Automatic Selection of Recognition Errors by Respeaking the Intended Text

Keith Vertanen, Per Ola Kristensson

Cavendish Laboratory, University of Cambridge
JJ Thomson Avenue, Cambridge CB3 0HE, UK
{kv227,pok21}@cam.ac.uk

***Abstract*—We investigate how to automatically align spoken corrections with an initial speech recognition result. Such automatic alignment would enable one-step voice-only correction in which users simply respeak their intended text. We present three new models for automatically aligning corrections: a 1-best model, a word confusion network model, and a revision model. The revision model allows users to alter what they intended to write even when the initial recognition was completely correct. We evaluate our models with data gathered from two user studies. We show that providing just a single correct word of context dramatically improves alignment success from 65% to 84%. We find that a majority of users provide such context without being explicitly instructed to do so. We find that the revision model is superior when users modify words in their initial recognition, improving alignment success from 73% to 83%. We show how our models can easily incorporate prior information about correction location and we show that such information aids alignment success. Last, we observe that users speak their intended text faster and with fewer re-recordings than if they are forced to speak misrecognized text.**

I. Introduction

In speech recognition there are situations in which it is necessary or desirable to correct misrecognitions by voice. For example, users with a repetitive strain injury (RSI) may want to avoid using the mouse and keyboard. Another example is mobile speech recognition. We previously found that about half of recognition errors made while dictating to a mobile device needed to be corrected by typing out the intended words [1]. Since precise motor actions are difficult while walking [2], a hands-free voice-only correction interface may be beneficial.

A common method of voice-only correction uses a two-step process. In the first step, users select a portion of the recognized text by voice (e.g. "select the *bat* sat"). Next, they speak their intended replacement text (e.g. "the *cat* sat"). In this paper we investigate an alternative one-step method, first proposed by McNair and Waibel [3]. Using this method, the user only speaks the intended replacement text with the location of the replacement being found automatically.

The one-step process promises to be faster and simpler for users. In addition, it may be more comfortable for users as it allows them to speak their intended text rather than text containing recognition errors (which may be ungrammatical or illogical). While our ultimate goal is to address the entire error correction process, in this paper we focus on the problem of correctly locating the error region. This is a critical first step in a complete one-step voice correction technique.

We propose and evaluate three new models for automatically aligning spoken corrections. We show that a model based on a word confusion network outperforms a model using only the 1-best recognition result. Furthermore, we develop a model that allows automatic alignment even if the user revises their initial recognition result by adding, changing or removing words. We validate our models in two user experiments. We show that, without explicit instruction, users tend to speak correctly recognized words surrounding an error. We also demonstrate that providing such correct context improves alignment success. In addition, we find that our users speak the intended text faster and with fewer re-recordings than if they are forced to speak the corresponding misrecognized text.

II. Automatic Alignment Models

We present three models for automatically aligning corrections and revisions. Each model creates a finite state grammar (FSG) based on the results from recognition on the full sentence. This grammar is used to determine the starting and ending indices of the correction or revision within the words of the original 1-best recognition result.

The FSG used for alignment is a set of states and the edges between those states (figure 1). An edge between two states specifies the word that must be spoken to traverse that edge and the probability for making that transition. We introduce a set of pseudo-words to track the starting and ending index positions found by recognition. These pseudo-words are denoted <0>, <1>, etc. These words are placed in the recognizer's dictionary with a pronunciation of the silence phone. The start and end index words are the main result of decoding using the grammar – we ignore the actual words recognized.

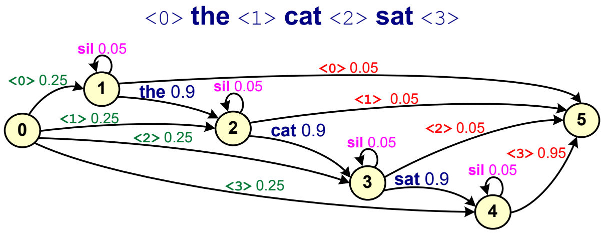


Fig. 1. A 1-best result (top) and the FSG (bottom) generated from it. The 1-best result has been annotated to show the location of the index pseudo-words. If recognition using the grammar resulted in the state sequence 0, 2, 3, 5, the selection would be from <1> to <2>, "cat".

978-1-4244-5479-2/09/$26.00 © 2009 IEEE

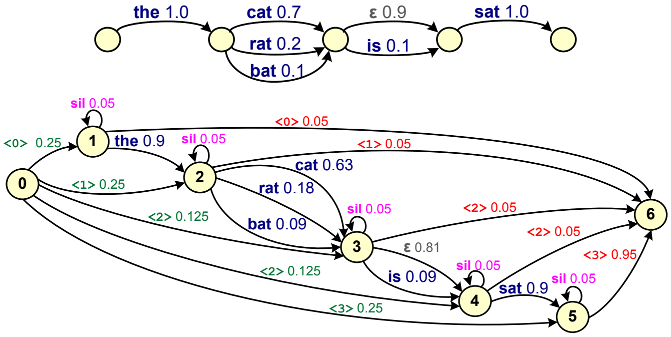


Fig. 2. A confusion network (top) and the FSG (bottom) generated from it.

All states in our grammars (except for the initial and final states) have an *end* and a *silence* probability. The end probability controls exit to the final state. The silence probability controls a self-loop that generates silence. After assigning probabilities to the end and silence edges, all remaining probability mass is used for word edges. The initial state has outgoing edges to each state except the final state. For now, we assume the edges from the initial state have equal probability.

A. 1-Best Model

Our simplest model for aligning corrections uses only the initial 1-best recognition result (denoted 1-BEST). This model allows alignment using any contiguous set of words in the 1-best result. Figure 1 shows an example grammar. Such a model was first suggested by McNair and Waibel [3] (although they used a bigram language model, not an FSG).

B. Confusion Network Model

The confusion network model (denoted CN) is based on a word confusion network [4] generated from the initial recognition. This model takes into account competing word alternatives for each word in the original recognition result. Figure 2 shows an example confusion network and the grammar generated from the network. This grammar has a state for each cluster in the confusion network. Edges between states in the grammar are added for each word hypothesis in the confusion network cluster, with the edge probability set based on a word's posterior probability in the confusion network.

Note that a confusion network cluster's most likely word may be a special "delete" word (denoted as ϵ). Such delete words do not have a corresponding word in the 1-best recognition result. In such cases, we divide the probability from the initial state to a particular starting index among the word at that index and any delete cluster states that follow it (e.g. the edges from state 0 to states 3 and 4 in figure 2).

The CN model also adds a *word smoothing* parameter. This parameter smooths the word posterior probabilities from the confusion network with a uniform distribution. The idea is to allow some of the less probable words from the confusion network to better compete with the original best words (since we expect some of these best words to be wrong). A smoothing value of zero uses the unaltered posterior probabilities. A smoothing value of one uses a completely uniform distribution.

C. Confusion Network + Unknown Word Model

This model (denoted CN+UNK) extends the confusion network model by allowing arbitrary word insertions, deletions, and substitutions. This model gives users the flexibility to change their mind, altering what was said in their initial utterance. For example, a user may wish to change the correctly recognized sentence "the cat sat" to "the *very fat* cat sat".

The CN+UNK model uses a set of unknown words. The pronunciation dictionary entry for each unknown word is a sequence of one or more garbage phones (`<unk1>` has one garbage phone, `<unk2>` has two garbage phones, and so on). The garbage phone was trained by replacing 10% of the words in our acoustic model training transcripts with an unknown word with the same number of phones as in the original word.

An example grammar is shown in figure 3. To enable arbitrary deletions, each state has an ϵ-transition added to the next state and uses a fixed *deletion* probability. If there already is an ϵ-transition, we add the probability to this transition.

To allow substitutions and insertions, we add a new substitution/insertion state for every cluster in the confusion network. Edges to this new state are added for 12 unknown words (having between 1–12 garbage phones). We set the probability of each unknown word edge according to how frequently pronunciations in the CMU dictionary had the corresponding number of phones. After generating an unknown word, edges go back to the original word state (an insertion) and to the next word state (a substitution). This model has a *substitution* and *insertion* probability. We assess these probabilities after the substitution/insertion state to keep the grammar more compact.

D. Prior Location Information

In real-world usage, we might have information regarding where the correction is likely to occur. For example, if the user is hovering the mouse pointer at a certain location in the recognition result, we might expect a correction near that location. It is easy to incorporate such knowledge into our model using the probabilities on the edges from the initial state and on the edges to the final state. While in most of our results we used an uninformative uniform prior, we also investigated the effect of having information about the correction location. In this work, we used a simple model that centered a Gaussian at the known starting and ending positions (i.e. our model used oracle knowledge).

E. Parameter Tuning

We tuned the model parameters using a set of utterances collected from three speakers (including one of the authors). Data collection followed the procedure to be described in section III (with the exception that we only collected guided corrections). Our development test set had 203 full sentences and 401 corrections. Besides our model parameters, we also tuned the language model scale factor that balances the importance of the grammar probabilities and the acoustic evidence.

We tuned each of the three models separately. We tuned to maximize the *alignment success*, which we define as the percentage of times our model exactly identified the correct

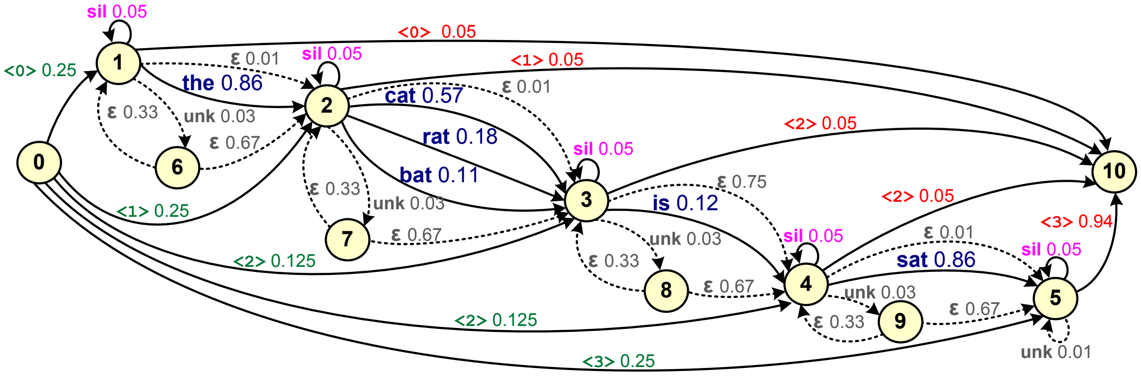


Fig. 3. Example grammar for the CN+UNK model. For clarity, only a single unknown word is shown. The dotted edges allow revisions involving arbitrary word insertions, substitutions, and deletions.

TABLE I
THE TUNED PARAMETERS USED FOR EACH MODEL.

Parameter	1-BEST	CN	CN+UNK
End	0.10	0.05	0.05
Silence	0.80	0.05	0.05
Word smoothing	-	0.00	0.10
Insertion	-	-	0.01
Substitution	-	-	0.02
Deletion	-	-	0.01
LM scale factor	8.00	24.00	12.00

starting and ending locations. We changed one parameter at a time, finding its optimum value, fixing its value, and then proceeding to the next parameter. Table I gives the best parameter values we found for each of the models.

III. USER EXPERIMENT 1

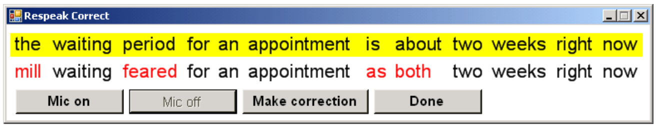
Our first user experiment had three goals. First, to investigate how (without explicit instruction) users would speak corrections. Second, to collect data of users speaking sentences and corrections using varying amounts of surrounding correct context. Third, to collect data of users revising correctly recognized sentences by changing the original text.

A. Recognition Setup

We used the CMU Sphinx recognizer and a US-English acoustic model trained on 211 hours of WSJ data. We used cross-word triphones and a 3-state left-to-right HMM topology. We parameterized audio into a 39-dimensional feature vector consisting of 13 Mel-frequency cepstral coefficients, deltas and delta deltas. We used 8K tied-states with each state having 16 continuous Gaussians with diagonal covariances. We used the CMU phone set without stress markings (39 phones plus silence) and the CMU pronunciation dictionary.

We trained a trigram language model using text from the CSR-III newswire corpus (222M words) and the most frequent 64K words. We trained the language model using interpolated modified Knesser-Ney smoothing and entropy-pruning [5].

We streamed audio sampled at 16 kHz to the recognizer as soon as the microphone was enabled. We performed cepstral mean normalization based on a prior window of audio. The recognizer was adapted to each participant's voice using maximum likelihood linear regression (MLLR). We adapted the model means using 7 regression classes.



the waiting period for an appointment is about two weeks right now
mill waiting feared for an appointment as both two weeks right now

Fig. 4. The user has read the top sentence and the recognition is shown below with word errors in red. During this unguided correction, the user must decide what utterance(s) to provide to correct the 3 error regions in the result.

We used PocketSphinx [6] and tuned it to provide near real-time recognition. During the user experiment, recognition took 2.7 × real-time on a 2 GHz laptop. For the offline experiments using FSGs, we used much wider decoding beam widths and utterance-wide cepstral mean normalization. The offline experiments took 0.3 × real-time on a 3 GHz computer.

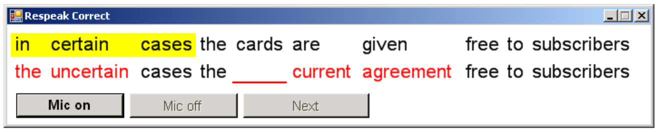
B. Materials and Participants

Eight speakers of North American English took part in the first experiment which lasted one hour. The speakers were different from those used for parameter tuning. Each participant recorded 40 sentences which we used for adaptation.

We presented users with sentences drawn equally from two WSJ test sets (WSJ0 si_et_05 and the SJM sentences from WSJ1 si_et_s2). We chose sentences with 4–18 words (mean 13). Using the 64K trigram language model, the sentences had a per-word perplexity of 270 and an OOV rate of 0.6%.

C. Procedure

Each participant was presented with a series of sentences. After the participant pressed a MIC ON button, a beep signaled recording was active. The participant then spoke the sentence and pressed a MIC OFF button. After a small recognition delay (11 s ± 12 s), a beep signaled recognition was complete. The recognition result was displayed below the reference text with word errors highlighted in red (figure 4). If the recognizer made a deletion error, the error was denoted by a red underlined empty space (figure 5).



in certain cases the cards are given free to subscribers

the uncertain cases the _____ current agreement free to subscribers

Fig. 5. The user is providing a guided correction for the first error region using one word of correct context on the right.

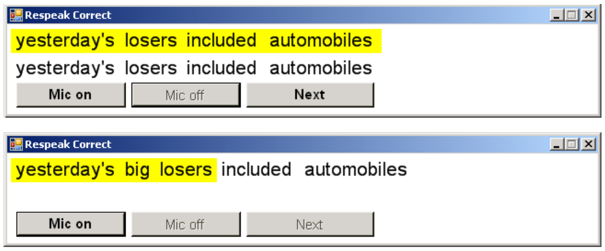
In part one, each participant received the same set of 10 sentences. For each sentence, the participant made zero or more *unguided corrections*. In the unguided corrections, the participant was told to provide utterances such that "an intelligent software program" could correct any recognition errors. The participant was not specifically instructed as to what words to speak and was free to use as many separate correction utterances as was deemed necessary. The participant could add a correction by pressing a MAKE CORRECTION button. The participant used the MIC ON and MIC OFF buttons to record corrections. No recognition took place on the correction audio. After completing any corrections, a DONE button brought the participant to the next sentence.

In part two, the participant was told what words to speak for each correction. The initial recognition proceeded as in part one, but after displaying the recognition result, the desired correction text was indicated by highlighting a portion of the reference sentence in yellow (figure 5).

Each highlighted section contained an *error region*. An error region is a contiguous number of words in a sentence that encapsulates one or more recognition errors. Error regions were created by first making a region for every word error. Each region was then merged with any adjacent regions that were separated by at most a single correct word.

For each error region, the participant was asked for 1–3 corrections. The correction used 0–2 words of correct left context and 0–2 words of correct right context. The number of corrections and the amount of context was chosen randomly, with the exception that corrections with no correct context were made twice as likely. The first two sentences in part two were designated as practice sentences and were excluded from analysis. Sentences were presented in random order.

In part two, if recognition was completely correct, the participant was prompted to record 1–5 revisions. The revisions made a substitution, insertion, or deletion of one or two words in the original reference sentence (figure 6). The set of allowed revisions were predetermined for each sentence to ensure the revisions were syntactically and semantically plausible.



yesterday's losers included automobiles

yesterday's losers included automobiles

yesterday's big losers included automobiles

Fig. 6. A correct recognition (top) followed by a revision (bottom).

TABLE II
SUCCESS ON CORRECTIONS WITH DIFFERENT TYPES OF CONTEXT.

Correct context	# utts	1-BEST	CN	CN+UNK
None	192	64.6%	67.2%	63.5%
Left	179	85.5%	85.5%	86.0%
Right	216	86.6%	89.4%	80.6%
Both	234	93.6%	93.2%	90.6%
Overall	821	83.2%	84.4%	80.6%

D. *Unguided Correction Results*

Participants provided unguided corrections for 80 sentences. The word error rate (WER) on these sentences was 18%. 57 of the sentences had at least one recognition error. In sentences with errors, there were an average of 1.5 error regions per sentence. Participants recorded on average 1.2 corrections per sentence. This indicates that users preferred to correct using longer utterances that contained several error regions.

We manually transcribed and annotated the utterances. We found that over half (54%) of the words in the utterances were correctly recognized words. Overall, 41% of the unguided corrections used no left or right context, 21% used left context, 20% used right context, and 18% used left and right context.

Six participants consistently used correct left or right context in their corrections. The remaining two participants consistently spoke only the words that were incorrectly recognized. It appears that even without instruction, users tend to use correct context.

E. *Guided Correction Results*

Participants completed 376 sentences in the second part of the study. Overall, the WER on these sentences was 17%. Participants provided a total of 821 guided corrections.

Table II shows each model's success at exactly determining the correction location (both the start and end position). Overall, the CN model without unknown words did the best. Without any correct context, finding the location was difficult. Providing either 1–2 words of left or right context helped considerably and roughly the same. As might be expected, using both left and right context was the most accurate.

Since these corrections matched a segment of the original sentence, the flexibility offered by the unknown words in the CN+UNK model was not necessary and we found it hurt alignment accuracy. Note that while words in the correction may not necessarily be in the original sentence's 1-best or confusion network result, the 1-BEST and CN models may still provide accurate alignments by relying on the recognizer preferring the same word errors during the alignment process.

As the number of context words was increased, alignment success improved (figure 7). Providing just a single word of context improved alignment success from 65% to 84% (averaged over all models).

We found that the type of recognition error influenced alignment success (table III). Corrections involving only substitution errors were the easiest to align. Corrections with one

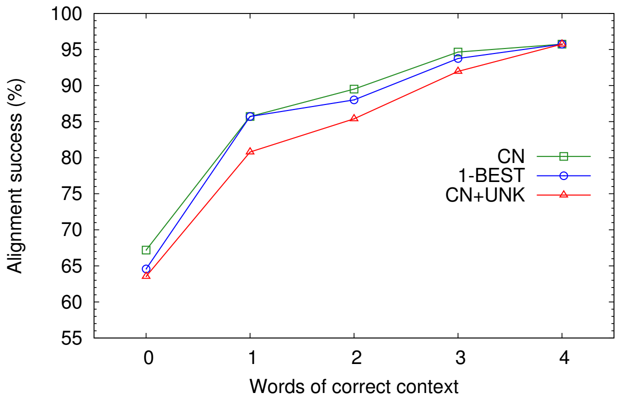


Fig. 7. Success of each model as the words of correct context was increased.

TABLE III
Success Depending on the Type of Error Being Corrected.

Recognition error type	1-Best	CN	CN+Unk
Substitution only	84.3%	84.8%	81.2%
$\geq$ 1 ins/del	81.8%	84.0%	79.9%
$\geq$ 1 ins/del, outside	77.6%	81.2%	75.8%

or more insertion or deletion errors were more difficult. Our data included examples of correcting insertion and deletion errors without context on the outside. For example, correcting "the *bat is* sat" to "the cat sat" by only saying "the cat". Such corrections were understandably hard and in practice users would likely provide additional context.

F. Revision Results

88 sentences were recognized completely correctly and we collected 292 revisions of these sentences. The revisions included the insertion, substitution or deletion of one or two words as compared to the reference text. The revisions always included at least one word of correct left and right context. The 1-Best model identified 73% of the revision locations, the CN identified 74%, and the CN+Unk identified 83%. Thus it appears that the CN+Unk model was effective at modeling the word changes present in the revisions.

G. Location Prior Results

We compared using a uniform distribution on the start and end locations versus using prior information to inform the alignment. We used knowledge of the actual start and end location to center a Gaussian distribution. We varied the variance and optionally randomly perturbed the mean one word position to the left or right of the actual starting/ending location. As shown in table IV, even a broad prior on the starting/ending location was able to improve alignment success. At least for broader variances, using a perturbed mean made little difference to alignment success.

IV. User Experiment 2

We conducted a second user experiment to quantify the difference in alignment success between the status quo correction approach based on speaking the erroneous text versus a method that enabled speaking the intended text. We also wanted to investigate the difference in human performance between reading and speaking the two types of text.

TABLE IV
Success of the CN Model Varying the Prior Distribution.

Start prior	End prior	σ	Success (exact μ)	Success (offset μ)
Uniform	Uniform	-	84.4%	-
Gaussian	Uniform	2	86.5%	86.1%
Uniform	Gaussian	2	86.1%	86.2%
Gaussian	Gaussian	2	86.9%	87.1%
Gaussian	Gaussian	1	89.0%	88.7%
Gaussian	Gaussian	0.5	93.4%	86.1%

A. Materials and Participants

Eight North American English speakers took part in a second study which lasted one hour. Participants read segments of 145 sentences (chosen at random) from the first study. These segments were located where a recognition error had occurred in the first study. The participants were only given the segment to be spoken and not any surrounding context.

B. Procedure

At the start of the user experiment, each participant read 40 adaptation utterances. These utterances were later used to create speaker-specific acoustic models for our offline experiments. In the user experiment, no recognition took place.

Each participant completed two conditions. In the Ref condition, the sentence segments were from the reference text. In the Rec condition, the segments were from the recognition result. For example, in the first study the sentence "the medical society can refer you" was misrecognized as "the medical society can *re for* you". Users provided corrections to this previous recognition by saying "can refer you" in the Ref condition and "can re for you" in the Rec condition. The order of the conditions was counterbalanced.

C. Alignment Success Results

We used the recognition results from the first study to construct grammars for each full sentence. We then performed recognition against the utterances collected in the second study. We used an acoustic model adapted to each participant.

Over all utterances (1160 per condition), alignment success was 97% in the Rec condition and 87% in the Ref condition (table V). On utterances with one or more words of context (888 per condition), alignment success was higher in both conditions and the difference between conditions was reduced (98% Rec versus 93% Ref). As in the first study, the CN model was the best when users spoke the reference text. The CN model also performed as well as the 1-Best model when users spoke the recognition text.

TABLE V
Success When Users Spoke the Reference Text (Ref) Versus the Recognition Result (Rec).

Context	Condition	1-Best	CN	CN+Unk
Overall	**Ref**	**85.6%**	**87.1%**	**85.3%**
None	Ref	65.1%	69.5%	64.3%
$\geq$ 1 word	Ref	92.3%	92.5%	91.8%
Overall	**Rec**	**97.2%**	**97.2%**	**95.7%**
None	Rec	95.6%	94.1%	90.4%
$\geq$ 1 word	Rec	97.8%	98.1%	97.3%

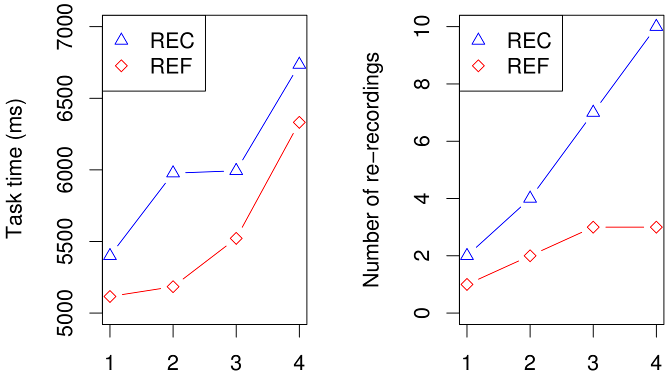


Fig. 8. Task completion time (left) and number of re-recordings (right) shown as a function of participant number, ranked by performance. The participants read either the reference text (Ref) or the recognition result (Rec).

D. *Human Performance Results*

A full investigation of the end-user benefits for this technique is out of scope for this paper. Nevertheless, here we provide some early quantitative *indicators* based on our second user experiment. Since we observed an asymmetrical skill-transfer, we analyzed performance only in the first condition encountered by each participant (as suggested by Poulton [7]).

We ranked each of the four participants in each condition using two measures of performance. The first was task completion time, which was the duration between when the reference text was first displayed and when the participant went to the next task. The second measure was the number of times a participant re-recorded a sentence segment. As shown in figure 8, at each corresponding ranking position, each participant who spoke the reference text had a lower task completion time and a lower number of re-recordings than his or her counterpart who spoke the recognition text.

These indicators show that our users found it easier to speak the reference text than the recognition text. However, we emphasize that these numbers are only indicators and a full user study is required to generalize these findings to the population. We also note that the full benefit of our technique is not demonstrated here since we only identified the error region and did not replace the misrecognized text. In an actual real-world task, users who spoke the misrecognized text would also have to respeak the intended text. With our automatic alignment models this second step is eliminated.

V. Discussion and Conclusions

We presented several new models for automatically aligning spoken corrections. The models were evaluated with data gathered from two user experiments. Among our models, we found that a model based on a confusion network performed the best. We showed that just a single word of context dramatically improved alignment success from 64% to 84%. We found that a majority of our users provided such context during corrections without being explicitly instructed to do so.

In addition, we presented an automatic alignment model that handles revisions as well as corrections. We showed that this model was superior to the other models when users added or subtracted words from their original sentence. This model improved revision alignment success from 73% to 83%.

We also provided some early indicators of human performance using our technique. We showed that our users spoke their intended text faster and with fewer re-recordings than if they spoke misrecognized text. Our data strengthens the hypothesis set forth by McNair and Waibel [3] that a voice-only correction mechanism similar to what we use in human-human communication is beneficial.

Last, we found that using a prior on the likely location of the error region improved success. Such priors can be obtained by letting users roughly indicate the error region by using a pointing device, such as a mouse, stylus, index finger or an eye-tracker. This may be especially important when a user's intended target sentence exists within a large body of text (as might occur when dictating a document or email).

Our next step is to build a complete correction interface that enables both automatic selection and subsequent correction using a single utterance. This will allow us to investigate the advantages offered by more natural voice-only correction.

Acknowledgment

The following applies to P.O.K. only: The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement number 220793.

References

[1] K. Vertanen and P. O. Kristensson, "Parakeet: A continuous speech recognition system for mobile touch-screen devices," in *Proc. International Conference on Intelligent User Interfaces*, 2009, pp. 237–246.

[2] A. Crossan, R. Murray-Smith, S. Brewster, J. Kelly, and B. Musizza, "Gait phase effects in mobile interaction," in *Extended abstracts of CHI*, 2005, pp. 1312–1315.

[3] A. E. McNair and A. Waibel, "Improving recognizer acceptance through robust, natural speech repair," in *Proc. International Conference on Spoken Language Processing*, 1994.

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

[5] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 270–274.

[6] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, 2006, pp. 185–188.

[7] E. C. Poulton, "Unwanted asymmetrical transfer effects with balanced experimental designs." *Psychological Bulletin*, vol. 66, no. 1, pp. 1–8, 1966.