Towards Fluid Speech-based Text Interaction

Keith Vertanen
Michigan Technological University
Houghton, MI, USA
vertanen@mtu.edu

CHI'17 Workshop on Designing Speech, Acoustic, and Multimodal Interactions, 7 May 2017, Denver, CO.

Abstract
Relying on just speech input to create, edit, and revise text can be challenging. While dictating the bulk of your text using speech recognition can be quick, subsequent editing steps are often best done using other input methods such as a keyboard or mouse. This position paper describes our efforts to make editing more fluid when speech is the primary or only input modality. We describe our approach to automatically inferring the location of a spoken correction or revision within the original speech recognition result. We describe our probabilistic merge model that combines information from the original recognition and the correction recognition to improve accuracy on the final correction. Lastly we describe how allowing users to provide spelling information can substantially improve accuracy.

Author Keywords
Speech recognition; error correction; text entry

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: Input devices and strategies

Introduction
Speech recognition accuracy has come along way in recent years [1]. It is now feasible to use your voice instead of your keyboard to complete many routine tasks such as sending

a short message or making a web search query. Often for short speech input tasks, a user can think of exactly their desired text and speak it fluidly to the computer. This can result in recognition at 100% accuracy and eliminate the need for any error correction.

For longer or more challenging texts, recognition errors may still occasionally occur. Further, text interactions can be more complicated and nuanced. For example we might be writing a tricky email to our boss or pitching our fabulous proposal idea in one-page or less. These types of writing tasks may require many iterations as the text morphs from humble beginnings into a polished gem. Iterative editing is typically well-supported with traditional input devices such as a keyboard and mouse. Using only your voice, rich text interaction is often tedious at best.

This position paper discusses how we might leverage probabilistic information from the speech recognition process, as well as information from the user, to achieve faster and more fluid text interaction. In particular, this paper outlines our past work on enabling one-step voice correction and revision of spoken input [5, 6], and how we might allow users to avoid recognition errors for difficult words [7].

Relationship to workshop

Text interaction at the desktop via keyboard/mouse or on a mobile device via touch are common in our work and home lives. But in the future, wearable and pervasive technologies mean we may increasingly want to interact without any physical input device. Speech represents a high bandwidth input method that most people can use in many situations.

Creating and consuming text offers advantages over flashier media types such video as it can be compactly represented, easily searched, and editing is more straightforward. Additionally for some people, due to a temporary or permanent motor disability, use of input methods such as a keyboard or mouse may be difficult or impossible. Thus creating more fluid interfaces based primarily on speech input makes text interaction more accessible.

Status quo voice editing

When editing or revising text, existing speech input interfaces such as Dragon NaturallySpeaking require a two-step process to make a change. For example, assume the user has already spoken the sentence “the cat sat” and that it was recognized correctly. Now assume the user decides to let the reader know the cat is really quite obese by revising the sentence to “the *really fat* cat sat”.

In conventional voice interfaces first the user would position the cursor by issuing a command such as “move left seven characters” or “insert before cat”. The user could then add the qualification by speaking “really fat”. Alternatively, the user might want to replace part of the text with new text, for example by saying “select cat” followed by “pretty kitty” to yield “the pretty kitty sat”.

One-step voice editing

A more fluid process might allow users to revise or correct errors in speech input in just one-step by uttering their intended new text such as “really fat”. The system would need to not only recognize the text of the correction, but also decide where to put it. This might be challenging as there is no explicit information in the spoken correction about where it fits into the text being revised.

One way a user could help the one-step correction process is by speaking some correct surrounding context to ground the revision in the existing text, e.g. “really fat *cat*”. Providing the correct surrounding context “cat” is quite natural; in a user study we found participants provided correct

surrounding context 54% of the time when correcting recognition errors without being given any instruction to do so [5].

Alignment model

The idea of a more seamless one-step correction process was first proposed by McNair and Wiebel [3]. Their algorithm aligns the spoken correction by using a bigram language model based on the 1-best recognition result. In [5] we extend this approach to leverage the richer probabilistic information available in a word confusion network [2].

For a correction or revision, the first recognition pass uses a finite state grammar (FSG) to determine the starting and ending indexes within an existing result. An edge between two FSG states specifies the word that must be spoken to traverse that edge and the probability for making that transition. We introduce the pseudo-words <0>, <1>, etc. to track the start and end index positions. These words have a pronunciation of the silence phone. Recognition of the pseudo-words is the main result of the first decoding pass.

From the original recognition's confusion network (Figure 1), we build a finite state grammar that takes into account competing word alternatives for each word in the recognition result (Figure 2). 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. Arbitrary word insertions, deletions, and substitutions are licensed via an unknown word model.

During recognition of the spoken correction, the decoder searches for the best path through the FSG from the initial state (0 in Figure 2) to the final state (10 in Figure 2). The edges taken from the initial state and to the final state are

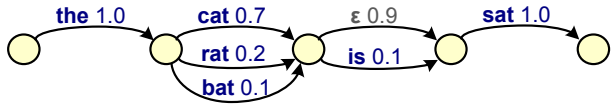

Figure 1: Word confusion network for "the cat sat". ϵ denotes the hypothesis that no word was spoken.

used to determine the most likely alignment of the correction with the original result.

Automatic alignment results

In a user study, participants spoke sentences that were recognized using CMU Sphinx. In cases in which a recognition error was made, users spoke corrections that contained the error plus 0–2 words of correct left context and 0–2 words of correct right context. Alignment success was 64% with no correct context, 86% with left context, 80% with right context, and 91% with both left and right context.

In cases when no recognition errors were made, we had users speak a number of pre-planned revisions. Revisions consisted of the insertion, substitution, or deletion of one or two words compared to the reference text. Revisions always included at least one word of correct left and right context. Alignment success was 83%. This showed the unknown words in the FSG allowed the model to handle words not seen in the original recognition.

Thus far we assumed no knowledge of a correction's location within the sentence. Thus the edges from the initial node and to the final node were uniformly weighted. In some cases, the user may be able to provide additional information about the start and end locations. This could be via an explicit action such as a touchscreen swipe, or via implicit information such as where the user is looking as reported by an eye-tracker.

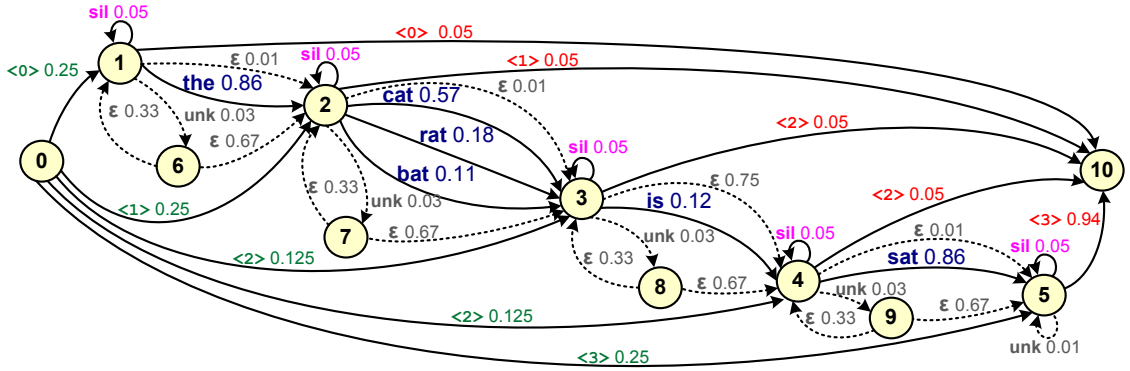

Figure 2: Example finite state grammar used to align a correction or revision with the original recognition result. The dotted edges allow revisions that contain arbitrary word insertions, substitutions, and deletions.

We simulated having such approximate location information by weighting the initial and final edges using Gaussians with different variances. This improved alignment success by 2-9% depending on the variance. This shows the model has the potential to use even approximate user guidance to improve alignment accuracy.

Improving recognition of corrections
In addition to inferring where to put a revision or correction, we also need to correctly recognize the content of the edit. In [6] we looked at ways to improve recognition accuracy of the correction utterances we collected in [5].

Improvements to the front-end processing, acoustic modeling, and decoder parameters reduced the word error rate (WER) on the corrections from 55% to 31%. Assuming the location of the correction was known within the surrounding sentence, applying the language model context to the left and right of the correction during the recognizer's search further reduced WER to 25%.

Merge model
Performing recognition just on the correction utterance in isolation is ignoring potentially useful information from the original recognition of the entire sentence. In [6] we also developed a *merge model* that combines information from multiple confusion networks.

The first confusion network is the spoken revision or correction. The second confusion network is cut out from the original sentence confusion network result based on the location of the edit. The cut out section is further adjusted based on which words were deemed likely to be correct surrounding context words based on their posterior confusion network probabilities. The merge model then searches for a joint path through the two confusion networks that has the highest probability.

The merge model reduced WER on the correction from 25% to 23%. If oracle knowledge was available about what words in the utterance were correct surrounding context, WER was further reduced to 21%.

Speak and spell

Some words such as proper names or uncommon words can be difficult to recognize. If voice is the only input modality available, spelling can provide additional signal to the recognizer. If users can anticipate problematic words, they may even be able to avoid errors in the first place by providing spelling in their initial utterance. The recognizer could handle this by placing spelled variants alongside the normal versions in its pronunciation dictionary.

In [7] we investigated the effectiveness of seven different ways users might provide spelling as additional information to a speech recognizer:

- **Word** – The word pronounced normally: “cat”.
- **Spelling** – The spelling of a word: “C A T”.
- **Word + spelling** – The word followed by its spelling: “cat C A T”.
- **Word + spelling + word** – The word before and after the spelling: “cat C A T cat”.
- **Phonetic** – The military phonetic-spelling of a word: “charlie alpha tango”.
- **Word + phonetic** – The word followed by its phonetic-spelling: “cat charlie alpha tango”.
- **Word + phonetic + word** – The word before and after its phonetic-spelling: “cat charlie alpha tango cat”.

We used Amazon Mechanical Turk to collect audio samples of people speaking and spelling words in all the different ways. In total we collected 2,793 utterances from ten unique workers.

We performed isolated word recognition from a 5K vocabulary that used no language model (all words were equally probable). The word utterance with no spelling were recognized at a WER of 50%. Spelling the word reduced WER substantially to 13%. Both speaking and spelling a word reduced WER to 5%. Finally the phonetic spelling variants reduced WER to 1% or lower. Thus even with no language model prior, spelling can offer quite accurate recognition.

We found all seven variants could be put into the recognizer’s pronunciation dictionary simultaneously with negligible accuracy reduction for normal word recognition. Thus it appears feasible to offer spelled variants during the initial dictation phase and not just as a special correction mode.

Future work

Our previous work has only looked at the two problems of automatic alignment [5] and recognition of corrections [6] in isolation. It remains to be seen how effective the approaches are in tandem as obviously mistakes in the alignment will influence the ability to get the correction right.

We have only explored alignment and correction within single sentence utterances. Providing robust performance in larger bodies of text would be more difficult and may require more sophisticated models and/or additional user signal.

Thus far we have assumed the system has some way to know whether an utterance is a revision or new text to be appended. A simple solution would be to instruct users to prefix a revision with a phonetically distinct keyword. But it would be more natural if the system could infer the user’s intent. This might be possible using information about the length of the utterance, the overlap in existing text, or where the user is looking (e.g. via an eye-tracker). When correcting errors, we might be able to use the hyperarticulate speech indicative of correction episodes to help differentiate corrections from new text. In past work [4] we found users markedly changed their speech during error corrections.

The results on correction/revision alignment and recognition

were done with offline experiments on recorded utterances. Our experiments used the CMU Sphinx recognizer trained using maximum likelihood, Gaussian mixture models, and modest amounts of WSJ training data (211 hours). User performance and satisfaction in a real interactive system using state-of-the-art recognition would be interesting.

We also only examined speaking and spelling in offline experiments. Whether users could learn to use spelled variants to preemptively avoid recognition errors or to assist in correction episodes requires further investigation.

Conclusions

Supporting a rich life cycle of text creation, correction, and revision is challenging using only speech as input. While certainly existing interfaces provide a base level of functionality, they require more user effort than is strictly necessary.

This paper described our efforts to create the building blocks for a more fluid correction process that utilizes speech as the primary input modality. Our one-step voice correction process leverages not only the rich probabilistic information available from the recognition process, but also help from the user such as approximately where a revision is located and the spelling of difficult words. Future work is needed to ascertain how these methods work in combination in real-world speech interfaces.

References

[1] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine* 29, 6 (Nov 2012), 82–97. DOI: http://dx.doi.org/10.1109/MSP.2012.2205597

[2] Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks. *Computer Speech and Language* 14, 4 (2000), 373–400.

[3] Arthur E. McNair and Alex Waibel. 1994. Improving Recognizer Acceptance Through Robust, Natural Speech Repair. In *Proceedings of the International Conference on Spoken Language Processing*.

[4] Keith Vertanen. 2006. Speech and Speech Recognition during Dictation Corrections. In *Proceedings of the International Conference on Spoken Language Processing*. 1890–1893.

[5] Keith Vertanen and Per Ola Kristensson. 2009. Automatic Selection of Recognition Errors by Respeaking the Intended Text. In *ASRU '09: IEEE Workshop on Automatic Speech Recognition and Understanding*. 130–135. DOI: http://dx.doi.org/10.1109/ASRU.2009.5373347

[6] Keith Vertanen and Per Ola Kristensson. 2010. Getting it Right the Second Time: Recognition of Spoken Corrections. In *SLT '10: Proceedings of the 3rd IEEE Workshop on Spoken Language Technology*. 277–282. DOI: http://dx.doi.org/10.1109/SLT.2010.5700866

[7] Keith Vertanen and Per Ola Kristensson. 2012. Spelling as a Complementary Strategy for Speech Recognition. In *Proceedings of the International Conference on Spoken Language Processing*.