Grapheme-to-Phoneme Models for (Almost) Any Language

Aliya Deri and Kevin Knight
Information Sciences Institute
Department of Computer Science
University of Southern California
{aderi, knight}@isi.edu

Abstract

Grapheme-to-phoneme (g2p) models are rarely available in low-resource languages, as the creation of training and evaluation data is expensive and time-consuming. We use Wiktionary to obtain more than 650k word-pronunciation pairs in more than 500 languages. We then develop phoneme and language distance metrics based on phonological and linguistic knowledge; applying those, we adapt g2p models for high-resource languages to create models for related low-resource languages. We provide results for models for 229 adapted languages.

1 Introduction

Grapheme-to-phoneme (g2p) models convert words into pronunciations, and are ubiquitous in speech- and text-processing systems. Due to the diversity of scripts, phoneme inventories, phonotactic constraints, and spelling conventions among the world's languages, they are typically language-specific. Thus, while most statistical g2p learning methods are language-agnostic, they are trained on language-specific data—namely, a pronunciation dictionary consisting of word-pronunciation pairs, as in Table 1.

Building such a dictionary for a new language is both time-consuming and expensive, because it requires expertise in both the language and a notation system like the International Phonetic Alphabet, applied to thousands of word-pronunciation pairs. Unsurprisingly, resources have been allocated only to the most heavily-researched languages. GlobalPhone, one of the most extensive multilingual text and speech databases, has pronunciation dictionaries in only 20 languages (Schultz et al., 2013)[1].

[1]We have been unable to obtain this dataset.

lang	word	pronunciation
eng	anybody	ẹ n iː b ɒ d iː
pol	żołądka	ʐ�o w o n̪ t̪ k a
ben	শাক	ʂ ɔ k ʈ ɔ
heb	חלומות	ʁ a l o m o t

Table 1: Examples of English, Polish, Bengali, and Hebrew pronunciation dictionary entries, with pronunciations represented with the International Phonetic Alphabet (IPA).

word	eng	deu	nld
gift	g ɪ f t^{h}	g ɪ f t	ɣ ɪ f t
class	k^{h} l æ s	k l aː s	k l ɑ s
send	s ẹ n d	z ɛ n t	s ɛ n t

Table 2: Example pronunciations of English words using English, German, and Dutch g2p models.

For most of the world's more than 7,100 languages (Lewis et al., 2009), no data exists and the many technologies enabled by g2p models are inaccessible.

Intuitively, however, pronouncing an unknown language should not necessarily require large amounts of language-specific knowledge or data. A native German or Dutch speaker, with no knowledge of English, can approximate the pronunciations of an English word, albeit with slightly different phonemes. Table 2 demonstrates that German and Dutch g2p models can do the same.

Motivated by this, we create and evaluate g2p models for low-resource languages by adapting existing g2p models for high-resource languages using linguistic and phonological information. To facilitate our experiments, we create several notable data resources, including a multilingual pronunciation dictionary with entries for more than 500 languages.

The contributions of this work are:

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 399–408,
Berlin, Germany, August 7-12, 2016.

- Using data scraped from Wiktionary, we clean and normalize pronunciation dictionaries for 531 languages. To our knowledge, this is the most comprehensive multilingual pronunciation dictionary available.
- We synthesize several named entities corpora to create a multilingual corpus covering 384 languages.
- We develop a language-independent distance metric between IPA phonemes.
- We extend previous metrics for language-language distance with additional information and metrics.
- We create two sets of g2p models for "high resource" languages: 97 simple rule-based models extracted from Wikipedia's "IPA Help" pages, and 85 data-driven models built from Wiktionary data.
- We develop methods for adapting these g2p models to related languages, and describe results for 229 adapted models.
- We release all data and models.

2 Related Work

Because of the severe lack of multilingual pronunciation dictionaries and g2p models, different methods of rapid resource generation have been proposed.

Schultz (2009) reduces the amount of expertise needed to build a pronunciation dictionary, by providing a native speaker with an intuitive rule-generation user interface. Schlippe et al. (2010) crawl web resources like Wiktionary for word-pronunciation pairs. More recently, attempts have been made to automatically extract pronunciation dictionaries directly from audio data (Stahlberg et al., 2016). However, the requirement of a native speaker, web resources, or audio data specific to the language still blocks development, and the number of g2p resources remains very low. Our method avoids these issues by relying only on text data from high-resource languages.

Instead of generating language-specific resources, we are instead inspired by research on cross-lingual automatic speech recognition (ASR) by Vu and Schultz (2013) and Vu et al. (2014), who exploit linguistic and phonetic relationships in low-resource scenarios. Although these works focus on ASR instead of g2p models and rely on audio data, they demonstrate that speech technology is portable across related languages.

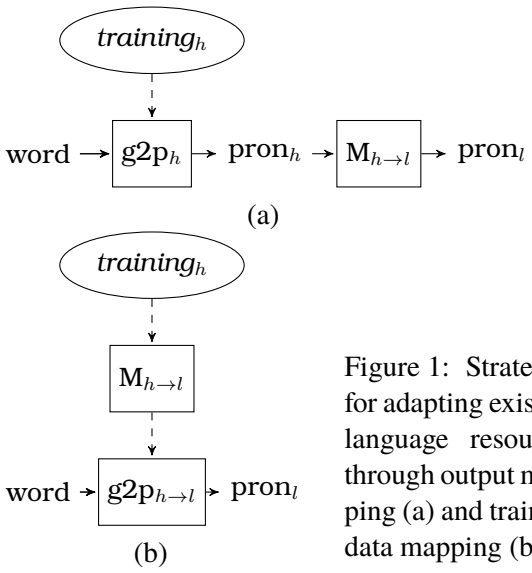

Figure 1: Strategies for adapting existing language resources through output mapping (a) and training data mapping (b).

3 Method

Given a low-resource language l without g2p rules or training data, we adapt resources (either an existing g2p model or a pronunciation dictionary) from a high-resource language h to create a g2p for l. We assume the existence of two modules: a phoneme-to-phoneme distance metric `phon2phon`, which allows us to map between the phonemes used by h to the phonemes used by l, and a closest language module `lang2lang`, which provides us with related language h.

Using these resources, we adapt resources from h to l in two different ways:

- Output mapping (Figure 1a): We use g2p$_h$ to pronounce word$_l$, then map the output to the phonemes used by l with `phon2phon`.
- Training data mapping (Figure 1b): We use `phon2phon` to map the pronunciations in h's pronunciation dictionary to the phonemes used by l, then train a g2p model using the adapted data.

The next sections describe how we collect data, create phoneme-to-phoneme and language-to-language distance metrics, and build high-resource g2p models.

4 Data

This section describes our data sources, which are summarized in Table 3.

4.1 Phoible

Phoible (Moran et al., 2014) is an online repository of cross-lingual phonological data. We use

<table>
<tr><td>Phoible</td><td>Wiki IPA Help tables</td><td colspan="2">Wiktionary</td></tr>
<tr><td>1674 languages
2155 lang. inventories
2182 phonemes
37 features</td><td rowspan="4">97 languages
24 scripts
1753 graph. segments
1534 phon. segments
3389 unique g-p rules</td><td colspan="2">531 languages
49 scripts
658k word-pron pairs</td></tr>
<tr><td>NE data</td><td>Wiktionary train</td><td>Wiktionary test</td></tr>
<tr><td>384 languages
36 scripts
9.9m NEs</td><td>85 languages
42 scripts
629k word-pron pairs</td><td>501 languages
45 scripts
26k word-pron pairs</td></tr>
</table>

Table 3: Summary of data resources obtained from Phoible, named entity resources, Wikipedia IPA Help tables, and Wiktionary. Note that, although our Wiktionary data technically covers over 500 languages, fewer than 100 include more than 250 entries (Wiktionary train).

two of its components: language phoneme inventories and phonetic features.

4.1.1 Phoneme inventories

A phoneme inventory is the set of phonemes used to pronounce a language, represented in IPA. Phoible provides 2156 phoneme inventories for 1674 languages. (Some languages have multiple inventories from different linguistic studies.)

4.1.2 Phoneme feature vectors

For each phoneme included in its phoneme inventories, Phoible provides information about 37 phonological features, such as whether the phoneme is nasal, consonantal, sonorant, or a tone. Each phoneme thus maps to a unique feature vector, with features expressed as `+`, `-`, or `0`.

4.2 Named Entity Resources

For our language-to-language distance metric, it is useful to have written text in many languages. The most easily accessible source of this data is multilingual named entity (NE) resources.

We synthesize 7 different NE corpora: Chinese-English names (Ji et al., 2009), Geonames (Vatant and Wick, 2006), JRC names (Steinberger et al., 2011), corpora from LDC[2], NEWS 2015 (Banchs et al., 2015), Wikipedia names (Irvine et al., 2010), and Wikipedia titles (Lin et al., 2011); to this, we also add multilingual Wikipedia titles for place names from an online English-language gazetteer (Everett-Heath, 2014). This yields a list of 9.9m named entities (8.9 not including English data) across 384 languages, which include the English translation, named entity type, and script information where possible.

4.3 Wikipedia IPA Help tables

To explain different languages' phonetic notations, Wikipedia users have created "IPA Help" pages,[3] which provide tables of simple grapheme examples of a language's phonemes. For example, on the English page, the phoneme z has the examples "**z**oo" and "ha**s**." We automatically scrape these tables for 97 languages to create simple grapheme-phoneme rules.

Using the `phon2phon` distance metric and mapping technique described in Section 5, we clean each table by mapping its IPA phonemes to the language's Phoible phoneme inventory, if it exists. If it does not exist, we map the phonemes to valid Phoible phonemes and create a phoneme inventory for that language.

4.4 Wiktionary pronunciation dictionaries

Ironically, to train data-driven g2p models for high-resource languages, and to evaluate our low-resource g2p models, we require pronunciation dictionaries for many languages. A common and successful technique for obtaining this data (Schlippe et al., 2010; Schlippe et al., 2012a; Yao and Kondrak, 2015) is scraping Wiktionary, an open-source multilingual dictionary maintained by Wikimedia. We extract unique word-pronunciation pairs from the English, German, Greek, Japanese, Korean, and Russian sites of Wiktionary. (Each Wiktionary site, while written in its respective language, contains word entries in multiple languages.)

[2]LDC2015E13, LDC2015E70, LDC2015E82, LDC2015E90, LDC2015E84, LDC2014E115, and LDC2015E91

[3]`https://en.wikipedia.org/wiki/Category:International_Phonetic_Alphabet_help`

Since Wiktionary data is very noisy, we apply length filtering as discussed by Schlippe et al. (2012b), as well as simple regular expression filters for HTML. We also map Wiktionary pronunciations to valid Phoible phonemes and language phoneme inventories, if they exist, as discussed in Section 5. This yields 658k word-pronunciation pairs for 531 languages. However, this data is not uniformly distributed across languages—German, English, and French account for 51% of the data.

We extract test and training data as follows: For each language with at least 1 word-pron pair with a valid word (at least 3 letters and alphabetic), we extract a test set of a maximum of 200 valid words. From the remaining data, for every language with 50 or more entries, we create a training set with the available data.

Ultimately, this yields a training set with 629k word-pronunciation pairs in 85 languages, and a test set with 26k pairs in 501 languages.

5 Phonetic Distance Metric

Automatically comparing pronunciations across languages is especially difficult in text form. Although two versions of the "sh" sound, "ʃ" and "ɕ," sound very similar to most people and very different from "m," to a machine all three characters seem equidistant.

Previous research (Özbal and Strapparava, 2012; Vu and Schultz, 2013; Vu et al., 2014) has addressed this issue by matching exact phonemes by character or manually selecting comparison features; however, we are interested in an automatic metric covering all possible IPA phoneme pairs.

We handle this problem by using Phoible's phoneme feature vectors to create `phon2phon`, a distance metric between IPA phonemes. In this section we also describe how we use this metric to clean open-source data and build phoneme-mapping models between languages.

5.1 `phon2phon`

As described in Section 4.1.2, each phoneme in Phoible maps to a unique feature vector; each feature value is `+`, `-`, or `0`, representing whether a feature is present, not present, or not applicable. (Tones, for example, can never be syllabic or stressed.)

We convert each feature vector into a bit representation by mapping each value to 3 bits. `+` to `110`, `-` to `101`, and `0` to `000`. This captures the idea that the features `+` and `-` are more similar than `0`.

lang	word	scraped	cleaned
ces	jód	ˈjoːd	j o d
pus	څلور	t͡saˈlor	t s a l o r
kan	ಭಾರತ	bhārata	b h a ɾ a ṭ a
hye	օդապարր	otʰaˈpɑɾ	o̤ t̤ʰ a p a l̤
ukr	тарган	tɑrˈɦɑn	t̪ a ɾ̪ h a n̪

Table 4: Examples of scraped and cleaned Wiktionary pronunciation data in Czech, Pashto, Kannada, Armenian, and Ukrainian.

Data: all phonemes P, scraped phoneme set S, language inventory T
Result: Mapping table M
initialize empty table M;
for p_s *in* S **do**
  if $p_s \notin P$ *and* ASCII$(p_s) \in P$ **then**
    $p_s = \text{ASCII}(p_s)$;
  end
  $p_p = \min_{\forall_{p_t \in T}} (\texttt{phon2phon}(p_s, p_t))$;
  add $p_s \rightarrow p_p$ to M;
end

Algorithm 1: A condensed version of our procedure for mapping scraped phoneme sets from Wikipedia and Wiktionary to Phoible language inventories. The full algorithm handles segmentation of the scraped pronunciation and heuristically promotes coverage of the Phoible inventory.

We then compute the normalized Hamming distance between every phoneme pair $p_{1,2}$ with feature vectors $f_{1,2}$ and feature vector length n as follows:

$$\texttt{phon2phon}(p_1, p_2) = \frac{\sum_{i=1}^{n} 1, \text{if} f_1^i \neq f_2^i}{n}$$

5.2 Data cleaning

We now combine `phon2phon` distances and Phoible phoneme inventories to map phonemes from scraped Wikipedia IPA help tables and Wiktionary pronunciation dictionaries to Phoible phonemes and inventories. We describe a condensed version of our procedure in Algorithm 1, and provide examples of cleaned Wiktionary output in Table 4.

5.3 Phoneme mapping models

Another application of `phon2phon` is to transform pronunciations in one language to another language's phoneme inventory. We can do this by

lang	avg	phon	script
English	German	Latin	French
Hindi	Gujarati	Bengali	Sanskrit
Vietnamese	Indonesian	Sindhi	Polish

Table 5: Closest languages with Wikipedia versions, based on `lang2lang` averaged metrics, phonetic inventory distance, and script distance.

creating a single-state weighted finite-state transducer (wFST) W for input language inventory I and output language inventory O:

$$\forall_{p_i \in I, p_o \in O} W.\text{add}(p_i, p_o, 1 - \texttt{phon2phon}(p_i, p_o))$$

W can then be used to map a pronunciation to a new language; this has the interesting effect of modeling accents by foreign-language speakers: *think* in English (pronounced "θ ɪ ŋ k^h") becomes "s̪ ɛ ŋ k" in German; the capital city *Dhaka* (pronounced in Bengali with a voiced aspirated "d̪ʱ") becomes the unaspirated "d æ k^h æ" in English.

6 Language Distance Metric

Since we are interested in mapping high-resource languages to low-resource related languages, an important subtask is finding the related languages of a given language.

The URIEL Typological Compendium (Littell et al., 2016) is an invaluable resource for this task. By using features from linguistic databases (including Phoible), URIEL provides 5 distance metrics between languages: genetic, geographic, composite (a weighted composite of genetic and geographic), syntactic, and phonetic. We extend URIEL by adding two additional metrics, providing averaged distances over all metrics, and adding additional information about resources. This creates `lang2lang`, a table which provides distances between and information about 2,790 languages.

6.1 Phoneme inventory distance

Although URIEL provides a distance metric between languages based on Phoible features, it only takes into account broad phonetic features, such as whether each language has voiced plosives. This can result in some non-intuitive results: based on this metric, there are almost 100 languages phonetically equivalent to the South Asian language Gujarati, among them Arawak and Chechen.

To provide a more fine-grained phonetic distance metric, we create a phoneme inventory distance metric using `phon2phon`. For each pair of language phoneme inventories $L_{1,2}$ in Phoible, we compute the following:

$$d(L_1, L_2) = \sum_{p_1 \in L_1} \min_{p_2 \in L_2} (\texttt{phon2phon}(p_1, p_2))$$

and normalize by dividing by $\sum_i d(L_1, L_i)$.

6.2 Script distance

Although Urdu is very similar to Hindi, its different alphabet and writing conventions would make it difficult to transfer an Urdu g2p model to Hindi. A better candidate language would be Nepali, which shares the Devanagari script, or even Bengali, which uses a similar South Asian script. A metric comparing the character sets used by two languages is very useful for capturing this relationship.

We first use our multilingual named entity data to extract character sets for the 232 languages with more than 500 NE pairs; then, we note that Unicode character names are similar for linguistically related scripts. This is most notable in South Asian scripts: for example, the Bengali ক, Gujarati ક, and Hindi क have Unicode names `BENGALI LETTER KA`, `GUJARATI LETTER KA`, and `DEVANAGARI LETTER KA`, respectively.

We remove script, accent, and form identifiers from the Unicode names of all characters in our character sets, to create a set of reduced character names used across languages. Then we create a binary feature vector f for every language, with each feature indicating the language's use of a reduced character (like `LETTER KA`). The distance between two languages $L_{1,2}$ can then be computed with a spatial cosine distance:

$$d(L_1, L_2) = 1 - \frac{f_1 \cdot f_2}{\|f_1\|_2 \, \|f_2\|_2}$$

6.3 Resource information

Each entry in our `lang2lang` distance table also includes the following features for the second language: the number of named entities, whether it is in Europarl (Koehn, 2005), whether it has its own Wikipedia, whether it is primarily written in the same script as the first language, whether it has an IPA Help page, whether it is in our Wiktionary test set, and whether it is in our Wiktionary training set.

Table 5 shows examples of the closest languages to English, Hindi, and Vietnamese, according to different lang2lang metrics.

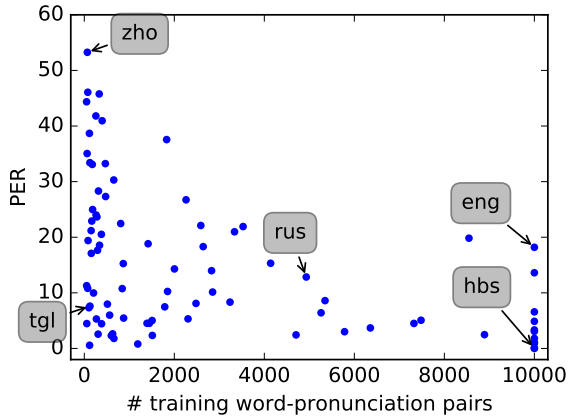

Figure 2: Training data size vs. PER for 85 models trained from Wiktionary. Labeled languages: English (eng), Serbo-Croatian (hbs), Russian (rus), Tagalog (tgl), and Chinese macrolanguage (zho).

7 Evaluation Metrics

The next two sections describe our high-resource and adapted g2p models. To evaluate these models, we compute the following metrics:

- % of words skipped: This shows the coverage of the g2p model. Some g2p models do not cover all character sequences. All other metrics are computed over non-skipped words.
- word error rate (WER): The percent of incorrect 1-best pronunciations.
- word error rate 100-best (WER 100): The percent of 100-best lists without the correct pronunciation.
- phoneme error rate (PER): The percent of errors per phoneme. A PER of 15.0 indicates that, on average, a linguist would have to edit 15 out of 100 phonemes of the output.

We then average these metrics across all languages (weighting each language equally).

8 High Resource g2p Models

We now build and evaluate g2p models for the "high-resource" languages for which we have either IPA Help tables or sufficient training data from Wiktionary. Table 6 shows our evaluation of these models on Wiktionary test data, and Table 7 shows results for individual languages.

8.1 IPA Help models

We first use the rules scraped from Wikipedia's IPA Help pages to build rule-based g2p models. We build a wFST for each language, with a path for each rule $g \rightarrow p$ and weight $w = 1/\text{count}(g)$. This method prefers rules with longer grapheme segments; for example, for the word *tin*, the output "ʃ n" is preferred over the correct "t^h ɪ n" because of the rule *ti*→ʃ. We build 97 IPA Help models, but have test data for only 91—some languages, like Mayan, do not have any Wiktionary entries.

As shown in Table 6, these rule-based models do not perform very well, suffering especially from a high percentage of skipped words. This is because IPA Help tables explain phonemes' relationships to graphemes, rather than vice versa. Thus, the English letter *x* is omitted, since its composite phonemes are better explained by other letters.

8.2 Wiktionary-trained models

We next build models for the 85 languages in our Wiktionary train data set, using the wFST-based Phonetisaurus (Novak et al., 2011) and MITLM (Hsu and Glass, 2008), as described by Novak et al (2012). We use a maximum of 10k pairs of training data, a 7-gram language model, and 50 iterations of EM.

These data-driven models outperform IPA Help models by a considerable amount, achieving a WER of 44.69 and PER of 15.06 averaged across all 85 languages. Restricting data to 2.5k or more training examples boosts results to a WER of 28.02 and PER of 7.20, but creates models for only 29 languages.

However, in some languages good results are obtained with very limited data; Figure 2 shows the varying quality across languages and data availability.

8.3 Unioned models

We also use our rule-based IPA Help tables to improve Wiktionary model performance. We accomplish this very simply, by prepending IPA help rules like the German *sch*→ʃ to the Wiktionary training data as word-pronunciation pairs, then running the Phonetisaurus pipeline.

Overall, the unioned g2p models outperform both the IPA help and Wiktionary models; however, as shown in Table 7, the effects vary across different languages. It is unclear what effect language characteristics, quality of IPA Help rules, and training data size have on unioned model improvement.

model	# langs	% skip	WER	WER 100	PER
ipa-help	91	21.49	78.13	59.18	35.36
wiktionary	85	4.78	44.69	23.15	15.06
unioned	85	3.98	44.17	21.97	14.70
ipa-help	56	22.95	82.61	61.57	35.51
wiktionary	56	3.52	40.28	20.30	13.06
unioned	56	2.31	39.49	18.51	12.52

Table 6: Results for high-resource models. The top portion of the table shows results for all models; the bottom shows results only for languages with both IPA Help and Wiktionary models.

lang	ben	tgl	tur	deu
# train	114	126	2.5k	10k
ipa-help	100.0	64.8	69.0	40.2
wikt	85.6	**34.2**	**39.0**	32.5
unioned	**66.2**	36.2	**39.0**	**24.5**

Table 7: WER scores for Bengali, Tagalog, Turkish, and German models. Unioned models with IPA Help rules tend to perform better than Wiktionary-only models, but not consistently.

9 Adapted g2p Models

Having created a set of high-resource models and our `phon2phon` and `lang2lang` metrics, we now explore different methods for adapting high-resource models and data for related low-resource languages. For comparable results, we restrict the set of high-resource languages to those covered by both our IPA Help and Wiktionary data.

9.1 No mapping

The simplest experiment is to run our g2p models on related low-resource languages, without adaptation. For each language l in our test set, we determine the top high-resource related languages $h_{1,2,...}$ according to the `lang2lang` averaged metric that have both IPA Help and Wiktionary data and the same script, not including the language itself. For IPA Help models, we choose the 3 most related languages $h_{1,2,3}$ and build a g2p model from their combined g-p rules. For Wiktionary and unioned models, we compile 5k words from the closest languages $h_{1,2,...}$ such that each h contributes no more than one third of the data (adding IPA Help rules for unioned models) and train a model from the combined data.

For each test word-pronunciation pair, we trivially map the word's letters to the characters used in $h_{1,2,...}$ by removing accents where necessary; we then use the high-resource g2p model to produce a pronunciation for the word. For example, our Czech IPA Help model uses a model built from g-p rules from Serbo-Croatian, Polish, and Slovenian; the Wiktionary and unioned models use data and rules from these languages and Latin as well.

This expands 56 g2p models (the languages covered by both IPA Help and Wiktionary models) to models for 211 languages. However, as shown in Table 8, results are very poor, with a very high WER of 92% using the unioned models and a PER of more than 50%. Interestingly, IPA Help models perform better than the unioned models, but this is primarily due to their high skip rate.

9.2 Output mapping

We next attempt to improve these results by creating a wFST that maps phonemes from the inventories of $h_{1,2...}$ to l (as described in Section 5.3). As shown in Figure 1a, by chaining this wFST to $h_{1,2...}$'s g2p model, we map the g2p model's output phonemes to the phonemes used by l. In each base model type, this process considerably improves accuracy over the no mapping approach; however, the IPA Help skip rate increases (Table 8).

9.3 Training data mapping

We now build g2p models for l by creating synthetic data for the Wiktionary and unioned models, as in Figure 1b. After compiling word-pronunciation pairs and IPA Help g-p rules from closest languages $h_{1,2,...}$, we then map the pronunciations to l and use the new pronunciations as training data. We again create unioned models by adding the related languages' IPA Help rules to the training data.

This method performs slightly worse in accuracy than output mapping, a WER of 87%, but has a much lower skip rate of 7%.

method	base model	# langs	% skip	WER	WER 100	PER
no mapping	ipa-help	211	12.46	**91.57**	**78.96**	**54.84**
	wikt	211	8.99	93.15	80.36	57.07
	unioned	211	**8.54**	92.38	79.26	57.21
output mapping	ipa-help	211	12.68	85.45	67.07	47.94
	wikt	211	15.00	86.48	66.20	46.84
	unioned	211	**11.72**	**84.82**	**63.63**	**46.25**
training data mapping	wikt	211	8.55	87.40	70.94	48.89
	unioned	211	**7.19**	**87.36**	**70.75**	**47.48**
rescripted	wikt	+10	15.94	**93.66**	81.76	**56.37**
	unioned	+10	**14.97**	94.45	**80.68**	57.35
final	wikt/unioned	229	6.77	88.04	69.80	48.01

Table 8: Results for adapted g2p models. Final adapted results (using the 85 languages covered by Wiktionary and unioned high-resource models, as well as rescripting) cover 229 languages.

lang	method	base model	rel langs	word	gold	hyp
eng	no mapping	ipa-help	deu, nld, swe	fuse	f j uː z	f ʏ s ɛ
arz	output mapping	unioned	fas, urd	بانجو	b æː n̪ g uː	b a n̪ d̪ ʃ uː
afr	training mapping	unioned	nld, lat, isl	dood	d ɔ t	d uː t
sah	training mapping	unioned	rus, bul, ukr	хатырык	k a t̪ ɯ r̪ ɯ k	k a t̪ i r̪ i k
kan	rescripted	unioned	hin, ben	ಡುಷ್ಟ	d̪ u ʂ ʈʰ a	d̪ uː ʂ ʈʰ
guj	rescripted	unioned	san, ben, hin	ક્રોએશિઆ	k ɾ o e ç ɪ a	k ɾ õː ə ʂ ɪ a

Table 9: Sample words, gold pronunciations, and hypothesis pronunciations for English, Egyptian Arabic, Afrikaans, Yakut, Kannada, and Gujarati.

9.4 Rescripting

Adaptation methods thus far have required that h and l share a script. However, this excludes languages with related scripts, like Hindi and Bengali.

We replicate our data mapping experiment, but now allow related languages $h_{1,2,...}$ with different scripts from l but a script distance of less than 0.2. We then build a simple "rescripting" table based on matching Unicode character names; we can then map not only h's pronunciations to l's phoneme set, but also h's word to l's script.

Although performance is relatively poor, rescripting adds 10 new languages, including Telugu, Gujarati, and Marwari.

9.5 Discussion

Table 8 shows evaluation metrics for all adaptation methods. We also show results using all 85 Wiktionary models (using unioned where IPA Help is available) and rescripting, which increases the total number of languages to 229. Table 9 provides examples of output with different languages.

In general, mapping combined with IPA Help rules in unioned models provides the best results. Training data mapping achieves similar scores as output mapping as well as a lower skip rate. Word skipping is problematic, but could be lowered by collecting g-p rules for the low-resource language.

Although the adapted g2p models make many individual phonetic errors, they nevertheless capture overall pronunciation conventions, without requiring language-specific data or rules. Specific points of failure include rules that do not exist in related languages (e.g., the silent "e" at the end of "fuse" and the conversion of "d̪ʃ" to "g" in Egyptian Arabic), mistakes in phoneme mapping, and overall "pronounceability" of the output.

9.6 Limitations

Although our adaptation strategies are flexible, several limitations prevent us from building a g2p model for any language. If there is not enough information about the language, our `lang2lang` table will not be able to provide related high-resource languages. Additionally, if the language's script is not closely related to another language's and thus cannot be rescripted (as with Thai and Armenian), we are not able to adapt related g2p data or models.

10 Conclusion

Using a large multilingual pronunciation dictionary from Wiktionary and rule tables from Wikipedia, we build high-resource g2p models and show that adding g-p rules as training data can improve g2p performance. We then leverage `lang2lang` distance metrics and `phon2phon` phoneme distances to adapt g2p resources for high-resource languages for 229 related low-resource languages. Our experiments show that adapting training data for low-resource languages outperforms adapting output. To our knowledge, these are the most broadly multilingual g2p experiments to date.

With this publication, we release a number of resources to the NLP community: a large multilingual Wiktionary pronunciation dictionary, scraped Wikipedia IPA Help tables, compiled named entity resources (including a multilingual gazetteer), and our `phon2phon` and `lang2lang` distance tables.[4]

Future directions for this work include further improving the number and quality of g2p models, as well as performing external evaluations of the models in speech- and text-processing tasks. We plan to use the presented data and methods for other areas of multilingual natural language processing.

11 Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments, as well as our colleagues Marjan Ghazvininejad, Jonathan May, Nima Pourdamghani, Xing Shi, and Ashish Vaswani for their advice. We would also like to thank Deniz Yuret for his invaluable help with data collection. This work was supported in part by DARPA (HR0011-15-C-0115) and ARL/ARO (W911NF-10-1-0533). Computation for the work described in this paper was supported by the University of Southern California's Center for High-Performance Computing.

References

Rafael E Banchs, Min Zhang, Xiangyu Duan, Haizhou Li, and A Kumaran. 2015. Report of NEWS 2015 machine transliteration shared task. In *Proc. NEWS Workshop*.

John Everett-Heath. 2014. *The Concise Dictionary of World Place-Names*. Oxford University Press, 2nd edition.

Bo-June Paul Hsu and James R Glass. 2008. Iterative language model estimation: efficient data structure & algorithms. In *Proc. Interspeech*.

Ann Irvine, Chris Callison-Burch, and Alexandre Klementiev. 2010. Transliterating from all languages. In *Proc. AMTA*.

Heng Ji, Ralph Grishman, Dayne Freitag, Matthias Blume, John Wang, Shahram Khadivi, Richard Zens, and Hermann Ney. 2009. Name extraction and translation for distillation. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. MT Summit*.

M Paul Lewis, Gary F Simons, and Charles D Fennig. 2009. *Ethnologue: Languages of the world*. SIL international, Dallas.

Wen-Pin Lin, Matthew Snover, and Heng Ji. 2011. Unsupervised language-independent name translation mining from Wikipedia infoboxes. In *Proc. Workshop on Unsupervised Learning in NLP*.

Patrick Littell, David Mortensen, and Lori Levin. 2016. URIEL. Pittsburgh: Carnegie Mellon University. `http://www.cs.cmu.edu/~dmortens/uriel.html`. Accessed: 2016-03-19.

Steven Moran, Daniel McCloy, and Richard Wright, editors. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Josef R Novak, D Yang, N Minematsu, and K Hirose. 2011. Phonetisaurus: A WFST-driven phoneticizer. *The University of Tokyo, Tokyo Institute of Technology*.

Josef R Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012. WFST-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding. In *Proc. International Workshop on Finite State Methods and Natural Language Processing*.

Gözde Özbal and Carlo Strapparava. 2012. A computational approach to the automation of creative naming. In *Proc. ACL*.

Tim Schlippe, Sebastian Ochs, and Tanja Schultz. 2010. Wiktionary as a source for automatic pronunciation extraction. In *Proc. Interspeech*.

Tim Schlippe, Sebastian Ochs, and Tanja Schultz. 2012a. Grapheme-to-phoneme model generation for Indo-European languages. In *Proc. ICASSP*.

Tim Schlippe, Sebastian Ochs, Ngoc Thang Vu, and Tanja Schultz. 2012b. Automatic error recovery for pronunciation dictionaries. In *Proc. Interspeech*.

[4] Instructions for obtaining this data are available at the authors' websites.

Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe. 2013. GlobalPhone: A multilingual text & speech database in 20 languages. In *Proc. ICASSP*.

Tanja Schultz. 2009. Rapid language adaptation tools and technologies for multilingual speech processing systems. In *Proc. IEEE Workshop on Automatic Speech Recognition*.

Felix Stahlberg, Tim Schlippe, Stephan Vogel, and Tanja Schultz. 2016. Word segmentation and pronunciation extraction from phoneme sequences through cross-lingual word-to-phoneme alignment. *Computer Speech & Language*, 35:234 – 261.

Ralf Steinberger, Bruno Pouliquen, Mijail Kabadjov, and Erik Van der Goot. 2011. JRC-Names: A freely available, highly multilingual named entity resource. In *Proc. Recent Advances in Natural Language Processing*.

Bernard Vatant and Marc Wick. 2006. Geonames ontology.
Online at http://www.geonames.org/ontology.

Ngoc Thang Vu and Tanja Schultz. 2013. Multilingual multilayer perceptron for rapid language adaptation between and across language families. In *Proc. Interspeech*.

Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek Motlicek, Tanja Schultz, and Hervé Bourlard. 2014. Multilingual deep neural network based acoustic modeling for rapid language adaptation. In *Proc. ICASSP*.

Lei Yao and Grzegorz Kondrak. 2015. Joint generation of transliterations from multiple representations. In *Proc. NAACL HLT*.