Efficient Speech Transcription Through Respeaking

Matthias Sperber^{1,2,3}, *Graham Neubig*^{2}, *Christian Fügen*^{3}, *Satoshi Nakamura*^{2}, *Alex Waibel*^{1}

^{1}Karlsruhe Institute of Technology, Institute for Anthropomatics, Germany
^{2}Nara Institute of Science and Technology, AHC Laboratory, Japan
^{3}Mobile Technologies GmbH, Germany

first.last@kit.edu, {neubig,s-nakamura}@is.naist.jp, first.last@jibbigo.com

Abstract

We propose a method for efficient off-line speech transcription through respeaking. Speech is segmented into smaller utterances using an initial automatic transcript. Respeaking is performed segment by segment, while confidence filtering helps save supervision effort. We conduct detailed experiments comparing speaking vs. typing, sequential vs. confidence-ordered supervision, and examine the effect of the respeaking word error rate on correction efficiency. Our results demonstrate that the proposed method can not only outperform typing in terms of correction efficiency, but is also much less demanding for the respeakers than traditional respeaking methods, consequently helping to keep costs down.

Index Terms: speech recognition, correction, segmentation

1. Introduction

While the transcription of speech is a necessity for an increasing number of applications, often quality requirements are high and cannot be met even by state-of-the-art automatic speech recognition (ASR) technology. On the other hand, manual transcription is very expensive. The combination of a human's skills and speech technology can help ameliorate these problems by providing a good trade-off between transcription quality and cost. One method for creating faithful transcripts effectively is respeaking, in which a second speaker repeats and records the same words uttered by the original speaker. Advantages of respeaking are that the respeaker's voice can be recorded in a controlled setting which leads to a better ASR performance than that of recognizing the original speech, and a higher speed compared to typing.

In this paper, we propose a method to enable more efficient off-line speech transcription through respeaking. In contrast to traditional real-time respeaking methods, in which respeakers speak all speech to be recognized, our method segments the input speech into short utterances, and selects only some of the utterances to be respoken based on confidence measure estimates. Combining the original speaker and respeaker's hypotheses further improves the results. The presented approach is "friendly" to the respeaker, as he no longer has to hurry to keep up with the original speaker. Consequently, the resulting transcripts stay closer to the original wording and respeaking requires less training than with traditional methods. We present results from experiments by two respeakers, as well as a simulation. The results demonstrate that the method is fast and more efficient than transcribing via typing or traditional real-time respeaking techniques, provided the speaker has a reasonable performance in terms of recognition word error rate (WER).

2. Relation to Prior Work

Speech transcription through respeaking has been investigated in a number of studies, mostly focusing on error recovery for speech interfaces, or real-time closed captioning of broadcasts by a dedicated respeaker. Suhm and Waibel [1, 2] study the former scenario and show that switching input modalities to recover from ASR errors is superior to simply repeating one's utterance. The approach to have a second person respeak misrecognized utterances is an example of an altered modality, and in fact is now predominantly used in live subtitling [3], as typing is too slow and stenography too expensive. Respeakers trained to create television subtitles in real-time are reported to achieve error rates of less than 4% [4], although results are often a summary rather than a faithful transcript [3]. To further eliminate errors, script recognition [5] or post-correction [6] can be used. Recently, a combined approach of respeaking and typing was introduced for correcting an automatic transcript in real-time by one person [4]. However, this method assumes a WER of only 10% for the initial transcript, and requires highly skilled respeakers and frequent breaks. In contrast, the proposed method requires little or no training, can be executed with fewer breaks, and allows the respeakers to stay closer to the original wording. To the authors' best knowledge, this study is the first to investigate respeaking methods that are optimized for the off-line transcription scenario.

Other related work uses word confidences for efficient typed transcriptions [7], and respeaking hypothesis combination for speech interfaces [8]. In contrast to the latter, our hypothesis combination approach relies on phonetic information rather than confusion networks. Also, exploiting repair context [9, 10] and adaptation towards the original speaker [11] has been proposed, these methods are applicable to our scenario as well.

3. Proposed Method

Our goal is to develop a method for improving the quality of a speech transcript efficiently through respeaking. We define efficiency as word error rate reduction achieved in a certain amount of supervision time. Our approach comprises a sequence of steps, summarized as follows. As a preparative step, the respeaker undergoes an enrollment procedure. For a given speech that should be transcribed: (1) An initial ASR transcript is created. (2) Using this transcript, the speech is segmented into short, sentence-like units. (3) Each segment is assigned a segment confidence. (4) The respeaker speaks each segment. (5) The recognition hypotheses from original speaker and respeaker are combined to improve the results.

3.1. Preparative Step: Enrollment

We assume that respeaking is to be performed by the same, known speaker(s) repeatedly, which justifies training speaker-adapted acoustic models by an enrollment procedure for each speaker. The speaker records training material for supervised model adaptation, preferably in the same recording environment in which the respeaking is to take place. In this study, we use unconstrained and constrained maximum likelihood linear regression [12, 13].

3.2. Step 1: Initial Recognition

As a guide for the respeaker, and to enable the succeeding steps, we use ASR to create an initial transcript from the original speech. We use confusion networks for decoding to estimate reliable confidence scores (see section 3.4).

3.3. Step 2: Segmentation

Next, the speech is divided into smaller segments. Segmentation is an important part of our approach, as it not only makes the actual respeaking easier, but also allows skipping segments via confidence filtering and simplifies navigation. Note that, as a limitation, segment-by-segment correction produces some overhead for each segment due to the delay that comes from the respeaker having to listen ahead before actually speaking. A suitable segmentation should be long enough to reduce this delay and ensure good recognition accuracy, but not so long that the respeaker has to speak more than is necessary. Segment breaks should also appear at natural positions in the sentence, as an awkward segmentation might be confusing and produce suboptimal language model scores when recognizing the respoken utterance.

We adopt a log-linear segmentation model proposed by Matusov et al. [14]. As features we chose language model scores for the inner segments and segment breaks, pause duration between segments, segment length and duration models, and a segmentation penalty. The estimation of length and duration models, as well as optimization of feature weights, was done using the manual segmentation of a selection of transcribed TED talks [15] into subtitles. These subtitles provide a reasonable, though not explicitly optimized, segment length for respeaking.

Advantages of this method are that it produces relatively natural segments, and that the segmentation penalty allows controlling the segment length by adjusting the feature weight. A drawback is that sometimes, due to a weak pause feature weight, segment breaks occur even when no prosodic break is present, which can make a word hard to understand, and produce ambiguity as to which segment a word should be respoken for. This could be improved for example by making prosodic breaks a hard requirement for segmentation.

3.4. Step 3: Segment Confidence Estimation

Next, we use confusion networks [16] to produce word confidences in the form of posterior probability estimates.[1] Estimating confidence measures is important because they allow us to identify segments with potentially high error rates. By first correcting these segments, either through respeaking or typing, we can reduce a larger number of errors in less time. Given word confidence scores, we define the segment confidence score as the arithmetic mean of the word posteriors. This provides an estimation of the word error rate, under the simplifying assumption that all errors are substitution errors.

3.5. Step 4: Respeaking

For respeaking, we define two supervision strategies. The first, more traditional strategy is *sequential* correction: Segments are corrected in temporal order, and every segment is presented to the respeaker regardless of its confidence. The second, proposed strategy is to make use of segment *confidences*: Segments are corrected in ascending order of confidence, and supervision can be aborted once a certain threshold is reached. The first strategy makes it easier for the respeaker to keep track of the speech's context, whereas the second strategy has the advantage of saving effort via the confidence filtering. Note that it would be easy and also reasonable to mix both strategies, i.e. using a sequential order but with confidence filtering; however, this would complicate interpretability of our results and is thus left for future work.

In practice, a respeaker would start listening to a segment, and start speaking while still listening. If the speaker notices that the original transcript is already correct, he would abort the recording and directly proceed to the next segment. This strategy of *skipping* segments that are already correct is effective both in saving time and increasing accuracy. Skipped segments have a correction effort roughly equal to their playback duration, while all other segments take longer, due to the inevitable delay between listening and speaking.

3.6. Step 5: Hypothesis Combination

An error analysis revealed that our respeakers were able to correct 60.7% of the original speakers' errors, but introduced 31.3% new errors. This surprisingly small overlap makes a strong case for using system combination techniques to combine both hypotheses, and hopefully have some errors cancel each other out. This can be done in a number of ways, but in this work we use ROVER [17], a method for combining one-best hypotheses that works even when the time alignment between the utterances is not consistent. Two hypotheses are combined based on their word alignment, and the word with the highest confidence is chosen at each position. In our experiments ROVER produced unstable alignments in some cases, so we propose two improvements: (1) We establish word alignments based on the phonetic similarity of the words rather than word identity. Specifically, the similarity between two words is derived from the edit distance between their phone sequences. This helps align corresponding recognized word that are different in spelling, but similar in pronunciation. (2) Moreover, we insert filler words recognized by the ASR as null links into the alignment graph. By doing this, we improve the alignment of deletion errors in which a word was misrecognized as a filler. These extensions yielded an additional decrease in WER of 3% and 2% relative, respectively.

4. Experiments

Our experiments are based on data provided by TED [15], a platform for talks on technology, entertainment, and design. The talks have a length between 5 and 20 minutes, are presented by skilled speakers, and recorded at good quality, although occasional non-speech events such as music are a disturbing factor for ASR.

We used a fairly standard decoding setup for our experi-

[1] Confusion networks outperformed feature combinations and estimating posterior probabilities from the word lattice in preliminary experiments.

ments. Acoustic models based on MFCC with 3000 codebooks, 64 Gaussians, and a 42-dimensional feature vector were trained on various audio sources, including a TED training set. We used a 4-gram language model tuned to minimize the perplexity on a held-out TED data set. The vocabulary size was 180k. Decoding was performed by the IBIS decoder [18].

For fast transcription, a good user interface is of critical importance, so we developed an efficient tool specifically for this task. Data was collected by 2 respeakers for the evaluation data[2]. One speaker was a native English speaker, one was a foreign speaker, both could be categorized as average speakers and above-average typists. The respeakers did not undergo any training procedure. The enrollment text had 7,416 words, and the evaluation data consisted of two 15-minute TED talks that were supervised fully and sequentially, and 5 talks that were supervised only partially (between 2 and 3 minutes per talk) and in order of segment confidence. These TED talks were not included in the training material, and contained only a minimal number of non-speech events. All segments were respoken and typed, in alternating order to remove bias. For evaluation, we measured the time spent respeaking or typing for every segment.

4.1. Effect of Using Confidence Scores

The proposed segment confidence scores are effective, as confirmed by a simple theoretical experiment that was carried out on a transcript with 28% WER, and 73.9% segment error rate: Ordering the segments by their assigned confidences and correcting them one by one yields a WER of 10% (5%) once 40.5% (60.6%) of all segments are corrected. This is a noticeable improvement over having to correct 64.3% (82.1%) of all segments to achieve the same WER when proceeding in random order.

4.2. Word Error Rates

Table 1 shows resulting word error rates for our experiments. It can be seen that speaker adaptation through enrollment is a crucial step of our method, leading to a 3–5% decrease in WER. Also, even our simple one-best hypothesis combination yielded good results with a 1.8% decrease, although skipping over correct segments weakened the positive effect. The typing WER was 5.7%, which is perhaps surprising. Analysis showed that about 1.8% of that was due to segmentation issues, in which the lack of a prosodic break complicated understanding and caused ambiguity as to which segment a word belongs to. We conclude that a better segmentation strategy is crucial to improve the method. The remaining 3.9% WER was mostly due to ambiguous reference transcripts, e.g. caused by speaking mistakes of the original speaker.

4.3. Correction Effort

Analysis of the correction time revealed a speaking rate of 189 wpm (words per minute) for the original speakers, and 131 wpm for the respeakers. The delay at the beginning of each recording, caused by having to listen ahead before respeaking, was 1.2 seconds on average and reduced the effective speaking rate to 100 wpm. There was significant additional overhead due to having to listen to a segment again when something was difficult to understand. Note that some of that overhead was due to segmentation issues and might thus be eliminated by a better segmentation. On the other hand, time was saved when the orig-

	DEV	EVAL	
	all	all	skip
original speaker	31.4	21.7	
respeaker	22.4	19.8	15.8
respeaker adapted	16.9	14.9	12.3
hypothesis combination	**15.9**	13.1	**11.9**
keyboard	-	5.7	

Table 1: *Recognition word error rates [%].* Listed for original speakers, respeakers with and without speaker adaptation, combined systems, and keyboard correction. Results differed when respeaking all segments, compared to skipping over segments that were already correct.

	Sequential	Confidence
Keyboard	61 wpm	58 wpm
Respeaking	97 wpm	83 wpm

Table 2: *Effective speaking and typing rates.* Numbers include time needed to record or listen to a segment again, and saving time by skipping correction for segments that were already correct.

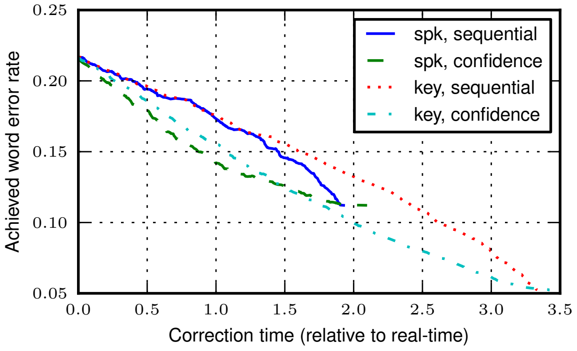

Figure 1: *Efficiency curves for native speaker.* Using confidences is clearly superior to sequential correction. When time is limited to less than 1.5×real-time, respeaking (spk) yields better efficiency than typing (key) when using confidences.

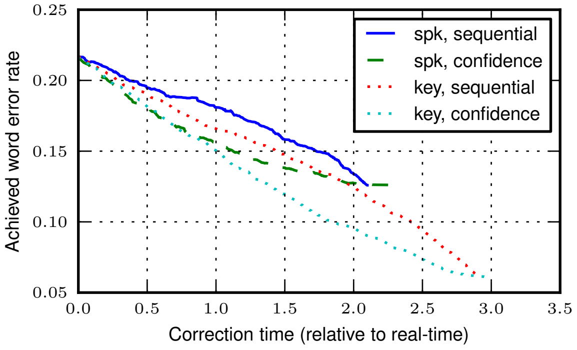

Figure 2: *Efficiency curves for foreign speaker.* Again, confidences are beneficial, but typing is consistently more efficient than respeaking.

[2] We had three respeakers for the development data.

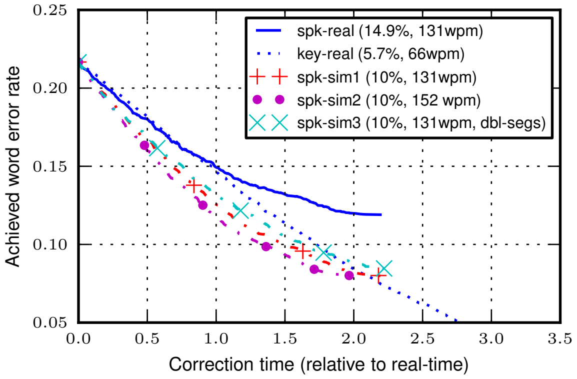

Figure 3: *Efficiency curves for confidence-based correction by real and simulated speakers.* WER and speaking rates in parentheses. Simulating (spk-sim) lower WER and higher speed improved efficiency over real speakers (spk-real), doubling the segment length (dbl_segs) degraded performance. Keyboard correction efficiency (key-real) is displayed for comparison.

inal transcript was already correct and the respeaking could be aborted early. Table 2 shows the speaking and typing rate when including all these factors. It can be seen that speaking was significantly faster than typing, though far from the original rate. Also, for respeaking, proceeding in the order of lowest confidences decreased the speaking rate significantly, as the lack of context information made it harder to understand the speech.

We also observed that a lower initial segment word error rate reduced only the typing effort, not the respeaking effort. This makes sense because for typing, one only needed to retype the incorrect words and could easily skip over the remaining words, whereas respeaking required supervision of the complete segment. In particular, segments with a WER of 5% or less needed less time for typing than for respeaking, on average. This observation may be used to give suggestions to the user as to whether a segment should be typed or respoken, based on the confidence score.

4.4. Analysis of Efficiency

Figures 1–2 allow us to analyze the efficiency of our approach, namely the achieved WER reduction compared to the overall supervision time, over various scenarios. Figure 1 shows that for the native speaker, choosing segments by confidence could achieve a lower WER in an equivalent amount of time when supervising only part of the speech, despite the slower speaking rate. Sequential order was faster when supervising the whole speech, an intuitive result as confidences are not useful in this case. The diagram shows that the native speaker had better results with respeaking than typing when spending less than 1.5 and 2.5 times real-time for supervision in confidence- and sequential order, respectively. In contrast, figure 2 shows that the foreign speaker, whose WER over the different scenarios was worse by about 12% relative on average, was consistently more efficient by typing than respeaking.

4.5. Simulating Altered Speaker Attributes

The previous section reveals a strong dependency of the transcription efficiency on the particular speaker, so it would be interesting to understand in what way this efficiency is affected by different speaker attributes. We summarize these attributes as recognition accuracy and speaking rate, and perform simulations in which both attributes are artificially altered. In particular, we use the combined results of both respeakers as a baseline, and improve the recognition accuracy by evenly removing errors until the desired WER is reached. The speaking rate is increased by multiplying the timestamps measured during our experiments by a suitable factor. We use a slightly pessimistic WER of 10% WER, considering that respeaking WERs below 4% are reported in [4], and that the best TED speaker in our test set achieved 8% WER even without speaker adaptation. We choose 152 wpm speaking rate (80% of the original speaking rate) as a similarly pessimistic value when compared to results in [3] that report respeaking rates close to the original rates, at least when accounting for the punctuation that the respeakers had to speak as well.

Figure 3 shows that changing only the *accuracy* attribute to 10% WER (spk-sim1) resulted in a noticeable efficiency gain. Next, we additionally increased the *speaking rate* attribute to 152 wpm (spk-sim2), which again resulted in a noticeable gain. This indicates that a skillful speaker could likely achieve further gains over typing. Finally, we simulated doubling the segment length (spk-sim3), and thus removing some of the overhead due to the delay between listening and speaking. The supervision time for the doubled segment is estimated by adding the times as originally measured, then only once subtracting the average overhead determined earlier. Perhaps surprisingly, doubling the segment length caused a drop in performance as compared to spk-sim1, since now the number of completely correct segments that can be skipped decreases. This indicates that the chosen segment length is already roughly a good value, despite not being explicitly optimized. Note that the final WERs in the chart are lower than the denoted recognition WERs, due to the effect of system combination and skipping correct segments.

5. Conclusion

We proposed a method to enable efficient speech transcription through respeaking via a combination of various techniques. In our experiments, respeakers were able to reduce the initial word error rate by 45% relative in about twice real-time. Consistently with [1], we showed that the efficiency strongly depends on the speaker's recognition rate, with respeaking outperforming typing for good speakers. We further demonstrated the potential of using segment confidences and hypothesis combination to increase efficiency, and showed that it depends on the particular segments whether respeaking or typing is a better choice.

In the future, we would like to use a strategy of proceeding sequentially, while using confidence filtering at the same time. An important point is the improvement of the segmentation. Hard requirement of a prosodic break, as well as an explicit optimization in terms of correction effort seem promising. Finally, results may be improved by using automatic respeaking region detection as in [19], a more sophisticated hypothesis combination strategy, better ASR setup, and various adaptation strategies.

Acknowledgments. The research leading to these results was supported in part by an interACT student exchange scholarship, and has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n 287658 – 'Bridges Across the Language Divide' (EU-BRIDGE).

6. References

[1] B. Suhm, B. Myers, and A. Waibel, “Multimodal error correction for speech user interfaces,” *ACM Transactions on Computer-Human Interaction*, vol. 8, no. 1, pp. 60–98, 2001.

[2] A. Waibel, B. Suhm, and A. E. McNair, “Method and Apparatus for Correcting and Repairing Machine-Transcribed Input Using Independent or Cross-Modal Secondary Input,” U.S. Patent 5,855,000, 1998.

[3] P. Romero-Fresco, “More haste less speed: Edited versus verbatim respoken subtitles,” *Vigo International Journal of Applied Linguistics*, vol. 6, pp. 109–134, 2009.

[4] A. Prazak, Z. Loose, J. Trmal, J. V. Psutka, and J. Psutka, “Novel Approach to Live Captioning Through Re-speaking: Tailoring Speech Recognition to Re-speakers Needs,” in *Interspeech*, 2012.

[5] M. J. Evans, “Speech Recognition in Assisted and Live Subtitling for Television,” *BBC Research & Development White Paper*, 2003.

[6] S. Homma, A. Kobayashi, T. Oku, S. Sato, T. Imai, and T. Takagi, “New Real-Time Closed-Captioning System for Japanese Broadcast News Programs,” in *11th International Conference on Computers Helping People with Special Needs*, 2008, pp. 651–654.

[7] I. Sanchez-Cortina, N. Serrano, A. Sanchis, and A. Juan, “A prototype for Interactive Speech Transcription Balancing Error and Supervision Effort,” *International Conference on Intelligent User Interfaces*, pp. 325–326, 2012.

[8] K. Vertanen and P. O. Kristensson, “Getting it Right the Second Time: Recognition of Spoken Corrections,” in *Workshop on Spoken Language Technology (SLT)*, 2010, pp. 289–294.

[9] B. Suhm and A. Waibel, “Exploiting repair context in interactive error recovery,” in *Eurospeech*, 1997, pp. 1659–1662.

[10] A. Waibel and A. E. McNair, “Locating and Correcting Erroneously Recognized Portions of Utterances by Rescoring Based on Two N-Best Lists,” U.S. Patent 5,712,957, 1998.

[11] S. Luz, M. Masoodian, and B. Rogers, “Interactive visualisation techniques for dynamic speech transcription, correction and training,” in *International Conference on Human-Computer Interaction Design Centered HCI (CHINZ)*, 2008, pp. 9–16.

[12] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech & Language*, vol. 9, pp. 171–185, 1995.

[13] M. J. F. Gales, “Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition,” *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[14] E. Matusov, A. Mauser, and H. Ney, “Automatic Sentence Segmentation and Punctuation Prediction for Spoken Language Translation,” in *International Workshop on Spoken Language Translation*, 2006, pp. 158–165.

[15] (2012) Ted: talks on technology, entertainment, and design. [Online]. Available: www.ted.com

[16] L. Mangu, E. Brill, and A. Stolcke, “Finding consensus in speech recognition: word error minimization and other applications of confusion networks,” *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[17] J. G. Fiscus, “A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER),” in *Automatic Speech Recognition and Understanding Workshop*, 1997, pp. 347–354.

[18] H. Soltau, F. Metze, C. Fügen, and A. Waibel, “A One-Pass Decoder Based on Polymorphic Linguistic Context Assignment,” in *Automatic Speech Recognition and Understanding Workshop*, 2001, pp. 214–217.

[19] K. Vertanen and P. O. Kristensson, “Automatic selection of recognition errors by respeaking the intended text,” in *Workshop on Automatic Speech Recognition & Understanding (ASRU)*, Dec. 2009, pp. 130–135.