry the position forward from the last confirmed word, by at most two words. Two supporting components are required: voice activity detection, which prevents silence from advancing the position, and a correction for the latency of the recogniser's token timestamps. Throughout the paper, we use *tracking* to refer to estimating the reader's current position, which in our system

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 15th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2026)
pages 288–294, November 19–20, 2026, University of Tartu Library

combines recogniser evidence with a pace-based prediction.

We demonstrate the approach in a web application for Estonian, where both the speech recogniser and the voice activity detector run locally in the browser using WebAssembly. Since no audio is transmitted to a server, the system offers stronger privacy guarantees. On 134 recordings from nine children, manually annotated at the word level, the prediction layer improves exact tracking accuracy by 8.0 percentage points and reduces the median transition delay by approximately a factor of three, with improvements observed for every child.

2 Related work

Reading tutors that listen have a long history. Project LISTEN (Mostow and Aist, 1999) used ASR to follow children reading aloud, detect miscues and give feedback, and established most of the evaluation practice in this area (Mostow, 2006). Position tracking was recognised early as a distinct problem from recognition accuracy: word error rate does not measure whether the tutor knows where the reader is (Mostow, 2006).

The work closest to ours is the evaluation of real-time tracking by Rasmussen et al. (2011) and Li and Mostow (2012). The latter distinguishes tracking computed from the recogniser's successive partial hypotheses, which is what a live tutor must use, from tracking computed after the fact from the final hypothesis, which can overestimate real-time performance. It defines exact accuracy as the share of time the displayed position matches the true position and near accuracy as the share within one word. We adopt both metrics. The main difference is that Li and Mostow (2012) improve tracking accuracy by introducing additional lag, whereas our goal is to reduce this lag by predicting the reader's position during the recognition delay.

A recent alternative avoids the cascade entirely: Sunder et al. (2023) train a pointer network that predicts positions in the reference text directly from streaming audio, reporting accuracies of 77.1% on CMU Kids and 65.3% on Reading Races. That approach needs aligned training data in the target language and a trained model at inference time. Our setting is a cascaded one, built on an existing streaming recogniser, which is what is available for a language the size of Estonian and what can be executed inside a browser.

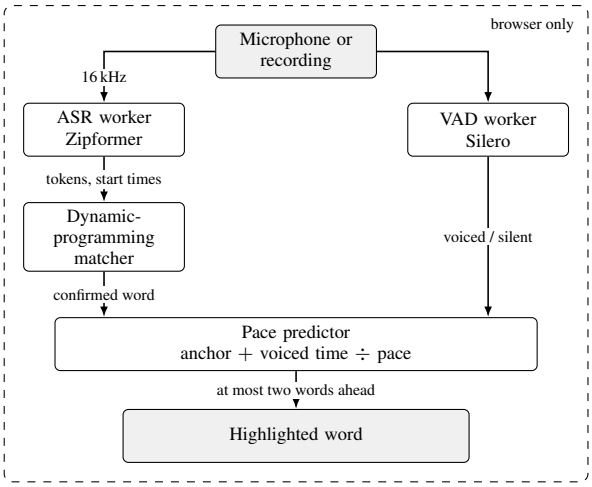


Figure 1: Architecture. Recognition and voice activity detection run as WebAssembly workers inside the browser. The matcher yields the confirmed position; the pace predictor carries it forward by the voiced time that has passed since the confirmed word began.

The task is also related to score following in music, where a system tracks a performer's position in a known score in real time (Orio et al., 2003). A common assumption in score following is that the performance follows a tempo that can be estimated from recent observations and extrapolated forward. Our approach applies a similar idea to oral reading: we estimate the reader's current pace and use it to predict their position between recognition updates. To our knowledge, this principle has not previously been applied to real-time tracking of oral reading.

3 System

The application is a single page in a Next.js web application. The user loads a reading text, presses record, and reads aloud; the word the system believes is being read is highlighted. Everything described below runs in the browser: ASR and voice activity detection are WebAssembly modules in dedicated workers, and no audio or transcript is sent anywhere. Figure 1 shows how the parts fit together. Source code is available at `https://github.com/Danbog32/estonian-reading-tracker` and a demo video at `https://github.com/Danbog32/estonian-reading-tracker#demo`.

The reading tracking system will be integrated into a reading tutor project funded by the Estonian Ministry of Education and Research (EKTR29).

3.1 Speech recognition

For ASR, we use a general-purpose Estonian streaming model[1] based on the Zipformer neural transducer architecture (Yao et al., 2024), trained using the Icefall toolkit[2]. The model has about 67 million parameters and was trained on roughly 1334 hours of manually transcribed Estonian speech from the TalTech Speech Dataset 1.0[3], around 4000 hours of automatically transcribed Estonian broadcast data, and a 500-hour subset of the Gigaspeech dataset (Chen et al., 2021), which was included so that the model can transcribe English terms often embedded in Estonian sentences. The training data contain adult speech only; the model has not been adapted to children's speech.

Audio is captured through the Web Audio API, resampled to 16 kHz and passed to the recogniser in 100 ms chunks. Recognition is executed with sherpa-onnx[4] compiled to WebAssembly, with greedy decoding. The recogniser emits partial hypotheses as audio arrives and a final hypothesis when an utterance ends.

For each hypothesis the worker exposes the underlying token sequence together with the start time of every token and its log probability. The token start times are the anchor the prediction layer needs: they say when a sound was produced, not when the recogniser reported it.

3.2 Matching the transcript to the text

The reading text is known in advance, so matching is a monotonic alignment problem rather than open recognition. On every recogniser event the matcher aligns the last ten transcript words against a window of the reading text that extends three words behind the current position and up to the end of the sentence ahead of it, using dynamic programming.

Word similarity combines ordinary edit distance with a phonetic key that collapses the contrasts Estonian beginning readers most often blur: voiced and voiceless stop pairs, doubled letters, and vowel quality. A child who produces *leib* for *loeb* scores 0.5 on letters but 1.0 on the phonetic key. Each candidate alignment receives a confidence score combining the quality of the matched pairs, how much of the transcript window found a match, the total alignment cost and a small term (weight 0.06) that favours alignments advancing the position further, with penalties for skipped and unmatched words. The position advances only when that score clears a threshold, and never moves backwards. This matcher is the system's confirmed position and is used unchanged as the baseline in Section 4.

3.3 Pace prediction

The matcher can confirm a word only after the recogniser has produced evidence for it. The prediction layer operates on top of the matcher and does not modify its output.

Anchor. The prediction is anchored at the most recently confirmed word and at the estimated time when the child began pronouncing it. This time is obtained from the ASR token start times after correcting for the systematic latency of the recogniser's timestamps (Section 4.4).

Pace. For each confirmed word, the layer estimates the reading pace as the voiced duration of the word divided by its number of letters. Estonian spelling is close to phonemic, so the number of letters approximates the number of speech sounds. Within a child, word duration in our data is approximately proportional to word length in letters. The current pace is the median of the eight most recent samples. A median is used because a single word on which the child hesitates would substantially increase a mean and delay all subsequent predictions. The history length of eight words was fixed before the evaluation as a compromise between adapting to changes in pace and stability of the estimate. Words on which the child lingers, for example by sounding them out or prolonging them, are referred to as held words. They are detected as samples exceeding 2.5 times the current pace and are excluded from the history.

Projection. Starting from the anchor, the layer uses the voiced time elapsed since the anchor word began to estimate how far the child has progressed. The expected duration of each following word is computed as the current pace multiplied by its length in letters. The elapsed voiced time is then allocated to successive words, and the word in which this time budget is exhausted is selected as the predicted position. The projection is recomputed from the anchor after every recogniser event

[1] https://huggingface.co/TalTechNLP/streaming-zipformer.et-en

[2] https://github.com/k2-fsa/icefall

[3] https://cs.taltech.ee/staff/tanel.alumae/data/est-pub-asr-data/

[4] https://github.com/k2-fsa/sherpa-onnx

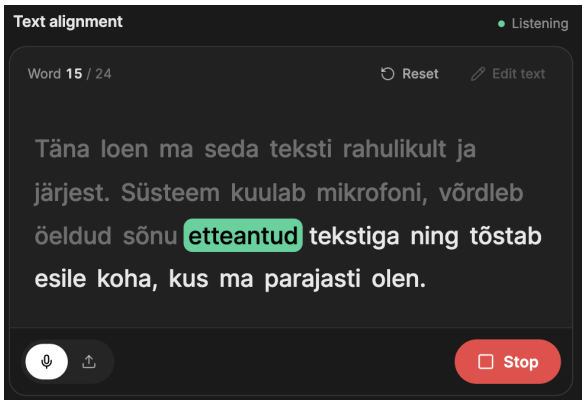


Figure 2: The reading view during live microphone input. Words already read are dimmed, the current word is highlighted, and the source switch and the start/stop control sit below the text.

and every 100 ms rather than updated incrementally. As a result, if a word is confirmed later than expected, the next prediction already takes into account all voiced time since the onset of the anchor word. The prediction is limited to at most two words beyond the anchor, that is, at most one word ahead of the matcher's current position, and is never placed behind the position confirmed by the matcher.

Pauses. Only voiced time contributes to the projection. A separate Silero voice activity detector (Silero Team, 2021) runs alongside the recogniser using a 0.15 s minimum silence duration. This is short enough to capture between-word pauses, whose median duration in our data is 259 ms. During detected silence, the elapsed-time budget remains unchanged and the highlight therefore does not advance.

Repetitions and self-corrections. The matcher aligns the transcript against a window that begins three words before the current position, so repeated words are absorbed as insertions and do not move the confirmed position backwards. If the child repeats or prolongs the anchor word, the recogniser produces further evidence for it; the layer then treats the word as held, extends its expected duration, and the prediction returns to the matcher's position. This is the main source of rollbacks. After a self-correction, the matcher may confirm several words in a single update, in which case the highlight advances by more than one word at once.

3.4 Interface

Figure 2 shows the interface. The user pastes the reading text, chooses between the microphone and an uploaded recording, and presses start. The current word is highlighted. A word to which the prediction layer has moved ahead of the matcher is shown in a lighter shade, so that a prediction that later proves wrong is withdrawn less conspicuously. The panel also reports the measured reading speed in words per minute. If voice activity detection fails to load, the panel reports that the speed is unavailable.

4 Evaluation

4.1 Data

Estonian read-speech data were collected from children aged 6 to 11, irrespective of their native language. The participants' native languages include Estonian, Russian, English, Nepali, and Yoruba. To date, 101 children (54 girls and 47 boys) have been recorded, yielding 2,551 recordings and approximately 14 hours of speech. The recordings were conducted in a small, quiet room at the children's schools using a web-based application and a close-talking headset microphone. Audio was stored in uncompressed mono WAV format at a sampling rate of 44.1 kHz and 16-bit resolution. The ongoing annotation process covers both phoneme- and word-level annotation, as well as the classification of speech errors, including substitutions, omissions, insertions, and repetitions. Written informed consent was obtained from a parent prior to recording. Speech data collection was conducted as part of a reading tutor project. The data collection protocol was approved by the ethics committee of Tallinn University of Technology.

We evaluate on all recordings for which manual word-level annotation was available at the time of the experiments: 134 recordings from nine children, five girls and four boys, reading Estonian texts aloud, 43.8 minutes in total. Word boundaries were annotated manually in Praat TextGrids, yielding 2,072 labelled word intervals. Since manual annotation at the word level is time-consuming, the remaining recordings have not yet been annotated at this level. The recordings cannot be released, as the consent covers research use only; the tracking algorithms are available with the source code.

	Matcher	+ pace
Exact accuracy	49.7%	**57.7%**
Within one word	79.0%	**83.5%**
Median delay	264 ms	**87 ms**
False jumps / min	1.33	**0.64**
Rollbacks / min	0.00	0.36
Ahead of the child	6.0%	13.5%
Behind the child	33.4%	17.9%

Table 1: Tracking quality over 134 recordings from nine children.

Variant	Exact	Delay
Full system	57.5%	87 ms
One pace for every child	48.6%	80 ms
No voice activity detection	53.5%	90 ms
No latency correction	53.1%	198 ms
Lead capped at one word	49.2%	257 ms
Lead capped at three words	57.1%	79 ms
Also held behind the matcher	51.9%	161 ms

Table 2: Ablations, one component removed or modified at a time. These runs predate a later refinement of the matcher, which changes two recordings; the full system therefore reads 57.5% here and the matcher alone 49.2%.

To characterise recognition quality on this subset, we computed the word error rate (WER) and character error rate (CER) of the recogniser's final hypotheses. Measured against the words the children actually produced, as transcribed in the annotation and excluding false starts, the WER is 22.0% and the CER 11.2%; the per-child WER ranges from 8.5% to 37.0%. Measured against the prompt text, the WER is 25.5% and the CER 13.3%. The difference mainly reflects reading errors, as the children's productions differ from the prompt text in 7.0% of words.

The recordings are replayed through the same recogniser worker the application uses, and all systems consume the same captured recogniser events, so differences between them come only from the tracking logic and not from variation in recognition. Positions are sampled every 100 ms while a labelled word is being spoken.

4.2 Metrics

Following Li and Mostow (2012) we report *exact accuracy*, the proportion of sampled timesteps at which the highlighted word is the word being read, and *within one word* (near accuracy in Li and Mostow, 2012), the share at which it is off by at most one word. We add three measures that matter for a live interface. *Median delay* is the time from the onset of a word until the highlight is at or past it. *False jumps* per minute counts movements to a word the child is not reading. *Rollbacks* per minute counts movements of the highlight to an earlier word; the matcher alone never rolls back, while a predictor can.

4.3 Results

Table 1 shows the effect of the prediction layer. Exact accuracy rises by 8.0 points and median delay falls from 264 ms to 87 ms. False jumps fall by half: predicting the next word from pace is more reliable than accepting a weak acoustic match for it. The cost is 0.36 rollbacks per minute, roughly one every three minutes.

Exact accuracy, within-one-word accuracy, and median delay improve for all nine children. At the level of individual recordings, performance improves for 102 recordings, degrades for 21, and remains unchanged for 11. The largest gains are observed for the children for whom the baseline matcher performs worst: among the three children with baseline exact accuracy below 30%, the prediction layer improves accuracy by 14–22 pp.

With pace prediction, the highlight is ahead of the child's true position more often (13.5% *vs* 6.0%), but behind substantially less often (17.9% *vs* 33.4%). Since the delay metric considers the highlight to have arrived once it reaches or passes the current word, part of the reduction in delay reflects this increased forward bias. However, exact accuracy penalizes predictions that are ahead and behind equally, and its improvement therefore provides separate evidence that the prediction layer tracks the reading position more accurately overall.

4.4 Ablation study

Table 2 reports the results when individual components are removed or modified. The largest decrease occurs when a single constant pace is used for all children: in this condition, exact accuracy falls below that of the matcher alone. This is consistent with the variation in reading pace in our data, where the time per letter ranges from 88 to 257 ms across recordings. Removing voice activity detection reduces exact accuracy by 4.0 percentage points, and removing the latency correction reduces it by 4.4 points and increases the median delay by 111 ms.

The latency correction is based on a compari-

son of token start times with the manual annotations, which showed that the onsets reported by the recogniser follow the annotated onsets by approximately half a second. For each child, the layer subtracts an offset estimated from the other eight children only, so that no correction is derived from the child's own data. Without this correction, the onset of the anchor word is estimated approximately half a second too late, and most of the improvement is lost.

When the lead is limited to one word, the results are identical to those of the matcher (49.2% and 257 ms in the same run), which indicates that the layer does not alter the output when it cannot advance beyond the matcher. Allowing a lead of three words does not improve accuracy compared with two words but increases the rate of rollbacks from 0.36 to 2.97 per minute; we therefore use a limit of two words.

An earlier version of the layer could also keep the highlight on a word behind the matcher when the estimated pace indicated that the child had not yet finished it, with the aim of handling long words that are read slowly. This variant reduced exact accuracy by 5.6 percentage points. A likely explanation is that onsets estimated from token times are accurate only to within a few hundred milliseconds, which is sufficient to retain the wrong word. The observed improvement therefore appears to result from advancing the position rather than from holding it.

5 Limitations

The evaluation covers one language, one recogniser and one group of readers. The approach itself assumes only a streaming recogniser that reports token times, but the values used here, the length of the pace history, a lead of two words and the latency offset, were established for this configuration and would have to be re-established for another. The configuration of the layer was developed on the same recordings that were used for evaluation, and its effect should therefore be confirmed on new data. The recogniser was trained on adult speech only, which is a likely contributor to the variation in word error rate across children.

Pace prediction relies on the assumption that reading speed can be extrapolated from recent observations. When a reader pauses to decode a difficult word, the predicted position continues to advance according to the estimated pace and is corrected only after the recogniser produces sufficient evidence. Limiting the prediction lead to two words and using a lighter highlight for predicted words reduce the visibility of such corrections but do not eliminate them. A related limitation occurs at the beginning of a text, when insufficient observations are available to establish a reliable pace estimate. An unusually slow first word may therefore bias the estimate until additional observations become available.

The present study does not examine how real-time highlighting of the reading position affects learners, particularly when the highlighted position is occasionally inaccurate. Such feedback may introduce distractions, encourage over-reliance on visual cues, or lead to unnatural reading habits. Future work will investigate whether real-time highlighting improves reading performance and learning outcomes, and how these benefits are affected by highlighting errors.

6 Conclusion

Most of the delay in a reading tracker originates from ASR rather than from the matching process itself. A streaming ASR system typically emits a word several hundred milliseconds after the child begins or completes saying it, so any matcher that waits for this evidence will necessarily lag behind, regardless of how accurate the matching is. Much of this delay can be recovered by estimating the child's reading pace and using the elapsed voiced time to predict the current position. The same mechanism also improves tracking accuracy, since advancing based on the estimated reading pace can be more reliable than accepting a weak acoustic match. Three components are particularly important: a voice activity detector with a short silence threshold, which prevents pauses from advancing the predicted position; a measured correction for the recogniser's systematic timestamp latency; and a median over recent words, which makes the pace estimate robust to individual words that are read unusually slowly.

Acknowledgments

This work was supported by the Estonian Ministry of Education and Research under project EKTR29, and by the Estonian Centre of Excellence in Artificial Intelligence (EXAI), funded by the Estonian Ministry of Education and Research grant TK213.

AI Disclosure

☐ Not applicable (no AI tools used)
☒ Assistive use
☒ Editing
☐ Generation of minor elements
☐ Translation
☐ Compression/expansion
☐ Generation from prompt

Details: AI assistants (Claude, Anthropic) were used to discuss the reviewers' comments, to check the description of the system against its source code, and to improve grammar, style and clarity of the revised text. The method, experiments and results are the authors' own, and all text was reviewed and edited by the authors.

References

Stuart K. Card, Thomas P. Moran, and Allen Newell. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Interspeech 2021*. ISCA.

Yuanpeng Li and Jack Mostow. 2012. Evaluating and improving real-time tracking of children's oral reading. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*.

Robert B. Miller. 1968. Response time in man-computer conversational transactions. In *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 267–277.

Jack Mostow. 2006. Is ASR accurate enough for automated reading tutors, and how can we tell? In *Proceedings of Interspeech*, pages 837–840.

Jack Mostow and Gregory S. Aist. 1999. Giving help and praise in a reading tutor with imperfect listening because automated speech recognition means never being able to say you're certain. *CALICO Journal*, 16(3):407–424.

Nicola Orio, Serge Lemouton, and Diemo Schwarz. 2003. Score following: state of the art and new developments. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME)*, pages 36–41.

Morten Hjelm Rasmussen, Jack Mostow, Zheng-Hua Tan, Børge Lindberg, and Yuanpeng Li. 2011. Evaluating tracking accuracy of an automatic reading tutor. In *Proceedings of the ISCA SIG Workshop on Speech and Language Technology in Education (SLaTE)*.

Silero Team. 2021. Silero VAD: pre-trained enterprise-grade voice activity detector. https://github.com/snakers4/silero-vad.

Vishal Sunder, Beulah Karrolla, and Eric Fosler-Lussier. 2023. End-to-end real time tracking of children's reading with pointer network. *arXiv preprint arXiv:2310.11486*.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. In *International Conference on Learning Representations (ICLR)*.