Interspeech 2018
2-6 September 2018, Hyderabad

On Training and Evaluation of Grapheme-to-Phoneme Mappings with Limited Data

Dravyansh Sharma

Google LLC, USA
drasha@google.com

Abstract

When scaling to low resource languages for speech synthesis or speech recognition in an industrial setting, a common challenge is the absence of a readily available pronunciation lexicon. Common alternatives are handwritten letter-to-sound rules and data-driven grapheme-to-phoneme (G2P) models, but without a pronunciation lexicon it is hard to even determine their quality. We identify properties of a good quality metric and note drawbacks of naïve estimates of G2P quality in the domain of small test sets. We demonstrate a novel method for reliable evaluation of G2P accuracy with minimal human effort. We also compare behavior of known state-of-the-art approaches for training with limited data. Finally we evaluate a new active learning approach for training G2P models in the low resource setting.

Index Terms: grapheme-to-phoneme models, low-resource languages, language resource evaluation, metric, scale, alignment, active learning

1. Introduction

A pronunciation model, also known as a letter to phoneme conversion system, is a linguistically informed system to produce a phonemic representation of a word. The word is converted from the sequence of letters in the orthographic script to a sequence of phonemes (sound symbols) in a pre-determined notation standard, such as IPA, X-SAMPA, etc. To perform this task, on one extreme one could store all word-pronunciation pairs in a lookup-table and on the other, one could approximate the sounds by simply storing the phonemes corresponding to each letter in the alphabet. While the latter works badly when the same orthographic segment can have a number of different pronunciations, the former does not scale well to proper names, languages with productive compounding, etc. Thus, neither is typically sufficient, and we have a spectrum of non-trivial models in between.

Pronunciation models are critical components of both speech recognition (ASR) and synthesis (text-to-speech, TTS) systems. Even though end-to-end models have been gathering recent attention [1] [2], most state-of-the-art models in current industrial production systems involve conversion to and from an intermediate phoneme layer. In synthesis, we convert a sequence of words to a sequence of "sound symbols" which are then converted to audio. This happens in reverse in ASR systems. It is in fact possible to share the same pronunciation model in traditional TTS and ASR systems, the mapping being used in opposite directions.

To scale pronunciation models across languages we need a data-driven approach, but data is often the bottleneck in several languages. The process of correctly labeling data (also called transcription of pronunciations) tends to be skillful work. For low-resource languages, there is typically no readily available pronunciation lexicon of size even large enough to train a reasonable G2P model. Further, transcribers are particularly hard to find as linguistic expertise is difficult to come by. Transcription reliability is relatively harder to ensure since we cannot typically augment transcription quality by playing back synthesis [3] for these languages, as good synthesis systems are hard to build without having a good pronunciation model first. For the same reason, it is also hard to build a good Pronunciation Learning model [4] [5].

Several other unique challenges are posed by this setting. Many recent advances in recurrent neural network based algorithms [6] don't apply as well because the data available is not large enough. In other words, we can't have powerful parametric algorithms with many parameters here as that would lead to overfitting. Another challenge noted during this work is the existence of scripts where there may still be multiple codepoint sequences in Unicode Normalized Canonical Form that correspond to the same visual rendering (resulting in noisy data), e.g. in Indian scripts, which make it tricky to do G2P reliably, can cause confusion in alignment, etc.

2. Background and Related Work

Handwritten G2P rules are often used as the G2P model when data is scarce to build a good model by standard techniques. More scalable data-driven techniques for building G2P models for low resource languages have been recently developed. For example, by identifying closely related high-resource languages and extrapolating between them [7], or using a mix of language specific and language independent techniques [8], it is possible to obtain G2P models without having a pronunciation lexicon. The robustness of the quality claims are however not clear. For example, only 200 words are used for test set for most languages, sometimes even just 50 words in [7, Sec. 4.4, pp. 402]. We demonstrate in the following sections that pure WER is unreliable for small number of test words due to high variance. We also provide a framework for comparing evaluation metrics in this setting and a novel metric for evaluation.

Another interesting outcome of a robust extrapolative accuracy estimation, would be a step towards the "elusive" goal of determining whether a pronunciation model meets a certain threshold coverage without explicitly transcribing all the words [9, Sec. 1]. To this end, we also study behavior of various state-of-the-art techniques of building G2P models with varying amounts of limited data. Well known successful techniques include doing a joint-sequence alignment using n-grams [10] or using recurrent neural networks [6] [11].

Active learning has been attempted in the past [9], [3] for G2Ps. We also present a new alignment based active algorithm which shows good quality with fewer transcriptions.

3. G2P Evaluation

We start with noting that most of the following work is not done using typical low-resource languages. This is deliberate since

10.21437/Interspeech.2018-1920

Figure 1: *Accuracy estimates by different metrics on different subset sizes.*

we want to evaluate our evaluation metrics against standard metrics which involve large pronunciation lexicons and creating a large pronunciation lexicon is, by definition, infeasible for low-resource languages. We get around this issue by pretending we have to do pronunciation model training and evaluation without a lexicon for languages where we actually have a pronunciation lexicon. This setting simply allows us to simulate transcriptions by simple lexicon look-up from large pronunciation dictionaries written by experts.

Note that throughout this work we assume we use the Word Error Rate, i.e. the absolute fraction (over a large vocabulary) of words where G2P yields an inexact phoneme sequence, as the measure of G2P accuracy. To evaluate a G2P, we consider the class of approaches which sample a subset of the vocabulary of interest and test the G2P on the sample to estimate the accuracy. In what follows, we overload the term *metric* to refer to the combined framework of sampling and estimating quality, as well as the measure of quality itself.

3.1. What's a "good" metric?

Any usable metric for G2P evaluation must converge to a limiting value in the limit of extensive evaluation. We identify following additional properties that a 'good' metric (for evaluation with few transcribed words) should satisfy

- Converges fast: The metric should achieve its limiting value using as few words as possible.
- Converges reliably, i.e. close to true error value (on full vocabulary): An estimator of word error rate should yield good approximation of word error rate on known vocabulary, using just pronunciations for the limited set.
- Converges stably: Variance should be low in the intended range of data size. Convergence properties should hold irrespective of G2P choice and vocabulary (as long as it's large enough).

Intuitively, these properties allow the metric to be a good proxy to full scale transcription on the known vocabulary.

3.2. Proposed metric

We describe below a metric inspired from the Feature Coverage Maximization algorithm of [12]. As suggested there we use a stratified version, i.e. sub-divide the budget of transcriptions among words of different lengths in proportion to frequency. This is to avoid choosing really long words, which tend to be difficult to transcribe accurately, as a practical matter. Let V be the set of words in the vocabulary. This is essentially the set of words we want to evaluate the G2P accuracy on. Also, let $W(n, w)$ denote the number of times a character 4-gram n appears in a word $w \in V$. Also for character 4-gram n and word w define,

$$wt_0(n) := \sum\nolimits_{w \in V} W(n, w) \tag{1}$$

$$cov_0(w) := \sum\nolimits_{n \in N(w)} wt(n) \tag{2}$$

where $N(w)$ is the set of 4-grams in w.

We greedily select (add to a set S) the word w_j ($w_j \in V \setminus S$) with maximum $cov_j(w)$ and update

$$wt_{j+1}(n) = \alpha wt_j(n),\ \forall\, n \in N(w_j) \tag{3}$$

$$cov_{j+1}(w) = \sum\nolimits_{n \in N(w)} wt_{j+1}(n),\ \forall\, w \in V \tag{4}$$

for *discount factor* $\alpha = 0.2$ (anything in $[0.1, 0.5]$ works as well). This is repeated until $|S|$ equals the budget of transcriptions. The estimate of accuracy η is given by

$$\eta = \frac{\sum_{w \in C} cov_0(w)}{\sum_{w \in S} cov_0(w)} \tag{5}$$

where C denotes the set of transcriptions from S where the G2P is correct. In addition to the definition of η, we note that the other significant difference from [12] is that $wt_0(n) = 1$ in their work. In the next section we note how this change is crucial in deciding the metric quality.

3.3. Comparative study

We show the importance of the weighting extension by simultaneously plotting weighted and unweighted versions of the feature coverage algorithm, alongside the baseline metric of computing vanilla WER on a random sampling. We do this by looking successively at the three parameters of metric goodness. For the empirical evaluation we estimate accuracy of G2P models built using the traditional joint n-gram approach in [10] for a

set of six languages. The known vocabulary consists of up to hundreds of thousands of words for each language, essentially all words in our pronunciation lexicons. We do 5 iterations from different vocabulary subsets for each language of selecting subsets of sizes $100, 200, ..., 1000$ and compute accuracy using three metrics: 'Random' sampling with usual WER, 'Weighted-FCM' or 'WFCM' i.e. metric of 3.2, and 'FCM' i.e. metric of 3.2 with unit initial weights. See Figure 1 for metric values for a single iteration. The 'True' line in the figure corresponds to WER for the known vocabulary (using pronunciation lexicon).

3.3.1. Convergence rate

We plot averages over iterations for different languages (omitted for space) and note that WFCM is the fastest to converge. For most languages WFCM is already very close to the limiting value in the range 200-300 words and stays stable after that. Random and FCM take significantly longer to stabilize, and for most languages are not as stable even with 1000 transcriptions.

3.3.2. Convergence point

Convergence point is computed by estimating the limit to be the average of the iterations $800, ..., 1000$ for the respective algorithms. Notice both Random and WFCM are good approximates to true error but FCM is less reliable. We know that Random does converge to the exact true error in the limit, but it does not perform well on the other two criteria. For WFCM we can't give equally strong guarantees, and rely on empirical convergence reasonably close to the true value.

3.3.3. Convergence stability

We compute coefficient of variance for iterations on sizes $200, ..., 500$ which is the typical range of transcriptions where the subset size should lie for our algorithm, and notice an over 25% reduction w.r.t. random choice. We also note that FCM stability is not better than Random on average.

4. Comparison of Training Algorithms

In this section we look at a few well-known state-of-the-art G2P training algorithms and compare their performance with limited training data.

4.1. Experiment Setup

We consider two broad categories of approaches: n-grams with different n values in [10] and sequence-to-sequence attention-based RNN-transducer models [11] [13]. We select subsets using the algorithm in [12] (also described in Section 3.2 above) of different sizes, train models based on pronunciations in the subsets and estimate model accuracies using the metric in Section 2.

We evaluate joint-sequence n-gram models for different values of n. The model qualities are observed to be near-identical for $n > 3$, so we note observations on just $n \in \{2, 3, 5\}$ here. Our RNN-transducer ('rnnt') models uses an encoder with 3 LSTM layers, each with 256 units. We used dropout [14] with a keep value of 0.9. The decoder network also uses 3 LSTM layers, each with 256 units, but with dropout keep value of 0.6.

We do training and evaluation on disjoint subsets, and in each case use a reliable amount of data for evaluation (See section 2). We repeat the experiment for training subset sizes from $\{100, 150, 200, 250, 300, 350, 400, 450, 500, 750, 1000\}$ for the different models. For rnnt models, we only run for $\{500, 750, 1000\}$ as the models don't readily converge for lower values. We perform the experiment on a large range of languages, and present observations from a representative subset below.

4.2. Evaluation

We note that 2-grams actually perform better than more complicated models in certain languages (for instance Kannada and Finnish in Figure 2) when we have very few words to train on. Similarly rnnt models work poorly in this domain even though they are known to be at par, or even better, with larger amounts of data. This indicates that more sophisticated models tend to overfit and generalize at the wrong level when given less data, and hence in a language with highly regular pronunciations like Finnish they are likely to learn complex rules from exceptions in training data and make more mistakes.

We can use the slope of the training curves in Figure 2 to determine whether adding more words will help with accuracy. For example, by observing the plots it's fair to deduce that English would benefit the most by transcribing more words and

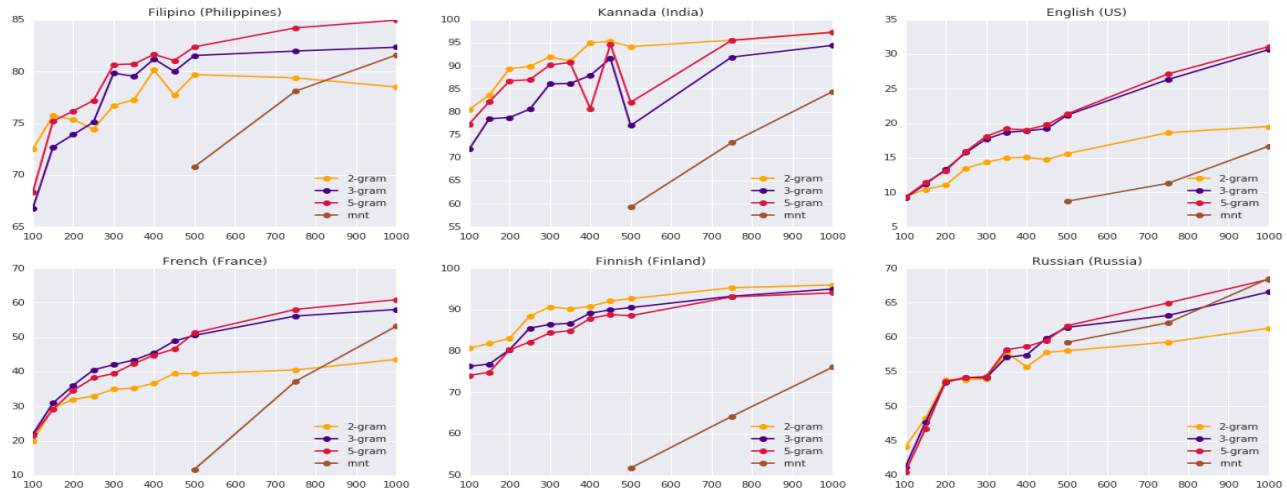

Figure 2: *Accuracy estimates for different training algorithms on different subset sizes.*

Finnish/Kannada would benefit the least. This can be tracked to allocate resources for transcription more judiciously while simultaneously scaling to multiple low resource languages. We can frame this as a simple submodular optimization problem [15] to get the maximum quality boost for our budget, and use greedy selection to get near-optimal results with theoretical guarantees [16] [17].

Based on the experiments above we conclude that the best model varies with language and amount of training data. While some trends can be guessed based on linguistic properties of the language, this is especially hard for the more black-box neural models. Thus it is useful to test out a bunch of different models using a robust, reliable metric and select the model with the highest accuracy.

5. Active G2P Training

5.1. Best known (offline) approach

We implemented a few known and new heuristics for offline optimization, but the algorithm in [12] , also described in Section 3.2 above, seems to outperform all.

5.2. New (active) approach

We describe a novel active learning approach to G2P training to determine successive batches for transcription and use them to train a G2P model. Empirical evaluation shows the new approach is as good as the offline approach in general, and can be significantly better in the low-resource setting.

We start with transcribing K_0 (250) words selected with the offline approach and train an n-gram G2P ($n = 5$) on these. We use a simple aligner for grapheme chunks and phoneme chunks to obtain a set of 'graphone' alignments for any word-pronunciation pair built in the bottom-up manner of [18]. For example an aligner with mappings [*a;{ a;A c;k s;s s;S sh;S h;*ϵ] (here ϵ indicates no phoneme, remaining phonemes are in the X-SAMPA notation) will give two alignments for the pair "cash, kAS", namely *c;k a;{ s;S h;*ϵ and *c;k a;{ sh;S*.

We perform alignments for all the transcribed word-pronunciation pairs as well as for G2P pronunciations and compute the graphone alignment errors (insertions+substitutions+deletions per occurrence in transcribed pair) for each graphone g, and scale by its frequency to get $wt_0(g)$. For unseen graphones we simply set $wt_0(g)$ to the frequency of the graphone in the alignments. This gives us an estimate for how error-prone a graphone is. To obtain the next subset we update equations similar to equations 3 and 4

$$w_j := \text{argmax}_{w \in V}\, cov_j(w),\ V = V \setminus \{w_j\} \tag{6}$$

$$wt_{j+1}(n) = \alpha wt_j(n),\ \forall\, n \in N(w_j) \tag{7}$$

$$wt_{j+1}(g) = \beta wt_j(g),\ \forall\, g \in G(w_j) \tag{8}$$

$$cov_{j+1}(w) = \sum_{n \in N(w)} wt_{j+1}(n) + \sum_{g \in G(w)} wt_{j+1}(g) \tag{9}$$

where $G(w)$ is the set of all graphone alignments for w, $\alpha = 0.2$ and $\beta = 0.5$. This way we can add the words containing orthographic segments with more potential alignments (we use truncation of at most 4 alignments per word to handle the potential exponential blow up).

5.3. Experiments

Transcriptions are simulated by simply looking up the pronunciation lexicon for the subset of words selected by the algorithm for transcription. We compute accuracies for the two approaches for subset sizes $500, 1000, 1500, 2000$.

Table 1: *Accuracy (%) active vs. best offline*

Lang\Size	500	1000	1500	2000
French (France)	**55/49**	**62/61**	**69/67**	72/72
Spanish (Spain)	**40/36**	**48/44**	**53/51**	58/57
Filipino (Phl.)	82/80	84/84	87/86	88/88
English (US)	**23/20**	**31/27**	**38/30**	**45/37**
Kannada (India)	91/92	96/97	97/97	98/98
Bengali (India)	63/68	68/72	75/76	78/78
Hindi (India)	29/45	38/54	49/59	59/62

For comparing the performance of the two approaches, we organize the accuracies in Table 1. We observe a significant improvement in accuracy by using the novel approach, especially in French, Spanish and English. The pattern is particularly prominent in low transcription range, and tends to wane with higher amounts of training data. The new approach actively identifies transcription errors (important, for instance, if one uses L1 speakers instead of linguists for transcription) and exceptional alignments, and we try to learn more words to learn the correct mapping for the error-prone graphones.

We note that obtaining correct graphone alignment can be a challenge for languages like Hindi (or Bengali) where there can be multiple words with correct Unicode Normalization corresponding to the same visual rendering. For example, Bengali letter 'RA' (U+09B0) is sometimes typed as Bengali letter 'BA' (U+09AC) followed by Bengali nukta sign (U+09BC). Often these are hard to clean or filter by normalization. As a result, using the graphone-based approach can actually slow down learning as we look at 'spurious ambiguities' in alignment.

6. Conclusion and Future Work

It's interesting to note how we can use novel heuristics presented here to better handle challenges posed by low-resource language G2P training and evaluation. We should, and are able to, do a more robust evaluation than verification on a small random subset. We also note that simpler models can beat more sophisticated models in the low resource setting, as they are less likely to overfit.

We plan to use and potentially improve these ideas to scale G2P evaluation and training, and are already looking at applying these to TTS and ASR systems for South Asian regional languages. We frame the transcription task as a correction task for the G2P pronunciations, which will likely make it faster and yield a better transcription.

Finally we note that although the work here describes the grapheme-to-phoneme problem, most of the paradigms introduced here easily generalize to sequence alignment problems in low data setting. It should be interesting to consider application of the ideas presented here to other such settings, especially in the speech technology domain. Another interesting direction to consider is to extend this study to other relevant metrics like edit-distance based phoneme error rates.

7. Acknowledgements

Thanks to our colleagues Emily Kaplan, Daan van Esch and Martin Jansche, interactions with them helped shape this work.

8. References

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[2] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[3] M. Davel and E. Barnard, "Bootstrapping in language resource generation," in *Fourteenth Annual Symposium of the Pattern Recognition Association of South Africa*. Citeseer, 2003, p. 97.

[4] A. Bruguier, D. Gnanapragasam, L. Johnson, K. Rao, and F. Beaufays, "Pronunciation learning with rnn-transducers," *Proc. Interspeech 2017*, pp. 2556–2560, 2017.

[5] F. Beaufays, A. Sankar, M. Weintraub, and S. Williams, "Method and system for learning linguistically valid word pronunciations from acoustic data," Sep. 4 2007, u.S. Patent 7,266,495.

[6] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4225–4229.

[7] A. Deri and K. Knight, "Grapheme-to-phoneme models for (almost) any language," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2016, pp. 399–408.

[8] M. Davel, E. Barnard, C. v. Heerden, W. Hartmann, D. Karakos, R. Schwartz, and S. Tsakalidis, "Exploring minimal pronunciation modeling for low resource languages," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] J. Kominek and A. W. Black, "Learning pronunciation dictionaries: language complexity and word selection strategies," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 232–239.

[10] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech communication*, vol. 50, no. 5, pp. 434–451, 2008.

[11] K. Yao and G. Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[12] Y.-B. Kim and B. Snyder, "Optimal data set selection: An application to grapheme-to-phoneme conversion," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1196–1205.

[13] S. Toshniwal and K. Livescu, "Read, attend and pronounce: An attention-based approach for grapheme-to-phoneme conversion," in *Workshop on Machine Learning in Speech and Language Processing (MLSLP), Interspeech*, 2016.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] S. Fujishige, *Submodular functions and optimization*. Elsevier, 2005, vol. 58.

[16] D. Sharma, A. Kapoor, and A. Deshpande, "On greedy maximization of entropy," in *International Conference on Machine Learning*, 2015, pp. 1330–1338.

[17] J. Altschuler, A. Bhaskara, G. Fu, V. Mirrokni, A. Rostamizadeh, and M. Zadimoghaddam, "Greedy column subset selection: New bounds and distributed algorithms," in *International Conference on Machine Learning*, 2016, pp. 2539–2548.

[18] M. Jansche, "Computer-aided quality assurance of an icelandic pronunciation dictionary." in *LREC*, 2014, pp. 2111–2114.