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

Zhenzhen Kou, Daisy Stanton, Fuchun Peng, Françoise Beaufays, Trevor Strohman

Google Inc., USA

ABSTRACT

The pronunciation dictionary, or lexicon, is an essential component in an automatic speech recognition (ASR) system in that incorrect pronunciations cause systematic misrecognitions. It typically consists of a list of word-pronunciation pairs written by linguists, and a grapheme-to-phoneme (G2P) engine to generate pronunciations for words not in the list. The hand-generated list can never keep pace with the growing vocabulary of a live speech recognition system, and the G2P is usually of limited accuracy. This is especially true for proper names whose pronunciations may be influenced by various historical or foreign-origin factors. In this paper, we propose a language-independent approach to detect misrecognitions and their corrections from voice search logs. We learn previously unknown pronunciations from this data, and demonstrate that they significantly improve the quality of a production-quality speech recognition system.

Index Terms— speech recognition, pronunciation learning, data extraction, logistic regression

1. INTRODUCTION

The speech recognition task is to find the word sequence W^* that has the mixmimum posterior probability given the acoustic observations $\mathcal{X}$,

$$W^* = \arg\max_i P(W_i|\mathcal{X}) \quad (1)$$

$$= \arg\max_i P(\mathcal{X}|W_i)P(W_i). \quad (2)$$

where $P(\mathcal{X}|W_i)$ is the acoustic model and $P(W_i)$ is the language model. Although some recent research [1] seeks to learn W_i directly from $\mathcal{X}$, a pronunciation model is typically introduced to generate phone sequences S_j^i for a word W_i:

$$W^* = \arg\max_i \sum_j P(\mathcal{X}, S_j^i|W_i)P(W_i) \quad (3)$$

$$= \arg\max_i \sum_j P(\mathcal{X}|S_j^i)P(S_j^i|W_i)P(W_i). \quad (4)$$

With the Viterbi approximation, Eq. 4 becomes

$$W^* = \arg\max_{i,j} P(\mathcal{X}, S_j^i|W_i)P(S_j^i) \quad (5)$$

$$= \arg\max_{i,j} P(\mathcal{X}|S_j^i)P(S_j^i|W_i)P(W_i). \quad (6)$$

As suggested before, the model $P(S_j^i|W_i)$ usually relies heavily on a handwritten list of word pronunciations, and defaults to a G2P engine for additional words. This often makes it a weak link across the entire ASR system: a word with an incorrect pronunciation will be systematically misrecognized for other words that better match its actual pronunciation.

We previously proposed a method to learn better word pronunciations through crowdsourcing [2]: words or phrases are sent to random contributors who speak them, thereby providing audio samples from which pronunciations can be automatically learned. Although extremely effective, this approach has some shortcomings, notably that unusually-pronounced words (like some street names) may be unknown to crowdsourcers located in different geographic regions. To alleviate this problem, we developed a new method to mine pronunciation learning training data directly from anonymized speech logs. Users confronted with a misrecognition may attempt to correct the error in various ways. In a voice search application, for example, they may repeat their query, rephrase it slightly differently, select an alternate recognition from a list (if offered the choice), type in a correction using the keyboard, or, of course, give up altogether.

In this paper, we focus on detecting and leveraging two types of user corrections: Keyboard Correction data and Alternate Selection data. The models we propose are language-independent. Side-by-side experiments demonstrate that the pronunciations learned via our methods significantly improve the quality of a production-quality speech recognition system.

2. RELATED WORK

There is a lot of research on applying machine learning for grapheme to phoneme conversion (G2P), including decision tree classifier to learn pronunciation rules [3], joint ngram model [4], maximum entroy model [5], active learning [6], and most recently recurrent neural network [7]. In this paper, instead of focuing on improving machine learning G2P techniques, we strive to learn pronunciations from recognition corrections data.

The literature contains many studies on detection speech recognition errors. Levow [8] investigated using acoustic and prosodic features to identify corrections. Orlandi et al. [9] proposed prosodic features to detect recognition errors. Soltau and Waibel [10] examined features related to the user's speaking style to detect speech errors. Levitan and Elson [11] proposed a decision-tree based method to detect voice query retries. Williams [12] investigated a dialog system that tracked a distribution over multiple dialogue states, the goal being to improve dialog ASR accuracy by modeling the entire N-best list. Shi and Zhou [13] examined the impact of different knowledge sources on human error correction. Sarpa and Palmer [14] proposed a co-occurrence method for detecting and correcting misrecognition. Our work introduces two new data mining methods to detect speech recognition errors from other retry behaviors. We use these corrections to improve word pronunciations, and demonstrate the end-to-end benefit of the proposed method on voice search

speech recognition accuracy.

3. KEYBOARD CORRECTION DATA

The first data source we propose mining is Keyboard Correction data. These are (speech, keyboard) query pairs derived from two consecutive actions observed in user session logs. If a user makes a voice search query, and then issues a typed query only a few seconds later, it may indicate an attempt at correcting a misrecognition (e.g. "Christmas Alex" followed by "Crispus Attucks"). The keyboard query can then be used as a supervised transcript to associate with the preceeding spoken query.

The event sequence for such a correction is illustrated Figure 1.

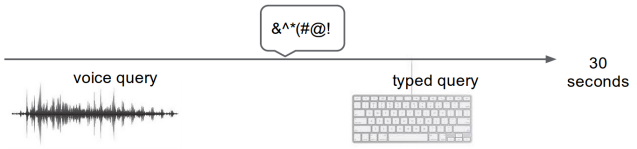

Fig. 1. Event sequence indicating a Keyboard Correction.

Since we expect corrections to come shortly after a failed recognition attempt, we mine spoken/typed query pairs that occured no more than 30 seconds apart. However, not all such candidate pairs do indicate a useable correction: the keyboard query may rephrase the spoken query in a slightly different way, or be completely unrelated. An analysis of candidate pairs across 11 languages showed that, in practice, only 30-40% of such pairs are true keyboard corrections. We thus propose a classification approach to identify pairs where the typed query is the exact transcription of the spoken utterance.

3.1. Correction Data Classifier

We treat the problem of identifying true Keyboard Corrections as a binary classification problem which we model with a logistic regression classifier [15],

$$h(X) = \frac{1}{1 + e^{-\theta^T X}} \tag{7}$$

where the weight θ for the feature vector X can be optimized from labeled training data with gradient descent. A prediction for X can then be classified as positive if $h(X)$ exceeds a threshold γ, and rejected as negative otherwise. The following features are used as inputs to the classifier:

- Word-based features, including unigram counts, number of word overlaps, and language model costs.
- Character-based features, including character counts, and edit distance between the recognized and typed queries.
- Phoneme features, including counts and edit distance between the phoneme sequences corresponding to the recognition resulst and typed query
- Acoustic features, including forced phone alignment costs, and waveform-to-transcript length ratio.

3.2. Keyboard Correction Classification Performance

We extracted 8000 consecutive spoken/typed query pairs from anonymized voice search session logs in each one of 11 languages. The keyboard entries were manually labeled as true/false transcripts of the voice queries, and were used to train a classifier using 10-fold cross validation.

Classification performance is measured in terms of precision and recall, as illustrated for American English in Figure 2. The classification threshold for each language, γ, was chosen to reach a precision of at least 90% in order to favor high quality data over data yield.

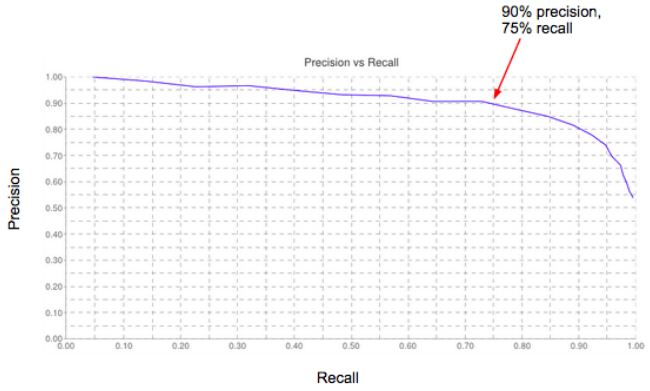

Fig. 2. P/R curve (en-US Keyboard Corrections classifier)

Table 1 shows some sample correction pairs in American English. We found that most corrections were related to homophone misrecognitions, or corresponded to words for which the pronunciation dictionary had no manual entry.

Table 1. Sample Keyboard Corrections (en-US)

Speech recognizer transcript	User keyboard query
Plus would Newton Kansas	Pluswood Newton Kansas
the cruise movie 3d	the croods movie 3d
what is a schematic	what is ischemic
the trucking part Arizona	the trotting park Arizona
Christmas Alex	crispus attucks

4. SELECTED ALTERNATE DATA

The Google voice search user interface allows the user to manually select from a list of alternative recognition results (Figure 3). This user feedback, which we call Selected Alternate data, provides high quality corrections.

Table 2 shows some examples of selected speech alternates. These transcripts, proposed from the N-best recognition output of the recognizer, are already well matched to the audio signal. The user selection is all that is needed to make it a strong supervised signal: no extra classifier is needed in this case.

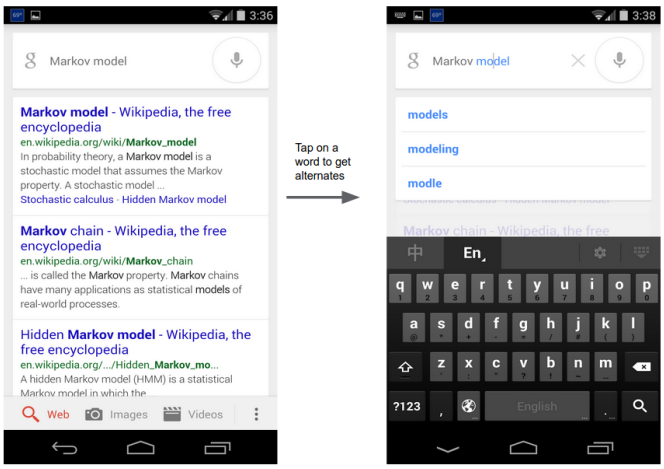

Fig. 3. Selection Alternates in Google voice search.

Table 2. Examples of Selected Alternates Data.

Speech recognizer transcript	User-selected alternate
my cat try to bite my Fi	my cat try to bite my thigh
which band was created in movie tomorrow	which band was Creed in movie Tamara
find a picture of a blue ku	find a picture of a beluga
Winston high Topeka Kansas	Quinton high Topeka Kansas
pictures of Renee swimming	pictures of Renee Fleming

5. ASR EXPERIMENTS

5.1. Pronunciation Learning

The experiments described in this section use the data mined in the two sections above as inputs to a pronunciation learning system. We followed the algorithm described in [2], which rewrites Eq. 6 for the purpose of pronunciation learning: it assumes that the word sequence W_i corresponding to the acoustic sequence $\mathcal{X}$ is given, but that multiple candidate pronunciations are available. We wish to find the pronunciation sequence S^* that maximizes the likelihood of the acoustic data, given that the word sequence is the user-corrected transcript:

$$S^* = \arg\max_j P(\mathcal{X}|S_j^i). \tag{8}$$

where $P(S_j^i|W_i)$ can be dropped if we assume equal priors on the pronunciation sequences.

The learned pronunciations were added to the recognition system and evaluated. The pronunciation candidates are generated from 20 best G2P pronunciations. More details on how pronunciation candidates are generated, and how learned pronunciations are added to the ASR lexicon can be found in found in [2].

5.2. Baseline Speech Recognizer

The baseline system used in these experiments is a production-level, large-vocabulary, state-of-the-art speech recognizer with a Deep Neural Network (DNN) acoustic model [16], a Finite State Transducer (FST) decoder [17], and a standard 5-gram language model trained on a variety of text corpora.

5.3. Evaluation Metrics

We performed word error rate (WER) evaluations on test sets containing anonymized speech queries randomly selected from traffic logs and human-transcribed. Because the underlying speech recognizer is a production-level system, the most frequent words in any given language already have good pronunciations. The words for which we seek to learn pronunciations lie in the tail of the query distribution, and most are thus unlikely to appear in the test sets. We nonetheless found it useful to compute the WER on such test sets to ensure we didn't learn any "rogue" pronunciation, i.e. a phone sequence for an infrequent word that matches the pronunciation of a different and more frequent word that would now be misrecognized. In addition to standard test set WER evaluations, we also measured the impact of pronunciation variants by performing side-by-side (SxS) tests.[1] For these experiments, we build two ASR engines: one without the learned pronunciations, and one with them. The acoustic model, language model, and vocabulary are the same in both engines. Only the lexicon changes. The two engines are used to recognize speech queries extracted from anonymized voice search logs. Queries for which the recognition transcripts differ between the two engines are evaluated by human raters, and marked as belonging to one of four categories:

1. nonsense: the transcript is nonsense.
2. unusable: the transcript does not correspond to the audio.
3. usable: the transcript contains only small errors.
4. exact: the transcript matches the spoken audio exactly.

We compute a weighted score for each engine's output as follows:

$$SxS\_Score = \frac{\sum_{i=1}^{4} W(i) * C(i)}{\sum_{i=1}^{4} C(i)} \tag{9}$$

where $C(i)$ is the number of samples from the i-th category, and $W(i)$ is the category weight (0, 0.25, 0.75, and 1, corresponding to the categories above).

An experiment is considered positive if its SxS score is higher than that of the corresponding baseline. Generally this means it has fewer nonsense/unusable queries and more usable and exact queries. A graphical representation of counts in the various categories, for the baseline and for the experiment, provides a more intuitive (if less rigorous) read on the experiment success than the SxS score, so we provide this as well.

Side-by-side experiments have the advantage of focusing on cases where pronunciation changes do affect the recognition results. They typically show more "movement" than WER measurements on fixed test sets.

5.4. Impact of Pronunciations Learned from Keyboard Correction Data

Table 3 reports recognition error rates in three US English test sets extracted from web search, mobile actions, and Maps traffic, respectively. We see a small reduction in word error rate on each test set.

[1] https://code.google.com/p/sxse/

Table 4 reports SxS score improvements from adding new pronunciations to the baseline ASR engine. Here we show results in British English, French, and German, to illustrate how the data classifier generalizes to other languages. Paired t-tests were used to compute a p-value of statistical significance in comparing the SxS scores.

Table 3. WER comparisons for Keyboard Corrections data on American English test sets.

Data set	# Utterances	Baseline WER	Experiment WER
Search	22K	10.9	10.8
Actions	12K	21.9	21.8
Maps	18K	11.4	11.3

Fig. 4 shows the distribution of each SxS category for British English; similar distributions were observed in other languages. Baseline system categories are shown in blue, and those for the experiment (with added pronunciations) are shown in red. The distributions clearly show a large movement from Nonsense to Exact.

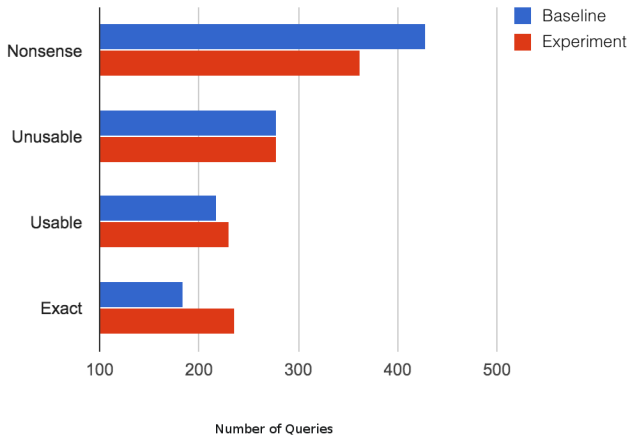

Fig. 4. SxS category distributions for Keyboard Corrections (en-GB)

5.5. Impact of Pronunciations Learned from Alternate Selection Data

The amount of data flowing through the Alternate Selection pipeline is smaller than that from Keyboard Corrections. As a result, the system learns fewer pronunciations, and our experiments showed no impact on standard test set word error rates. However, SxS evaluations showed significant improvements. Results for American English are shown in Fig. 5. Here the baseline score was 0.418 (blue), and the experiment 0.457 (red). The p-value for these scores is less than 0.001.

Table 4. SxS scores for Keyboard Corrections in three languages.

Language	Baseline score	Experiment score	p-value
English	0.376	0.432	<.001
French	0.382	0.468	<.001
German	0.416	0.489	<.001

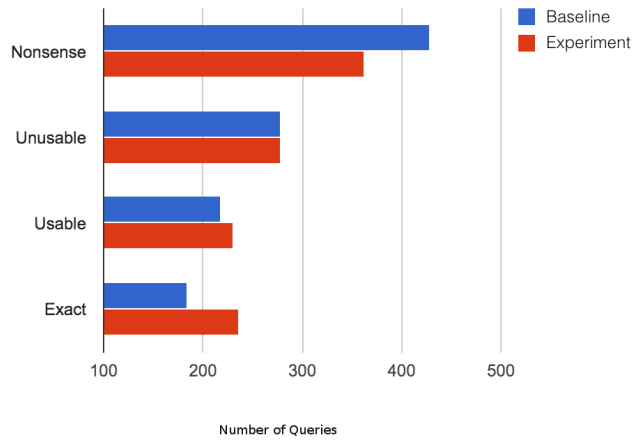

Fig. 5. SxS category distributions for Alternate Selections (en-US)

5.6. Examples

The pronunciations learned from Keyboard Corrections and Selected Alternates data fix bad pronunciations for words already in the lexicon word list, or replace G2P pronunciations for words not in the list. Table 5 shows some sample pronunciations of the initial best G2P pronunciation and the final learned pronunciation, expressed in X-SAMPA[2] phone notation, learned through these methods. They mostly consist of tail words extracted from business names or websites.

Table 5. Pronunciations learned from Keyboard Corrections and Selected Alternates

word	best G2P pronunciation	learned pronunciation
sephora	s E f O r @	s @ f O r @
tasca	t A s k @	t { s k @
estas	E s t @ z	E s t { s
verdi	v @ ‘ d i	v E r d i
newman	n u m @ n	n j u m @ n

6. CONCLUSIONS

In this work, we presented an approach to mine untranscribed voice search logs to extract training data from which to learn word pronunciations. The fact that a misrecognition has been corrected by a user provides supervision to the learning process. Correction data has the advantage of focusing specifically on the areas of weaknesses of the system: we do not need to identify bad pronunciations ahead of time to know which words to learn. Also, corrections are provided by the users who spoke the queries, and presumably know how the words (often proper names) should be pronounced, or at least how they want to pronounce them.

We discussed two types of correction data: Keyboard Correction and Selected Alternate data, and showed that both provide new pronunciations whose quality was demonstrated through side-by-side speech recognition experiments.

[2]http://en.wikipedia.org/wiki/X-SAMPA

7. REFERENCES

[1] Alex Graves and NavDeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of ICML*, 2014.

[2] Attapol Rutherford, Fuchun Peng, and François Beaufays, "Pronunciation learning for named-entities through crowd-sourcing," in *Proceedings of Interspeech*, 2014.

[3] Alan W Black, Kevin Lenzo, and Vincent Pagel, "Issues in building general letter to sound rules," in *International Speech Communication Association*, 1998.

[4] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communications*, vol. 50, no. 5, pp. 434–451, 2008.

[5] Stanley F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proceedings of InterSpeech*, 2003.

[6] John Kominek and Alan W Black, "Learning pronunciation dictionaries: language complexity and word selection strategies," in *Proceedings of HLT-NAACL*, 2006.

[7] Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *Proceedings of ICASSP*, 2015.

[8] Gina-Anne Levow, "Characterizing and recognizing spoken corrections in human-computer dialogue," in *Proceedings of ACL*, 1998, pp. 736–742.

[9] Marco Orlandi, Christopher Culy, and Horacio Franco, "Using dialog corrections to improve speech recognition," in *Error Handling in Spoken Language Dialogue Systems*, 2003.

[10] Hagen Soltau and Alex Waibel, "On the influence of hyperarticulated speech on recognition performance," in *Proceedings of ICSLP*, 1998.

[11] Rivka Levitan and David Elson, "Detecting retries of voice search queries," in *Proceedings of ACL*, 2014, pp. 230–235.

[12] Jason D. Williams, "Exploiting the asr n-best by tracking multiple dialog state hypotheses," in *Proceedings of Interspeech*, 2008, pp. 191 – 194.

[13] Yongmei Shi and Lina Zhou, "Examining knowledge sources for human error correction," in *Proceedings of Interspeech*, 2006.

[14] Arup Sarma and David D. Palmer, "Context-based speech recognition error detection and correction," in *Proceedings of HLT-NAACL*, 2004.

[15] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning (2nd ed.)*, New York: Springer, 2009.

[16] Vincent Vanhoucke, Matthieu Devin, and Georg Heigold, "Multiframe deep neural networks for acoustic modeling," in *Proceedings of ICASSP*, 2013.

[17] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.