AUTOMATIC PRONUNCIATION VERIFICATION FOR SPEECH RECOGNITION

Kanishka Rao, Fuchun Peng, Françoise Beaufays

Google Inc., USA

ABSTRACT

Pronunciations for words are a critical component in an automated speech recognition system (ASR) as mis-recognitions may be caused by missing or inaccurate pronunciations. The need for high quality pronunciations has recently motivated data-driven techniques to generate them [1]. We propose a data-driven and language-independent framework for verification of such pronunciations to further improve the lexicon quality in ASR. New candidate pronunciations are verified by re-recognizing historical audio logs and examining the associated recognition costs. We build an additional pronunciation quality feature from word and pronunciation frequencies in logs. A machine learned classifier trained on these features achieves nearly 90% accuracy in labeling good vs bad pronunciations across all languages we tested. New pronunciations verified as good may be added to a dictionary, while bad pronunciations may be discarded or sent to experts for further evaluation. We simultaneously verify 5,000 to 30,000 new pronunciations within a few hours and show improvements in the ASR performance as a result of including pronunciations verified by this system.

1. INTRODUCTION

Speech recognition systems require a lexicon containing the pronunciations of valid words in a language. Traditionally these lexicons are written manually by linguists which tends to be a costly and slow process. Further, a lexicon recorded by experts may not cover all the valid words and requires frequent updates as new words and pronunciations are created. Linguists may also make errors in writing pronunciations, for example, for the word "Bexar" (a Texas county), an expert writes a pronunciation 'b eh k s ao r' which seems reasonable; however, the correct pronunciation for this word is 'b eh r'. As an alternative, scalable and fast solutions to generate pronunciations automatically exist, either using grapheme to phoneme (G2P) conversion algorithms [2, 3, 4] or data-driven pronunciation learning techniques [1, 5]. Such automated tools are powerful as a single mechanism to generate pronunciations for any new word in a particular language, but they may generate incorrect pronunciations for many words (e.g. [6] cites 24% word error rate for US English). A recognition engine may use both linguist-written lexicons along with a G2P model but still may perform poorly due to the lack of coverage from the lexicon and errors from the either linguists or the G2P. One may validate a batch of pronunciations by comparing ASR performance with and without that batch but this does not verify the quality of the individual pronunciations.

In this work, we propose an automated pronunciation verification system that verifies whether a new individual pronunciation, either transcribed by a linguist or automatically generated, is good for ASR. The verification system is automated, fast, scalable and language independent and can be used in conjunction with any pronunciation generation system. The verification system relies on historical audio logs and picks those that may be affected by the new pronunciations and then uses a series of features to estimate whether each new pronunciation affected recognition in a positive or negative manner. Experiments on a state of the art speech recognizer confirm that the approach can predict the pronunciation quality to a high level of accuracy, 90% across many different languages. The quality score is strongly correlated to the human judgment of speech recognition quality.

2. RELATED WORK

The field of pronunciation modeling has seen a lot of research on grapheme to phoneme conversion [2, 7] and data driven pronunciation learning [5, 1]. In this paper, rather than learning new pronunciations, we focus on validating the pronunciations generated from G2P or audio data.

Literature on pronunciation verification has mostly focused on predicting if a pronunciation is close to a human expert pronunciation from the point of view of sounds rendition, tones and stress. For example, Tepperman et al. [8] uses a classification approach to verify children's pronunciations based on acoustic features for the purpose of assessing children's literacy. Molinaa et al. [9] use a classification approach for computer aid pronunciation training. Wuth et al. [10] and Neumeyer et al. [11] assess the foreigner's pronunciation quality compared to native speakers. Our system instead validates pronunciations for the purpose of improving ASR, thus, we directly aim at improving the quality of speech recognition results. To our knowledge, this is the first published work for an automated, language independent pronunciation verification system that shows improvements in ASR.

3. PRONUNCIATION VERIFICATION

The input to the verification system is a list of new pronunciations for verification, we setup three different recognizers with three different sets of pronunciations using the input list. Finally, a classifier uses features from these recognizers to label each individual pronunciation as good or bad.

The pipeline consists of several steps.

1. Setup: Create ASR engines with the new pronunciations for re-decoding speech utterances.
2. Relevant Utterance extraction: Select a large number of speech utterances whose recognition results may be affected by any of the new pronunciations.
3. Comparison to baseline: Extracted utterances are re-recognized and those results that differ from the baseline (before the inclusion of the new prons) are selected for further analysis.
4. Responsible pronunciation detection: The new pronunciation(s) that is responsible for each recognition result difference is identified.

5. Aggregated metrics: Metrics are used to gauge if the result difference is a good, bad or neutral change. For each of the new prons these metrics are aggregated over all the differences it is responsible for. We create recognition based metrics and log based metrics.
6. Pronunciation quality classifer: A machine learned model classifies every new pronunciation as good or bad based on the aggregated metrics.

Below we describe these steps in more details.

3.1. Setup

In this first step, we set up three speech recognition engines that only differ by their lexicons:

1. Baseline, E_1: Includes only the baseline pronunciations for the words to be verified.
2. Append, E_2: The new pronunciations are appended to the baselines.
3. Replace, E_3: The new pronunciations replace the baseline pronunciations for the words to be verified.

3.2. Utterance Filter

We play a very large amount of *untranscribed* anonymized user utterances through the three speech engines, and we select the (much fewer) utterances for which we get different speech recognition results from the *Append* E_2 or *Replace* E_3 engines compared to the *Baseline* E_1 engine. It is important to stress that the input utterances do not need to be hand-transcribed. This allows us to essentially push as many utterances as we need through the pipeline to find the interesting ones: those whose recognition is affected by the pronunciation changes.

However, redecoding a lot of utterances is a resource and time extensive task. Most utterances will not be impacted by the proposed lexicon change. To speed up, we implemented a much faster pre-filtering step that filter un-affected utterances. An utterance is selected if only one of the conditions are met:

1. Word Match: its transcript contains any word in the proposed pronunciation list. This make sure that all utterances currently containing interested words are covered.
2. Phone Match: the phoneme sequence of its transcript contains any of the proposed pronunciations.

All other utterances can be safely discarded.

3.3. Comparison to baseline

For all the redecoded utterances we compare the recognition results t_2, t_3 from the Replace and Append engines to the Baseline t_1. We examine the language model costs (LMC) [1] and the acoustic model costs (AMC) as calculated by the ASR for all three results. These recognition costs are used as a relative measure to evaluate $t_1 \rightarrow t_2$ and $t_1 \rightarrow t_3$. For example, if both the LMC and AMC are lower for t_2 compared to t_1 then we count $t_1 \rightarrow t_2$ as a *win*. Specificly, we design 6 utterance level metrics.

1. $Activity_{append}$ = 1 if $t_1 \neq t_2$, else 0.
2. Win_{append} = 1 if $LMC_2 < LMC_1$ and $AMC_2 < AMC_1$, else 0.
3. $Loss_{append}$ = 1 if $LMC_2 > LMC_1$ and $AMC_2 > AMC_1$, else 0.
4. $Activity_{replace}$ = 1 if $t_1 \neq t_3$, else 0.
5. $Win_{replace}$ = 1 if $LMC_3 < LMC_1$ and $AMC_3 < AMC_1$, else 0.
6. $Loss_{replace}$ = 1 if $LMC_3 > LMC_1$ and $AMC_3 > AMC_1$, else 0.

The idea behind these metrics is that a good pronunciation will reduce acoustic model and language model costs when aggregated over many utterances while bad pronunciations will do the opposite.

The metrics for *Append* and *Replace* capture different aspects of a pronunciation. For example, in an interesting case where a word may have multiple valid pronunciations a good pronunciation may result in more aggregated losses than aggregated wins in the Replace engine. For example for the word 'cosplay', the baseline contains a pronunciation 'k ao s p l ey' and if we verify another good pronunciation 'k aa z p l ey' we find this new prounciation has 15 Replace wins and 17 Replace losses. Both pronunciations are good but 'k aa z p l ey' is only favorable for 15 of the utterances while 'k ao s p l ey' is favorable for 17 utterances. In contrast, the Append engine has both pronunciations available and will choose the most favorable one for each utterance, in this example 'k aa z p l ey' has 15 Append wins and 0 Append losses. Having both Replace and Append metrics allows us to properly handle such cases. In another scenario a bad pronunciation may never be seen in the comparison between the Baseline and Append system if good pronunciations already exist for that word (not an uncommon case), the new bad pronunciation may never be utilized by the ASR and recognition results would not change. This case is avoided in the comparison of the Baseline with the Replace system, here since the new pronunciation in the Replace system is the only one for that word the ASR is forced to use it. Again, using both Replace and Append metrics allows us to identify such situations.

3.4. Pronunciation Detection

For each recognition result difference we wish to associate the differences and the metrics to one of the new pronunciations being verified. For example, if we are verifying a new pronunciation for the word 'Buuren' we may observe that a given utterance u was transcribed as t_1 : "play Armin Van Buren music" with the Baseline engine and as t_2 : "play Armin Van Buuren music" with the experimental Append engine. In this case it is easy to attribute this difference to the new pronunciation for 'Buuren' as it is introduced in t_2.

However, in reality things are not always this simple as we verify up to 30,000 new pronunciations simultaenously. We use the following rules to assign each difference to a new pronunciation; (1) if a word and its new pronunciation appear in the new result and not in the baseline result (2) if a word appears in the baseline and not in the new result and there is new pronunciation for this word. A typical utterance contains about 10 words and we encounter cases where multiple pronunciations are found responsible for the same utterance, since we cannot attribute the change to a single new pronunciation we discard such utterances.

In some utterances we may see recognition results change even though none of the pronunciations or words involved in their two recognition results were included in the batch of new pronunciations to evaluate. Instead, the change in pronunciation lexicon affected

[1] Cost is defined as the negative of the log probability.

the decoder graph in a way that confuses the decoding of these utterances. Such differences will not be associated with any new pronunciation and will be ignored by the pronunciation verification system. However such differences are infrequent and typically we find them to comprise less than 5% of all the differences.

4. AGGREGATED METRICS

We designed two types of metrics that are later used by the classifier, based on recognition results and based on search and speech logs.

4.1. Recogntion Metrics

Once metrics at the utterance level are created, we need to aggregate them at the pronunciation level to be used by the classifier. For a given pronunciation we aggregate recognition-based features over the set of recognition results, U, affected by it. We construct the following three recognition-based aggregated metrics:

1. Append Gain, $\frac{\sum_U Win_{append} - \sum_U Loss_{append}}{\sum_U Activity_{append}}$
2. Replace Gain, $\frac{\sum_U Win_{replace} - \sum_U Loss_{replace}}{\sum_U Activity_{replace}}$
3. Append Fraction,

$$\begin{cases} \frac{\sum_U Activity_{append}}{\sum_U Activity_{replace}}, & \text{if } \sum_U Activity_{replace} > 0 \\ 0, & \text{otherwise} \end{cases}$$

4.2. Log based metrics

The signals discussed so far rely on spech model costs (acoustic model and language model) as an indicator of pronunciation quality, however, in some situations these are not reliable. Common words may lower language model costs while short pronunciations might result in small acoustic model costs. To complement the recognition cost signal we construct an orthogonal feature from historical logs of spoken and typed queries. We call this metric a "pronunciation potential". If a given word had a bad pronunciation, it is less likely to appear in spoken logs than typed logs, while a word with robust pronunciations will have comparable relative occurence in spoken and typed logs. We define the word potential $\phi_{word} = n_s/n_t$, words with $\phi \approx 0$ may have very poor pronunciations and thus do not make it in to the spoken logs. If the word potential is indeed due to the lack of a good pronunciation then a new good pronunciation will have more phone matches than word matches as described in section 3.2, the idea being that the phonetic sound associated with the good new pronunciation exists in the historical utterance but since the good pron was not available when the utterance was first decoded the word was not recognized in the transcript. Then the ratio of phone matches (m_{ph}) to word matches (m_w) must be at least equivalent to the word potential, $m_{ph}/m_w \geq n_s/n_t$ or $m_{ph}n_t/n_s - m_w \geq 0$. We construct a normalized feature, pronunciation potential,

$$\phi_{pron} = \frac{m_{ph}n_t/n_s - m_w}{m_{ph}n_t/n_s + m_w}$$

Figure 1 shows the distribution of the pronunciation potential feature for true good and true bad pronunciations.

Pronunciation potential, which is independent of the six recognition costs related features, serves as an additional feature in determining pronunciation quality.

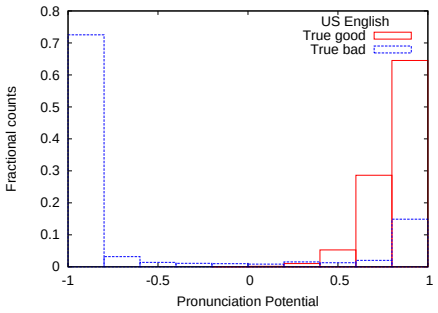

Fig. 1. The distribution of pronunciation potential for truth good and truth bad pronunciations.

5. PRONUNCIATION QUALITY CLASSIFIFER

5.1. Model and Features

A classifier labels the new pronunciation as good or bad based on the aggregated recognition metrics and the pronunciation potential features. We train a Boosted Decision Tree [12] with 100 rounds and maximum depth of 10 per tree. We perform a 6-fold cross validation using data from each language as one fold as a check for over-fitting.

The four features are described in Section 4.

5.2. Data

We created data sets for 6 different languages to demonstrate the approach is language independent: US English, British English, French, Spanish, Italian and Brazilian Portugese.

For each of the languages we have 2500 words manually transcribed to obtain prounciations, these are considered as true good pronunciations. For another 2500 words, we apply a G2P and pick the 30th candidate as true bad pronunciations. This is because that the quality of G2P is usually decreasing fast with the order of candidate, and the 30th candidate is a bad pronunciation in most cases we looked at. We run these 5000 new pronunciation through the pronunciation verification system and measure performance based on accuracy of predicted the truth good or bad labels. Figure 2 shows wins minus losses from the Replace system for US English.

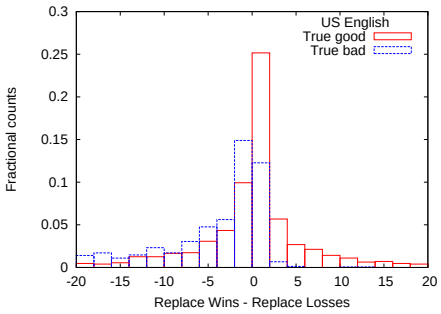

Fig. 2. The distribution of the number of wins minus losses from the Replace engine compared the Baseline engine for truth good and truth bad pronunciations.

Language	Classifier Accuracy
US English	91.5%
GB English	88.7%
FR French	90.2%
ES Spanish	91.0%
IT Italian	92.9%
BR Portugese	97.0%

Table 1. The accuracy of the pronunciation verification system across all tested languages using 30 days worth of audio logs.

5.3. Classifier performance

We observe a range of 89% to 97% accuracy across the 6 languages (Table 1). Of the new pronunciations, some may not appear in any of the filtered historical logs, for example with filtering over the previous 30 days worth of logs we find the coverage to be >70%. The remaining words may be captured by increasing the number of days of logs, however, if a word is never seen over a long period in the logs then it would have no impact on the ASR performance and may be neglected.

6. ASR EXPERIMENTS

6.1. Impact on ASR

To evaluate the impact of pronunciation verification on ASR we verified a set of 10,000 new candidate pronunciations generated by the Pronunciation Learning system [1]. Of all the new pronuciations only 81% were found in the historical logs and thus were verified. Of these found pronunciations 37% were labeled as good and 63% were labeled as bad. On average, each pronunciation affected 98 speech utterances and only 1.7% of all the affected utterances could not be matched to a new pronunciation.

Measuring the impact of a new pronunciation relies on that word being in a test set, however, most of the verified pronunciations are for tail queries and they show no word error rate impact on our test sets. Thus to evaluate the impact of these pronunciations we use human raters, we re-recognize a test set of anonymised audio logs before and after adding in the good labeled pronunciations to the ASR system. Human raters evaluate the resulting transcripts from both engines as either ”Nonsense”, ”Unsuable”, ”Usable” or ”Exact”, only the cases where the transcripts are different from both engines are rated. We see an overall improvement in ASR after adding in the good labeled pronunciations in Figure 3. The same procedure is followed for the bad labeled pronunciations and we see an overall drop in performance, Figure 4.

This shows that the verification labels agree with ASR performance and that the pronunciation-level labels allow one to identify only the good pronunciations from a batch of candidate pronunciations.

6.2. Error Analysis

An example of a false positive is the pronunciation ‘latte l ih t iy’ which was labeled as a good pron. Most poor pronunciations show considerable losses in the *Replace* system and a fewer number of losses in the *Append* system. However, this pronunciation showed *Replace* losses but no losses were found in the *Append* system. If the baseline pronunciations for a word are comprehensive, as they are for the word ‘latte’, an additional poor pronunciation may not

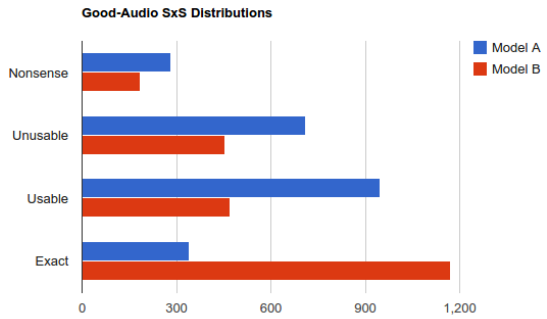

Fig. 3. The impact on ASR due to the good labeled pronunciations as rated by humans. Model A is the Baseline system and model B is with the good labeled pronunciations.

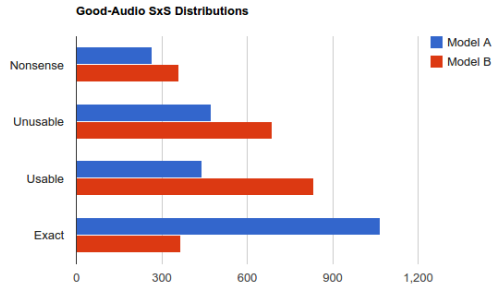

Fig. 4. The impact on ASR due to the bad labeled pronunciations as rated by humans. Model A is the Baseline system and model B is with the bad labeled pronunciations.

cause any negative impact, especially if it does not conflict with any other word. The lack of enough poor examples in the Append system and a co-incidental match for pronunciation caused a mis-labeling by the verification system.

False negatives are less egregious and are typically mis-labelled due to lack of examples in the logs, this is typical for good pronunciations for very rare words. In other cases a linguist may deem a pronunciation as appropriate for a word without the knowledge of the existing pronunciations for that word. If the existing pronunciations are exhaustive then adding a new very similar pronunciation may not result in any additional wins. For example, a valid pronunciation ‘colfax k ax l f ae k s’ was labeled as bad by the verification system since the existing pronunciation ‘colfax k ow l f ae k s’ correctly described all the audio logs.

7. CONCLUSIONS

We have proposed a data-driven and language-independent framework to automatically predict the quality of word pronunciations, pronunciations that are labeled good or bad have a 90% accuracy on labeled test sets. Quality signals are built from recognition costs and frequencies in spoken and typed logs. We show that the good labeled pronunciations improve overall ASR performance while the bad labeled pronunciations hurt performance. Thus such a pronunciation verification pipeline can be used to determine which pronunciations of a list of new candidates should be included and which discarded.

8. REFERENCES

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

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

[3] Ian McGraw, Ibrahim Badr, and James Glass, "Learning lexicons form speech using a pronunciation mixture model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, 2013.

[4] B. Maison, "Automatic baseform generation from acoustic data," in *Proceedings of Eurospeech*, 2010.

[5] Liang Lu, Arnab Ghoshal, and Steve Renals, "Acoustic data-driven pronunciation lexicon for large vocabulary speech recognition," in *In proceedings of ASRU*, 2013.

[6] J. R. Novak, N. Minematu, and K. Hirose, "Failure transitions for joint n-gram models and g2p conversion," in *Proceedings of Interspeech*, 2013.

[7] Xiao Li, Asela Gunawardana, and Alex Acero, "Adapting grapheme-to-phoneme conversion for name recognition," in *Proceedings of ASRU*, 2007.

[8] Joseph Tepperman, Jorge Silva, Abe Kazemzadeh, Hong You, Sungbok Lee, Abeer Alwan, and Shrikanth Narayanan, "Pronunciation verification of children2̆019s speech for automatic literacy assessment," in *Proceedings of ICSLP*, 2006.

[9] Carlos Molinaa, Nstor Becerra Yomaa, Jorge Wutha, and Hiram Vivancob, "Asr based pronunciation evaluation with automatically generated competing vocabulary and classifier fusion," *Speech Communications*, vol. 51, no. 6, pp. 485–498, 2009.

[10] Jorge Wuth, Nstor Becerra Yoma, Leopoldo Benavides, and Hiram Vivanco, "Asr based pronunciation evaluation with automatically generated competing vocabulary and classifier fusion," in *Proceedings of InterSpeech*, 2012.

[11] Leonardo Neumeyer, Horacio Franco, Vassilios Digalakis, and Mitchel Weintraub, "Automatic scoring of pronunciation quality," *Speech Communication*, vol. 30(2-3), pp. 83–93, 2000.

[12] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning (2nd ed.)*, chapter 10. Boosting and Additive Trees, pp. 337–384, New York: Springer, 2009.