FIX IT WHERE IT FAILS: PRONUNCIATION LEARNING BY MINING ERROR CORRECTIONS FROM SPEECH LOGS

Zhenzhen Kou, Daisy Stanton, Fuchun Peng, Franc¸oise Beaufays, Trevor Strohman Google Inc., USA

This article examines the subject of automatic speech recognition (ASR) systems and misrecognitions or incorrect pronunciations. Typically, an ASR system contains a pronunciation dictionary (or lexicon) which consists of word-pronunciation pairs for known word pronunciations and a grapheme-to-phoneme (G2P) engine, which generates pronunciations for unknown words. Keeping the hand-generated list of know word pairs up to date is impossible, especially when pronunciation is often influenced by region. The authors in this article take a unique approach in helping solve this issue, by taking a language-independent approach of looking at misrecognitions and their corrections.

Others have looked into learning pronunciations from recognition corrections data using acoustic and prosodic feature, examining features of the speaker’s style to detect errors, decision-tree based methods to detect retries, and even a co-occurrence method for detecting and correcting misrecognition. The authors of this article look to use two new data mining techniques for detecting the misrecognitions: Keyboard Correction Data and Selected Alternate Data.

Keyboard Correction Data refers to query pairs (speech, keyboard) in which the user initiates a voice query and within a certain time limit (30 seconds for this article) types the same query with the keyboard. There are some challenges with this technique as the user may rephrase the query they had previously spoken when they type it in with a keyboard. The user may also choose to ask a completely unrelated question, but within the time limit. To account for these situations the correction data is classified using four feature-based criteria: word-based features, character-based features, phoneme features, and acoustic features.

Selected Alternate Data refers to a user interface that is part of the Google voice search that allows the user to manually select a corrected word from a list of possible alternate results. This provides high quality corrections, as the user knows what word they were trying to communicate and presumably know how the word should be pronounced or how they want it to be pronounced. This also eliminates the need for determining bad pronunciations ahead of time since the user will identify the misrecognitions themselves.

Evaluation of the two data mining techniques for detecting misrecognitions included metrics on the word error rate (WER) on test sets containing speech queries randomly selected from traffic logs and human transcriptions. Even though many of the words in this random set would already have good pronunciations since they were from a production-level system, the authors found it useful in determining their system had not unlearned any well-known/good pronunciations. In addition to the WER experiments, they also performed side-by-side (SxS) tests. Side-by-side experiments are useful because they focus on cases where pronunciation changes affect the recognition results. For the SxS experiments two ASR were built, one with the learned pronunciations and one without. Queries used for the recognition transcripts differed by one of four categories: nonsense – the transcript is nonsense, unusable – the transcript does not correspond to the audio, usable – the transcript contains only small errors, and exact – the transcript matches the spoken audio exactly.

Results of the experiments completed across 11 different languages showed a small reduction in WER for the Keyboard Correction Data as well as improvements in the SxS score. For the Alternate Selection Data, there was a smaller amount of data to evaluate (as less people use the interface than typing in the query themselves), so unfortunately the results showed no difference in the WER. However, there were significant improvements in the SxS evaluation going from a 0.418 to 0.457 for American English language and similar results for the other languages.