Evaluating Tracking Accuracy of an Automatic Reading Tutor

Morten Højfeldt Rasmussen[1], *Jack Mostow*[2], *Zheng-Hua Tan*[1], *Børge Lindberg*[1], *Yuanpeng Li*[2]

[1]Department of Electronic Systems, Aalborg University, 9220 Aalborg Ø, Denmark
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
{mr,zt,bli}@es.aau.dk[1], {mostow,yuanpeng.li}@cs.cmu.edu[2]

Abstract

In automatic reading tutoring, tracking is the process of automatically following a reader through a given target text. When developing tracking algorithms, a measure of the tracking accuracy – how often a spoken word is aligned to the right target text word position – is needed in order to evaluate performance and compare different algorithms. This paper presents a framework for determining the observed tracking error rate. The proposed framework is used to evaluate three tracking strategies: A) follow the reader to whichever word he/she jumps to in the text, B) follow the reader monotonically from left to right ignoring word skips and regressions (going back to a previous text word), and C) the same as B but allowing isolated word skips. Observed tracking error rate for each of the three tracking strategies is: A: 53%, B: 56%, and C: 47%, on 1883 utterances from 25 children.

Index Terms: Automatic Reading Tutor, Tracking Speech, Tracking Error Rate

1. Introduction

One of the tasks of automatic reading tutors (ART) is to help students to become better readers by tracking the student's position, using a combination of automatic speech recognition (ASR) and an alignment algorithm, and using this information to provide help and support when needed. The reading tutor can provide word-level assistance in a number of ways, e.g. by providing a visual representation of the word or by reading the word out loud [1]. Accurate tracking is important in order not to discourage the reader from using the system; if the system makes too many mistakes the reader will lose confidence in it. Also, the system's ability to detect misreading requires knowing which word the student is (or should be) trying to read. Within the area of ART much research has been conducted with the goal of improving miscue detection accuracy – where the focus is on locating misread words and other disfluencies [2], [3], [4]. Tracking the reader's position in a text has received less attention [5], [6]. This paper focuses on the latter.

The work presented here compares three strategies for automatically tracking the reader's position in the prompt (or target) text: A) tracking by following the reader wherever he/she jumps in the target text (chase-the-reader), B) tracking by advancing strictly from left to right, staying put when the reader regresses to earlier in the text or skips words (left-to-right), C) tracking as in B but allowing for skipping one word at a time (L2R-skip-one). In order to compare the three tracking strategies, an error measure is needed. However, as the task of tracking is different from detecting words, simply calculating the word error rate of the ASR output is not an option. Since timing of the tracking method is of importance for some real-time automatic reading tutors and in order to give an honest performance-measure and (to the authors' knowledge) since there are no standard ways of evaluating tracking accuracy when considering timing in ART, we develop a new framework.

The rest of the paper is organized as follows: Section 2 describes the three tracking approaches. Section 3 describes the evaluation setup. Section 4 presents and discusses the results. Section 5 concludes.

2. Tracking

We distinguish the automatic reading tutor's internal estimate of the reader's position from the position the tutor displays externally as feedback. The two estimates may be the same but can be different. The reading tutor might e.g. only update the displayed position when a certain amount of silence has been observed. Though the final goal is to determine the accuracy of the displayed position, the accuracy of the internal estimate can more easily be determined quantitatively and will therefore be the focus of this paper.

2.1. Automatic tracking methods

The automatic reading tutor estimates the current target word position by using a speech recognizer ("ASR" in Figure 1) in connection with an alignment algorithm ("Align ASR output to target text" in Figure 1). The task of the speech recognizer is to recognize the target words uttered by the reader and to handle miscues and the alignment algorithm aligns the speech recognizer's output to the target text. The language model [7] is the same for all presented tracking methods. It allows for transitions between all target words and is thus capable of modeling both regressions, skips, and going forward one word – with a very high probability of going forward one word. We assume that the more faithfully the language model models actual reading behavior, such as regressions, the more accurate the speech recognizer will be. We call a sequence of aligned recognized words an ART trace.

The chase-the-reader tracking strategy requires a method capable of tracking regressions and word skips. We use dynamic programming to find the alignment of the recognized words to the target text with the lowest cost. The cost of aligning the recognized word to target word position is the sum of transition costs plus the costs of aligning recognized words with target words having different orthographies. The cost associated with transitions is 0 for advancing one word position at a time, 0.01 for staying at the same word position, and 1 for jumping. The cost is 0 for aligning a recognized word to a target word when the orthographies are identical and 1 when the orthographies differ. The left-to-right tracking strategy requires a method that ignores regressions and word skips. The recognized words are aligned to the target text by starting from target word 1 and only allowing transitions from word n to $(n+1)$ if word n is accepted as being read correctly. The L2R-skip-one approach is like left-to-right but allows for isolated word skips (transitions from word n to $(n+2)$). We believe that this relaxation of the tracking constraint in the L2R-skip-one method will prevent it from getting stuck at word n in some cases.

2.2. Creating reference traces

We define a trace as a sequence of integers whose absolute value represents a target word position and whose sign represents whether the word is read correctly. Reference traces are created for each tracking strategy and could be created by human reading tutors. However, since this would be a time-consuming task the traces are created automatically based on the human transcription of what has been uttered. We create reference traces by forced alignment of the human-transcribed words to the speech ("Forced alignment", Figure 1) and aligning each transcribed word to a target word ("Align transcription to target text", Figure 1). This approach has the added benefit of making the evaluation method easily adaptable to other tracking strategies.

Chase-the-reader alignment of the transcribed words to the target text is done by using dynamic programming as in the automatic tracking case. Here the transition cost is 0 for advancing one word position, 0.01 staying at the same word position or jumping back, and the number of skipped words is the cost of jumping forward. The cost of aligning a transcribed word to a target word is the minimum of either the Levenshtein distance between the sequence of letters in the words, the Levenshtein distance between the sequence of phonemes in the words, or a cost for spelling out the word as a sequence of letter names (e.g. reading "*C A T*" instead of "*cat*"). The cost of spelling out a word (or part of the word) is the reciprocal of the number of letters in the word for the first letter plus the reciprocal of the number of letters in the word times the number of times one or more letters are skipped. In this way the cost of one spelling attempt, without restarts, is never larger than one.

The cost of spelling out a word (or a part of it) is the reciprocal of the number of letters in the word plus the number of skipped letters normalized by the number of letters in the word.

The left-to-right and L2R-skip-one alignments are done in the same way as they are done in the automatic tracking case (Section 2.1).

3. Evaluating tracking accuracy

An overview of the evaluation setup can be seen in Figure 1. Everything to the left of the dashed line is concerned with creating a reference trace and an ART trace (see Section 2). The module named "Calculate tracking error rate" to the right of the dashed line compares the two traces and calculates the tracking error rate.

3.1. Comparing ART and reference traces

One way of visualizing tracking is to create a stair case plot as shown in Figure 2. The plots show how the reader's actual utterance *"it was a pig pig it was his pig pen"* is aligned to the target text *"It was his pig pen."* for each of the three tracking strategies. On the x-axis are the target words with indices, where the symbol "<b>" is used for the case where the reader has not started reading yet and the symbol "<e>" is used when the reader has finished reading the target text. The solid line segments between two '•' markers correspond to reference speech events and the segments between two '+' markers correspond to ART speech events. The words in the boxes (e.g. "#>IT") correspond to the transcribed event (#) and the recognized event (IT) separated by an alignment symbol (>). We pair transcribed and recognized events as follows. We consider three different types of alignment: the midpoint of each falls within the other's segment (:), the midpoint of the reference event falls within the recognized event segment but not vice versa (<), and the midpoint of the recognized event falls within the reference event segment but not vice versa (>). We use this criterion as we are only interested in knowing which reference speech event overlaps with which recognized event and not the exact boundaries. Figure 3 shows all the paired events for the specific case of the L2R-skip-one setup. Here '+' indicates a correctly read word, '-' indicates a miscue, and # is silence. The integer indicates the positional error, i.e. how many words the reading tutor is ahead of the reference (if the number is negative the ART lags behind the reference).

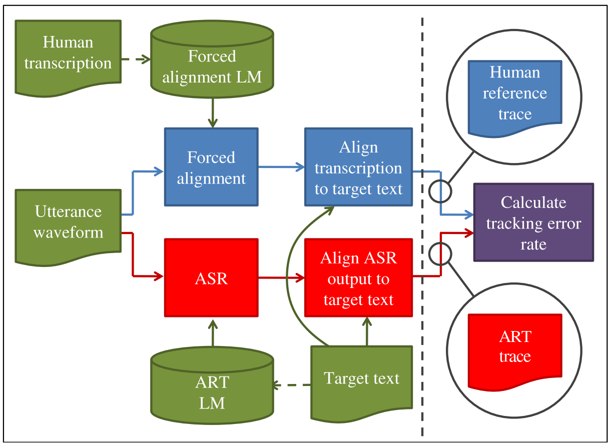


Figure 1: *Evaluation setup overview.*

Both the stair case for the L2R-skip-one in Figure 2 and the speech event pairs in Figure 3 show that the first two ART speech events (pair 1 and 2) are not aligned to reference speech events – in other words, the speech recognizer hallucinated the two events (made insertion errors). Another interesting event pair is number 7. Here the child reads the word *"a"* which is being recognized as the word *"his"*. Chase-the-reader aligns the reference trace for this word to the target word *"his"* and both left-to-right and L2R-skip-one aligns it to the word *"was"*.

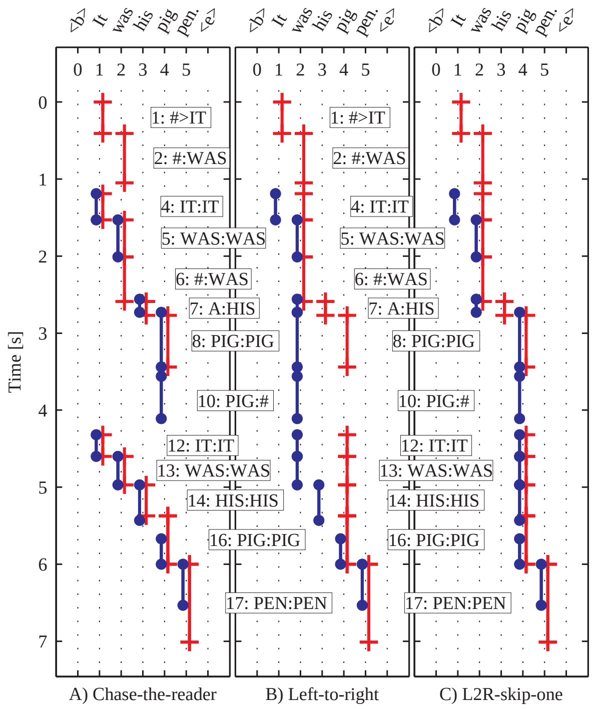
Event pairs where a silence segment's midpoint falls within a speech event segment – reference or recognized – but not vice versa are ignored (18), and so are pairs of silence events (3, 9, 11, 15, and 19).

3.2. Tracking error rate

We define tracking error rate (TER) as the number of times the reading tutor is off track. Alternative measures include the percentage of time off track, or the average positional error (calculated as e.g. mean absolute error or root mean square error). The tracking errors can be partitioned in three groups as shown in Equation 1, normalized by the number of speech events in the human transcription.

$$TER = \frac{I+D+S}{R}, \quad (1)$$

Here I is the number of times the reading tutor detects a target word when no speech events occur, D is the number of times an actual speech event is not assigned a target word position by the reading tutor, S is the number of times a speech event is assigned a wrong target word position by the reading tutor, and R is the total number of reference speech events. I, D, and S are analogous to the insertion, deletion, and substitution errors of the word error rate. The TER for the examples in Figure 2 is $4/10$ for chase-the-reader, $10/10$ for left-to-right, and $6/10$ for L2R-skip-one.



A) Chase-the-reader B) Left-to-right C) L2R-skip-one

Figure 2: *Comparing time-aligned ART traces (red plus-capped lines) with time-aligned reference traces (blue circle-capped lines). Spoken: "it was a pig pig it was his pig pen", recognized: "it was it was was his pig it was his pig pen".*

Since the purpose of internal tracking is to provide information about the most recently accepted target word position, two types of errors will be masked. The first is when the reader reads a single word from the target text and the reading tutor correctly tracks the word once but then hallucinates it a second time or more (event pair number 6 in Figure 2). This repetition error will not change the last observed correctly read word. The second is when the reader reads the same target word twice or more and the ART correctly tracks the first word but misses the succeeding words (number 10 in Figure 2). This deletion error will not change the last observed correctly read word. In both cases the reading tutor is still on track after making a mistake. We call the TER ignoring these two types of errors "observed tracking error rate" or $OTER$. The $OTER$ for the examples given in Figure 2 is $2/10$ for chase-the-reader, $9/10$ for left-to-right, and $4/10$ for L2R-skip-one as pair number 6 is masked for all three and pair number 10 is masked for chase-the-reader and L2R-skip-one. Note that the number of reference events is the same for all methods (10) since the same ASR configuration is used for all three tracking methods.

```
1: (#>+)1     8: (+:+)0    15: (#<#)0
2: (#:+)2     9: (#<#)0    16: (+:+)0
3: (#>#)2    10: (+:#)0    17: (+:+)0
4: (+:-)1    11: (#<#)0    18: (#<+)0
5: (+:+)0    12: (-:-)0    19: (#>#)0
6: (#:+)0    13: (-:-)0
7: (-:+)1    14: (-:-)0

 '#' silence,  '+' accepted word,
 '-' rejected word

 t>r  t contains midpoint of r
 t:r  t contains midpoint of r
      and vice versa
 t<r  r contains midpoint of t
```

Figure 3: *Aligned event pairs.*

4. Results and discussion

The basic setup for testing the three tracking methods is the same – with the overall setup as in Figure 1 and using the same acoustic and language models – the only difference being the alignment methods used (called "Align to target text" in the Figure). The language model is created from the target text as a finite state grammar with all possible transitions between words – with higher penalties for regressions and skips than for transitions from one word to the next as this is the expected reading behavior. The test set used in the experiment is comprised of 1883 utterances spoken by 25 children.

In order to show the difference between TER and $OTER$, both results are presented in Table 1. A significance test (Friedman, children as blocks, tracking strategies as treatments) shows that the three mean $OTERs$ are not the same, at $p < 0.001$. Subsequent pair-wise comparison (Wilcoxon Signed-Rank test, children as subjects) show that $OTERs$ for chase-the-reader and L2R-skip-one differ ($p = 0.037$, two-tailed), and that $OTERs$ for left-to-right and L2R-skip-one differ ($p < 0.0001$, two-tailed) – we cannot say that the $OTERs$ for chase-the-reader and left-to-right differ with high probability ($p = 0.11$, two-tailed). These tests were chosen since the $OTERs$ are not normally distributed and that the data is paired (different tracking strategies on the same data).

Table 1: Tracking error rate in percent for the three methods.

	TER	$OTER$	Masking effect
Chase-the-reader	55.6%	52.8%	2.8%
Left-to-right	69.1%	56.2%	12.9%
L2R-skip-one	60.0%	46.5%	13.5%

For each method of tracking, we define $TER - OTER$ as its masking effect. Note the large masking effect for left-to-right (12.89%) and L2R-skip-one (13.42%), compared to the masking effect for chase-the-reader (2.79%). This shows that a large percentage of repetition and deletion errors are masked for left-to-right and L2R-skip-one. The chase-the-reader tracking method has the smallest TER, the L2R-skip-one has the smallest $OTER$. Since we argue that $OTER$ is more suited for evaluating the accuracy of automatic reading tutors than TER, we prefer the L2R-skip-one tracking method.

The difference in $OTER$ between chase-the-reader and L2R-skip-one alone does not show how different the two tracking methods really are. To perform a more detailed analysis of the two, we group the tracking errors into subcategories based on reference transition types as seen in Table 2. Left-to-right is excluded from the table, since left-to-right and L2R-skip-one are very similar tracking strategies. The category labels should be interpreted as follows. The reference target word position was: not advanced ($c = p$), advanced by one ($c = p + 1$), advanced by more than one ($c > p + 1$), decreased by one or more ($c < p$). Relative within-category $OTER$ is calculated as the number of off-track ART speech events within the category divided by the number of reference speech events in the category. Absolute within-category $OTER$ is calculated as the number of off-track speech events within the category divided by the total number of reference speech events. ART insertions are the number of observed insertion errors made by the reading tutor. The difference in the number of insertion errors for the two tracking strategies is due to the fact that more insertion errors are masked when using L2R-skip-one.

Table 2: Tracking errors grouped wrt. reference transition. c is the current reference position and p is the previous.

	Chase-the-reader	
	relative within-category $OTER$	absolute within-category $OTER$
$c = p$	385/783 $= 49.2\%$	3.4%
$c = p + 1$	1177/9472 $= 12.4\%$	10.4%
$c > p + 1$	223/571 $= 39.1\%$	2.0%
$c < p$	317/509 $= 62.3\%$	2.8%
ART ins.	3887/– = –	34.3%
	L2R-skip-one	
	relative within-category $OTER$	absolute within-category $OTER$
$c = p$	1294/3460 $= 37.4\%$	36.0%
$c = p + 1$	919/7322 $= 12.6\%$	9.4%
$c > p + 1$	108/553 $= 19.5\%$	1.2%
$c < p$	0/0 = –	0.0%
ART ins.	2954/– = –	26.1%

The numbers show that a big difference between the two tracking strategies is the difference in the number of current speech events aligned to the same target word position as the previous speech event; this number is higher for L2R-skip-one (3460) than for chase-the-reader (783). This makes sense as regressions are not tracked when using the L2R-skip-one method, thus leaving the target word position unchanged. It is also evident that even though L2R-skip-one relatively is much better for each category (except for $c = p + 1$), globally it's only slightly better. This is due to the increased number of speech events in the $c = p$ category in combination with a high relative within-category observed tracking error rate (37.4%).

It is important to note that all presented results are calculated for an off-line system. This has two implications. The first is that any difference in reader behavior that would result from using the presented tracking methods to provide feedback in an on-line system are not expressed in the results. The extend of this effect on the results is unknown. The second is that we need to rely on unstable partial hypotheses if we are doing on-line tracking. Since a recognized partial hypothesis will not always be successive word prefixes of the final hypothesis, tracking errors are introduced which are not expressed in the current results. The tracking error rates are therefore expected to be higher for on-line tracking. A number of partial hypotheses (3855), generated using chase-the-reader tracking, are analyzed to get an impression of the extend of this effect. It is found that of all the partial hypotheses, 35.1% of them differ from the final hypothesis. This difference can be lowered to 15.6% if we disregard the last word of each partial hypothesis.

5. Conclusions

In this paper we have presented a new framework for evaluating tracking accuracy of an automatic reading tutor along with a way of visualizing tracking. The observed tracking error rates ($OTER$) when trying out three different tracking methods in an automatic reading tutor were estimated using this framework. $OTER$ for the three methods are: chase-the-kid: 53%, left-to-right: 56%, and L2R-skip-one: 47%

The presented experiment has been done offline. This means that any differences in reader behavior that would result from using the presented tracking methods to provide real-time feedback will not be observed. Tracking errors introduced from using partial hypotheses in an on-line reading tutor are also not observed in the off-line results. To use the presented evaluation framework for evaluating an on-line reading tutor we need to first record the reader's utterances when using the on-line reading tutor and then use these utterances as test-utterances in the presented framework.

6. Acknowledgements

The research reported here was supported by Oticon Fonden [8] and by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080628. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education. We also thank the educators, students, and LISTENers who helped generate our data.

7. References

[1] Pedersen, J. S., "User Centred Design Of a Multimodal Reading Training System for Dyslexics", Ph.D. thesis, Aalborg University, 2009.

[2] Witt, S. M., "Use of Speech Recognition in Computer-assisted Language Learning", Ph.D. thesis, University of Cambridge, 1999.

[3] Banerjee, S., Beck, J., and Mostow, J., "Evaluating the Effect of Predicting Oral Reading Miscues", European Conference on Speech Communication and Technology (Interspeech), Geneva, Switzerland, 2003, pp. 3165–3168.

[4] Rasmussen, M. H., Tan, Z.-H., Lindberg, B., and Jensen, S. H., "A System for Detecting Miscues in Dyslexic Read Speech", European Conference on Speech Communication and Technology (Interspeech), Brighton, U.K., 2009, pp. 1467–1470.

[5] Duchateau, J., Kong, Y. O., Cleuren, L., Latacz, L., Roelens, J., Samir, A., Demuynck, K., Ghesquière, P., Werner Verhelst, W., and hamme, H. V., "Developing a Reading Tutor: Design and Evaluation of Dedicated Speech Recognition and Synthesis Modules", Speech Communication, volume 51, No. 10, October 2009, pp. 985–994.

[6] Banerjee, S., Mostow, J., Beck, J., and Tam, W., "Improving Language Models by Learning from Speech Recognition Errors in a Reading Tutor that Listens", Second International Conference on Applied Artificial Intelligence, Fort Panhala, Kolhapur, India, December 2003, pp. 187–193.

[7] Mostow, J., Roth, S., Hauptmann, A. G., and Kane, M., "A Prototype Reading Coach that Listens", Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), American Association for Artificial Intelligence, Seattle, WA, August 1994, pp. 785–792.

[8] Oticon Fonden, On-line: http://www.oticonfonden.dk, accessed on 29 Mar 2011.