# Grapheme-to-Phoneme Transduction for Cross-Language ASR

Mark Hasegawa-Johnson[1(✉)], Leanne Rolston[2], Camille Goudeseune[1], Gina-Anne Levow[2], and Katrin Kirchhoff[3]

[1] University of Illinois, Champaign, USA
{jhasegaw,cog}@illinois.edu
[2] University of Washington, Seattle, USA
{rolston,levow}@uw.edu
[3] Amazon Alexa, Seattle, USA
katrin.kirchhoff@gmail.com

**Abstract.** Automatic speech recognition (ASR) can be deployed in a previously unknown language, in less than 24 h, given just three resources: an acoustic model trained on other languages, a set of language-model training data, and a grapheme-to-phoneme (G2P) transducer to connect them. The LanguageNet G2Ps were created with the goal of being small, fast, and easy to port to a previously unseen language. Data come from pronunciation lexicons if available, but if there are no pronunciation lexicons in the target language, then data are generated from minimal resources: from a Wikipedia description of the target language, or from a one-hour interview with a native speaker of the language. Using such methods, the LanguageNet G2Ps now include simple models in nearly 150 languages, with trained finite state transducers in 122 languages, 59 of which are sufficiently well-resourced to permit measurement of their phone error rates. This paper proposes a measure of the distance between the G2Ps in different languages, and demonstrates that agglomerative clustering of the LanguageNet languages bears some resemblance to a phylogeographic language family tree. The LanguageNet G2Ps proposed in this paper have already been applied in three cross-language ASRs, using both hybrid and end-to-end neural architectures, and further experiments are ongoing.

**Keywords:** Grapheme-to-phoneme transducers · Cross-language speech recognition · Automatic speech recognition · Under-resourced languages

## 1 Why IPA?

Imagine a small group of community organizers, trying to develop a spoken dialog system for the speakers of their language, using an open-source cross-language

This research was supported by the DARPA LORELEI program. Conclusions and findings are those of the authors, and are not endorsed by DARPA.

L. Espinosa-Anke et al. (Eds.): SLSP 2020, LNAI 12379, pp. 3–19, 2020.
https://doi.org/10.1007/978-3-030-59430-5_1

portability app. The first thing they might do is record examples of a few key words. If their language has a writing system (about 4000 languages do [17]), or if they have invented one [1], then they might write each word as they say it, expecting the app to use the same orthography to transcribe their speech in the future. The app creates an internal pronunciation model for each word, and reads the words back to them. After correcting its mistakes, they test it by narrating a few stories.

Such an app does not yet exist. Although the technologies necessary to create it are currently available, their error rates are still too high for casual uses. These technologies are, essentially, cross-linguistic automatic speech recognition (ASR) and cross-linguistic text-to-speech synthesis (TTS): ASR and TTS models that can be trained on several well-resourced languages, and then applied or adapted to a never-before-seen target language on the basis of one or two pronunciations, each, of a few dozen words. Every existing ASR or TTS paradigm with the potential to be applied, in such a scenario, uses the phone symbols of the international phonetic alphabet (IPA) [28] to organize the various sources of knowledge that need to be transferred from the training languages to the test language. This article discusses methods for converting text (graphemes) to phonemes (grapheme-to-phoneme transduction, or G2P) in a manner that can be extended to a previously unseen language in a few minutes using data that is usually available on Wikipedia or in elementary grammar primers.

The IPA is designed based on two key principles, which we might call the distinctive feature principle and the linguistic principle. The distinctive feature principle insists that IPA symbols should not be viewed as atomic, but rather, as "shorthand ways of indicating certain intersections of... natural classes of sounds that operate in phonological rules and historical sound changes" [34]. The interpretation of IPA phones as intersections of "distinctive features" (to use Ladefoged's term [34]) permits us to generalize from phones that we have seen (in one of the training languages) to novel phones (in the test language) by interpolating in the feature space [13], or by simply copying the acoustic parameters of an IPA phone from the training languages to the test language [51].

The linguistic principle, by contrast, limits the granularity with which the symbols of the IPA may sample distinctive feature space, by insisting that "the sounds that are represented are primarily those that distinguish one word from another" in at least one language [34]. The phonemes of any given language are the sounds that distinguish one word from another; the IPA phone symbols are intentionally designed to have only the granularity necessary so that every language's phoneme inventory can be written as a list of phones. The symbols of the IPA are therefore a "summary of agreed phonetic knowledge" [34] that can usefully prevent us from trying to model acoustic variability that is so small, or so context-dependent, that it never distinguishes words in any known language.

Because of the benefits of the distinctive feature principle and the linguistic principle, most cross-linguistic knowledge transfer, for speech technology applications, makes use of units that are indexed by IPA phones. Typically, acoustic spectra are clustered to form fenones [5], or triphone states are clustered to form senones [26] or projected onto a bottleneck feature space [21], each of which is considered to be the refinement of an IPA phone category. When speech technology needs to be rapidly developed in a previously unstudied language, some sort of knowledge-guided [51] or unsupervised [58] method is used to determine which of the IPA phones it should use. Models of those phones (including their component fenones, senones, bottleneck features, or Gaussian modes [31]) are then adapted from the training languages to the test language.

## 2 Related Research

Rule-based grapheme-to-phoneme transducers are as old as writing; for example, the Ashtádhyáyi of Páṅini is a sequence of context-dependent rules specifying the relationship between the grapheme sequence and the phoneme sequence of Sanskrit [55]. Prior to 1960, ASR used either whole-word models [12] or isolated phone models [16]. In 1961, Hughes used a pronunciation lexicon (a table matching the graphemic form of each word to its phonemic form) to measure phone error rate [25], and Peterson proposed using a similar table to automatically map recognized phone strings to recognized words [45]. A proposal to deal with out-of-vocabulary words by decomposing them into component graphemes and digraphs was published in 1963 [33], and the name "grapheme-to-phoneme translation" was given to this process in 1969 [35]. Weighted finite state transducers (WFSTs) for grapheme-to-phoneme translation were proposed in 1991 [20]. The joint-sequence modeling approach was proposed in [6], and refined in the software toolkit Phonetisaurus [42].

G2P transducers were developed for most of the languages of Europe in the 1970s and 1980s; G2Ps for Dutch, English and German were tested in the same speech synthesis system in 1988 [53]. WFST G2Ps were trained for seven languages in 1996 [47], and for 85 languages in 2016 [14]. The latter was tested on a 292-language corpus, which was further used to train a 311-language neural sequence-to-sequence G2P [44]. Apparently none of these efforts have been released as open source, but the 61-language rule-based open-source G2P `epitran` [40] is available at https://github.com/dmort27/epitran.

## 3 Training and Testing the G2Ps

This article introduces the LanguageNet G2P transducers, available under an MIT Open Source License from https://github.com/uiuc-sst/g2ps/. At the date of this writing, lookup tables for the most common graphemes and digraphs, derived from Wikipedia descriptions, are available for 142 languages. The dataset includes WFST transducers that have been trained and tested in Phonetisaurus [42] for 122 of these languages, using additional sources of data described below.

**Table 1.** In languages with no available pronunciation lexicon, G2Ps were trained using descriptions of their orthography copied from Wikipedia. Left: a copy of six lines from the table on the Wikipedia page, "French orthography." Right: lines from the table at left, reformatted into a simplified partial-word pronunciation lexicon that can be used to train a G2P.

| Spelling | | Major value (IPA) | Examples of major value |
|---|---|---|---|
| ç | | /s/ | ça, garçon, reçu |
| c | before e, i, y | /s/ | cyclone, loquace, ciel |
| | elsewhere | /k/ | cabas, crasse, lac |
| cc | before e, i, y | /ks/ | accès |
| | elsewhere | /k/ | accord |
| ch | | /ʃ/ | chat, douche |

| | |
|---|---|
| ç | s |
| ce | s ə |
| ci | s i |
| cy | s i |
| c | k |
| cce | k s ə |
| cc | k |
| ch | ʃ |

### 3.1 Data Collection

Three sources of data were used to train G2Ps in this article: Wikipedia symbol tables, LanguageNet open-source mined lexicons, and commercial lexicons.

The first source of data used to train the LanguageNet is a set of letter-to-sound rules, for each language, mined from Wikipedia. Wikipedia symbol tables were mined for each language by searching for entries of the form "<language> orthography" or "<language> alphabet." HTML tables on Wikipedia were reformatted into partial-word dictionaries, as shown in the last two columns of Table 1. Tables on Wikipedia do not provide information about letter-to-sound probabilities, but they often provide information about context: contexts are encoded as explicit digraphs and trigraphs (e.g., <ce,ci,cy,cce,cc,ch> in Table 1), while the unigraph entry (<c>→/k/ in Table 1) expresses the "elsewhere" case from the table on Wikipedia.

The second source of data used to train the LanguageNet is a set of pronunciation lexicons, mined incidentally during the collection of Rolston and Kirchhoff's master lexicon files (masterlex) [49]. The masterlexes are a set of bilingual translation dictionaries, mapping words from 103 non-English languages into their English near-equivalents. The data were mined semi-automatically from sources including Blench [7], Chaihana [19], Sözlük [11], IATE [27], wiktionary, ICD [48], OMWN [8], Panlex [29], TaaS [2], and a number of LDC sources. Each word, in each of the 103 source languages, is tagged with up to 11 attributes, depending on the type of information provided by the original data source: orthography, lemma, part of speech, transliteration, pronunciation, English translation, score, dialect, domain, data source, and morphological variants. Fields were populated only if a source provided the relevant information, and some fields were not usefully populated by any source. The pronunciation field was populated in about half of the available sources (40 out of 103). Entries with a pronunciation were used to train G2P transducers.

Finally, data from a number of other sources were used to train the G2P models. The Gulf Arabic model was trained using the Qatari Arabic Corpus [18], available at http://ifp-08.ifp.uiuc.edu/public/QAC/. Masterlex data on German, Dutch, and English were augmented with data from CELEX [4]. LDC corpora from BABEL and CALLHOME were also used to train G2Ps in the 24 BABEL languages, and in three of the CALLHOME languages.

### 3.2 Training the FSTs

Grapheme-to-phoneme finite state transducers (G2P FSTs) have been trained and tested thus far in 122 languages, using the Phonetisaurus [42] toolkit.

Phonetisaurus is based on graphone language modelling [6]. A graphone is defined to be an alignment between a sequence of graphemes and a sequence of phonemes. For example, the longest graphone in Table 1 is the trigraph-to-triphone alignment <cce>:/ksə/. Phonetisaurus does not permit 3-to-3 graphones of the form <cce>:/ksə/; instead, it requires each graphone to be either an $s_1$-to-1 or a 1-to-$s_2$ alignment, for $s_1 \leq S_1$ and $s_2 \leq S_2$, where $S_1$ and $S_2$ are user-defined parameters. Training proceeds as follows:

1. For each word in the lexicon, an initial graph of candidate alignments is created. The initial alignment graph contains all possible alignments of $s_1$ graphemes to one phoneme, and all alignments of one grapheme to $s_2$ phonemes, for $s_1 \leq S_1$ and $s_2 \leq S_2$. All such graphones are initially given equal probability.
2. Several iterations of the expectation maximization (EM) algorithm are used to re-estimate the probability of every graphone. After EM re-estimation, the Viterbi algorithm is used to compute the maximum likelihood graphone sequence for each word in the dictionary, which is printed out as training data for a graphone language model.
3. A graphone N-gram language model is trained using Kneser-Ney backoff [30], where the context length, $N$, is a user-specified parameter. The fully trained language model is then compiled into FST form using methods described in [3].

Language model backoff arcs prevent the FST from assigning zero probability to any sequence of known graphones. For example, if the graphone <ough>:/u/ occurs in the training lexicon only at the end of a word, then the learned lexicon can compute the test analysis <throughput>:/θrupʊt/ only by following a backoff arc from the end-of-word state to the start-of-word state, then following the unigram arc <p>:/p/. For this reason, it is possible to treat the Wikipedia symbol tables as if they were pronunciation lexicons; Phonetisaurus learns the mappings listed there, and learns backoff weights that permit them to be sequenced in any novel word.

Source data files (including Wikipedia, masterlex, and other sources) for each language were divided between training, development test, and evaluation test. Wikipedia symbol tables were assigned, in their entirety, to the training set. Other sources were divided: out of every five sequential entries, three were assigned to training, one to development, and one to evaluation. Graphone language models were trained using all hyperparameters in the range $S_1 \in \{2,3,4\}$, $S_2 \in \{2,3,4\}$, $N \in \{1,2,4,8\}$. Combinations with the lowest error rate on the development test set were then evaluated using the evaluation test set. Models were also trained, but not tested, for languages with no data source other than the Wikipedia symbol table (e.g., because the masterlex file contained no pronunciations) using the hyperparameters $S_1 = 2$, $S_2 = 2$, $N = 2$.

### 3.3 Testing the FSTs

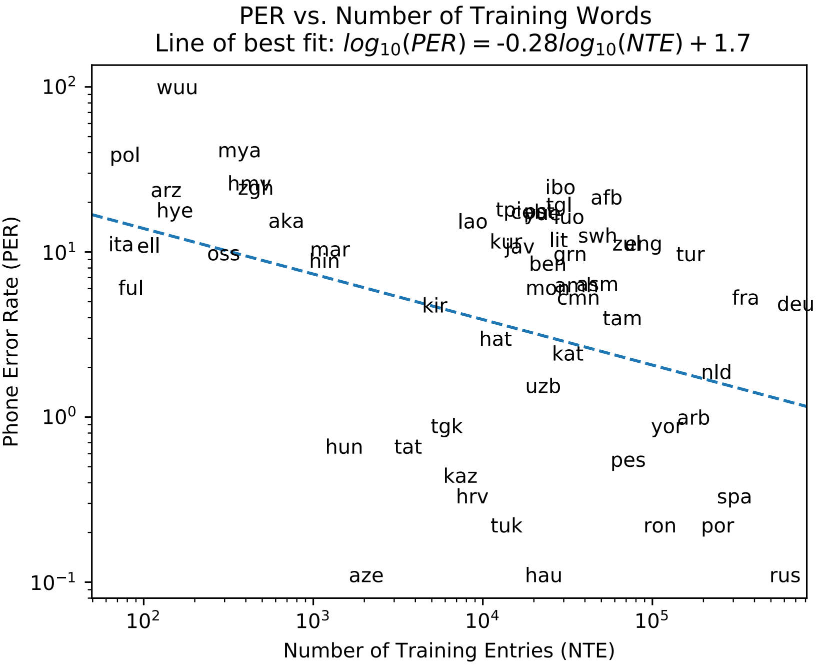

**Fig. 1.** Phone error rate of trained grapheme-to-phoneme transducers, as a function of the number of training words, for 59 languages. Each language is indicated by its ISO 639-3 code.

Figure 1 shows phone error rate (PER) on the evaluation test set, as a function of the number of training entries (NTE), for the 59 languages that have evaluation test data. Languages are labeled by their ISO 639-3 codes. PER is generally a decreasing function of NTE: the figure shows the line of best fit in log-log space, $\log_{10}(PER) = -0.28\log_{10}(NTE) + 1.7$. The figure shows some languages with extremely high PER, and some with extremely low PER. Post-hoc analysis suggests that most outliers are explained by one of two causes.

First, data sparsity: some of the languages with the highest PER are languages with complex grapheme systems, whose training datasets are insufficient to represent all of the characters in the orthography. The most obvious such example is Wu Chinese (ISO 639-3: wuu), which has a 90% PER. The training data contains almost 200 Chinese characters, and their phonetic pronunciations; none of the characters in the test dataset are in the training dataset. Similarly, Tamazight (zgh) includes both Latin and Tifinagh characters, Burmese (mya) includes both Latin and Burmese characters, and Hmong Dô (hmv) includes a variety of Latin-coded lexical tones, all of which result in a low degree of overlap between the training and test datasets.

Second, label noise: some of the pronunciation lexicons used as training material were apparently created not by humans, but by rule-based G2P transducers from the provided orthography. Automatically generated reference pronunciations result in unrealistically low or unrealistically high PER, depending on whether the distributed data include just one pronunciation per word (e.g., Azerbaijani (aze), Hausa (hau), and Russian (rus)), or two to three alternate pronunciations per word (Igbo (ibo), Tok Pisin (tpi) and Tagalog (tgl)). All of these G2Ps might be useful in a real application, but there is no way to be sure, because the evaluation test corpora were apparently constructed from the same algorithms as the training corpora.

Data sparsity and label noise seem to have less influence on the languages near the regression line. There is reason to believe, therefore, that the phone error rate of a G2P is reasonably well modeled as $10^{1.7-0.28\log_{10}(NTE)} \approx 50/\sqrt[3.5]{NTE}$.

## 4 Applications

The LanguageNet G2Ps were designed for zero-resource speech technology applications, e.g., for the purpose of deploying automatic speech recognition in a language for which no training data exist. Sections 4.2 and 4.3 describe two such systems. Section 4.1 explores the information the G2Ps have learned, by clustering the languages of the world according to a novel G2P distance proposed here.

### 4.1 Clustering Languages Based on Their G2Ps

A language family is a group of languages whose "divergent development…does not completely obscure the fact that these languages are descended from a common source" [23]. Comparative linguistics attempts to reconstruct the ancestral language by the analysis of regular relationships among word forms and syntax. In particular, the pronunciations of words change gradually over time, in a manner that "is remarkably regular…This fact is, of course, a great boon to historical linguists, since it makes the job of tracing linguistic forms through history much easier" [23]. By studying sound change in particular, one can not only reconstruct the ancestral language shared by two modern languages, but also estimate how many centuries have passed since they diverged, resulting in phylogeographic language family trees that can be used to infer the movements of peoples in prehistoric times [9,32] or that, conversely, may need to be modified or dissolved in the face of new scholarship [37]. Unlike pronunciation, though, graphemes often change in step discontinuities. Often, multiple scripts co-exist, but a government decision may suddenly cause documents in one particular script to become far more common, sometimes with the intention of promoting collaboration with a specified international community. Striking recent examples include the adoption of the Latin script for Turkish [24], Malay [43], and Uzbek [52], of Cyrillic for Kazakh and Kirghiz, and of the Arabic script for Uighur [38]. These considerations suggest the following hypothesis: The G2P transducers for two languages produce similar pronunciations, for any given written form, if the two languages are part of the same language family and use the same script, or if the two languages have both recently adapted their orthography from a common international origin.

Phonetisaurus trains G2Ps so that the cost of any given path is the joint probability of the orthographic word $w$ and its pronunciation $\pi$, so the G2P defines a joint probability mass function $p(w, \pi)$ over the set of all possible word-pronunciation pairs. The distance between two G2Ps $p$ and $q$ can usefully be defined as the expected distance between the pronunciations $\pi$ and $\rho$ that they produce in response to the same orthographic form $w$, averaged over the orthographic forms of both languages:

$$D_{G2P}(p,q) = \frac{1}{2} \mathop{\mathbb{E}}_{p(w,\pi)q(\rho|w)} [D_{PRON}(\pi,\rho)] + \frac{1}{2} \mathop{\mathbb{E}}_{q(w,\pi)p(\rho|w)} [D_{PRON}(\pi,\rho)] \quad (1)$$

The distance between two pronunciations should be proportional to the string edit distance between their phone strings $\pi = \{\pi_1, \ldots, \pi_K\}$ and $\rho = \{\rho_1, \ldots, \rho_L\}$. String edit distance is symmetric if deletion and insertion are treated identically as the pairwise phone distances $D_{PH}(\pi_k, \varnothing) = D_{PH}(\varnothing, \rho_l) = 1$ between any non-null phone symbol, $\pi_k$ or $\rho_l$, and the null-phone symbol $\varnothing$. Let $I_M(\pi)$ be an operator that inserts $M - K$ copies of the null phone between the elements of $\pi$, resulting in a string $\tilde{\pi} = \{\tilde{\pi}_1, \ldots, \tilde{\pi}_M\}$. Then the normalized string edit distance can be written:

$$D_{PRON}(\pi,\rho) = \frac{1}{\max(K,L)} \min_{M} \min_{\tilde{\pi}=I_M(\pi)} \min_{\tilde{\rho}=I_M(\rho)} \sum_{m=1}^{M} D_{PH}(\tilde{\pi}_m, \tilde{\rho}_m) \quad (2)$$

Normalization by $\max(K, L)$ guarantees that $0 \leq D_{PRON}(\pi, \rho) \leq 1$ if and only if $0 \leq D_{PH}(\pi_k, \rho_l) \leq 1$ for each pair of phone symbols $(\pi_k, \rho_l)$.

Ladefoged defines the distinctive features to be “natural classes of sounds that operate in phonological rules and historical sound changes” [34]; more precisely, historical sound changes tend to modify only a few of the distinctive features of a sound, while leaving the remainder unchanged. PHOIBLE [39] defines 37 distinctive features for 2908 different phones, including both simple phones (composed of a single IPA character, possibly with diacritics) and complex phones (composed of one or more IPA symbols in sequence, e.g., affricates and diphthongs). Possible values of each feature include the symbols [+], [−], blank (feature unspecified), and a variety of feature contours. For example, the dipthong /aɪ/ has the height feature [−+], meaning that the tongue moves from a [−high] position to a [+high] position. LanguageNet augments these 37 features with 8 features that can be easily specified for lexical tone sequences in all of the LanguageNet languages, and that are unspecified for all non-tonal IPA symbols: 4 tone height features (topTone, highTone, lowTone, bottomTone) and 4 tone contour features (riseTone, fallTone, hatTone, dipTone). The concatenation of the segmental and tonal features yields a total of $D = 45$ distinctive features per phone. Let each phone $\pi_k$ be a vector of such feature values, $\pi_k = [f_1(\pi_k), \ldots, f_D(\pi_k)]$. Then

$$D_{PH}(\pi_k, \rho_l) = \frac{1}{D} \sum_{d=1}^{D} \mathbb{1}\left(f_d(\pi_k) \neq f_d(\rho_l)\right) \tag{3}$$

where $\mathbb{1}()$ is the unit indicator function.

The 122 Phonetisaurus G2Ps of the LanguageNet were agglomeratively clustered, using nearest-centroid agglomeration [15], resulting in a complete binary phylogenetic tree over all of the 122 languages.[1] Expectations in Eq. 1 were approximated by selecting up to 1000 orthographic words from each of the two languages. Normalizations in Eq. 2 and 3 guarantee that the distance between any pair of languages is $0 \leq D_{G2P} \leq 1$. The distance at which any pair of clusters are merged is therefore an intuitively meaningful measure of the family relationship between the two languages. If two languages use completely different character sets (for example, one uses Cyrillic characters and one uses Arabic characters), then the G2P of one language is completely unable to process words from the other language, resulting in a distance very close to $D_{G2P}(p, q) \approx 1$. On the other hand, if the two languages produce very similar pronunciations in response to the same orthographic string, then the distance between their G2Ps is $D_{G2P}(p, q) \approx 0$.

[1] The complete tree is at github.com/uiuc-sst/g2ps/blob/master/g2ppy/cluster/agglomerative_cluster_output_2020-07-18.txt.

**Table 2.** Agglomerative clustering results: Clusters with internal distance $D_{G2P} \leq 0.2$. Numbers between rows show the distance separating the two clusters. Languages separated from the nearest cluster by $D_{G2P} > 0.2$ are not shown. Dashed horizontal lines indicate an inter-cluster distance of $D_{G2P} > 0.4$; solid horizontal lines indicate an inter-cluster distance of $D_{G2P} > 0.8$. Parentheses show the agglomerative structure within each cluster.

| | |
|---|---|
| Malayo-Polynesian, Bantu, Indo-Aryan: | (((((((Sundanese, Malay), Indonesian), Luba-Lulua), Kongo), Shona), Rohingya), Kinyarwanda) |
| | 0.202 |
| Cushitic, Polynesian: | ((Somali, Oromo), Fijian) |
| | 0.311 |
| Polynesian: | (Samoan, Tonga) |
| | 0.441 |
| Tahitic: | (Rarotongan, Maori) |
| | 0.404 |
| Finnic, Germanic: | ((Estonian, Finnish), Danish) |
| | 0.873 |
| South Slavic: | ((Serbian, Bosnian), Macedonian) |
| | 0.958 |
| Iranian: | (Dari, Persian) |

Table 2 shows all of the clusters that were merged at levels of $D_{G2P} \leq 0.2$. Other languages that joined each of these clusters at levels of $D_{G2P} > 0.2$ are not shown in the table, but the maximum distance threshold separating each pair of clusters is shown as a three-digit floating point number separating the corresponding rows. Several observations are salient.

- Each cluster is composed primarily, but not exclusively, of members of the same language family: Sundanese, Malay, and Indonesian are members of the Malayo-Polynesian family, while Dari and Persian are members of the Iranian family.
- Neighboring clusters tend to be from related language families. For example, Malayo-Polynesian, Polynesian, and Tahitic languages are spread across the first four clusters.
- Recent history sometimes trumps family relationships: the Danish G2P is similar to those of Estonian and Finnish, despite the lack of any family relationship.
- Script differences are marked by large inter-cluster distances, of $D_{G2P} = 0.873$ between the languages that use Latin vs. Cyrillic characters, and of $D_{G2P} = 0.958$ between those that use Cyrillic vs. Arabic characters. These distances are less than 1.0 only because some of the source dictionaries, in the Slavic and Iranian clusters, include small numbers of Latin-spelled words.
- Not all cluster results are well explained by family or historical relationships among languages. The largest cluster includes three Malayo-Polynesian

languages, four Bantu languages, and an Indo-Aryan language. These three language families share no common history, except that all eight languages have, during the twentieth century, developed national standards based on the Latin alphabet.

### 4.2 ASR24

By converting script into IPA phone symbols, one can build an ASR in a previously unknown language in about two hours. The ASR24 [22] cross-language ASR toolkit was built and tested for a number of such experiments. It was designed to solve the problem of recognizing speech in a language for which we have monolingual speech samples, monolingual texts (usually including a highly skewed assortment of religious texts and technical manuals, quickly but incompletely normalized), but no transcribed speech.

Without transcribed speech one cannot train an acoustic model. Therefore ASR24 uses pretrained acoustic models, from the English-language ASpIRE model [54] distributed by the maintainers of the Kaldi toolkit [46]. The phone set of the ASpIRE recognizer was mapped to IPA, so that its acoustic models can be appropriated for use in any language.

The target language's G2P is created by reformatting the Wikipedia description of its alphabet, and then running Phonetisaurus, as described in Sect. 3.2. If the target language lacks a Wikipedia description (as for Ilocano, at the time of the experiments described here), then its G2P is initialized using its closest related language in LanguageNet, and then refined on the basis of one hour of interaction with a paid native speaker of the target language. If the language's character set is not in LanguageNet, and if its Wikipedia description lacks some characters (e.g., Odia), then a symbol table is created from scratch, by asking a paid native speaker consultant to read each of the characters, and by transcribing that speech into IPA.

Test data are collected from an additional five hours of interaction with a paid native speaker consultant. The native speaker is asked to read texts in the target language. Although these texts are not sufficient to train the acoustic model, they are useful for estimating word error rate (WER). Available texts are divided into those used to train the language model (LM), and those read by the native speaker consultant as test material.

**Table 3.** Cross-language ASR experiments on seven languages. **Models** = salient details of LM or G2P (Trigram LM = from raw text, Alt LM = includes Brown clusters, clean = remove Bible stopwords and non-standard text, better truecasing, separate language-specific apostrophized affixes, Gaz = include words from a gazetteer of relevant place names). **Train time** = time required to clean monolingual text and train a language model (2 h and 8 h were maximum permitted wall-clock times, including data cleaning; some systems required less training time). **Build time** = wall-clock time required to compose LM with acoustic model (measured only for the first LM in each language). **Transcribe speed** = minutes of transcribed speech per minute of computation (measured only for the first LM in each language). **WER** = word error rate.

| Language | Models | Train Time (h) | Build Time (min) | Transcribe Speed (×RT) | WER |
|---|---|---|---|---|---|
| Somali | Trigram LM | 2 | 90 | 7 | 93.45% |
| | Alt LMs, clean | 8 | | | 84.58% |
| Hindi | Trigram+Gaz LM | 2 | 25 | 20 | 95.09% |
| | Alt LMs, clean | 8 | | | 93.71% |
| Zulu | Trigram+Gaz LM | 2 | 60 | 20 | 108.26% |
| | Alt LMs, clean | 8 | | | 90.22% |
| Sinhala | Trigram LM | 2 | 67 | 25 | 92.4% |
| | Alt LMs, clean | 8 | | | 93.5% |
| Kinyarwanda | Trigram LM | 2 | 76 | 23 | 88.1% |
| | Alt LMs, clean | 8 | | | 87.1% |
| Odia | Trigram+Gaz LM | 2 | 20 | 20 | 98% |
| | Alt LMs, clean | 8 | | | 106% |
| Ilocano | Tagalog G2P+Trigram | 2 | 30 | 20 | 93% |
| | Ilocano G2P+Trigram | 2 | | | 88% |
| | Alt LMs+Gaz, clean | 8 | | | 77% |

Experiments using this setup were performed for seven languages, five whose G2Ps were already in LanguageNet (Somali, Hindi, Zulu, Sinhala, and Kinyarwanda) and two that were not (Odia and Ilocano). Results are shown in Table 3. In all cases, build time (composition of the LM with the acoustic model) took 25 to 90 min, and transcription of novel audio was performed 20× faster than real time. Two checkpoints are listed. Checkpoint 1 used a trigram LM (trained in less than two hours). Checkpoint 2 used a class-based LM (if it gave lower perplexity than a trigram) trained after data cleaning (removing Bible stopwords, improved truecasing and sentence segmentation, separate language-dependent apostrophized function words from their neighbors). Word error rates (WER) are still quite high, but for the most part, they reduce substantially as a result of the six hours of extra modeling effort and data cleaning that were undertaken between checkpoints 1 and 2.

### 4.3 Discophone

End-to-end neural cross-language ASR experiments using LanguageNet's G2Ps were reported in [58] and [36]. In [58], speech recognizers were trained and tested using a transformer [56] sequence-to-sequence neural network implemented using the ESPnet [57] framework. In [36], speech recognizers were trained and tested using a listen, attend and spell architecture [10], implemented in the Dynet XNMT framework [41]. Speech included transcribed speech data from thirteen languages: five from the GlobalPhone distribution [50] (Czech, French, Spanish, Mandarin and Thai), and eight from the BABEL distribution (Cantonese, Bengali, Vietnamese, Lao, Zulu, Amharic, Javanese, and Georgian). The training subsets for each language varied from 11.5 h (Spanish) to 126.6 h (Cantonese). Training data transcriptions for 13 languages were converted to IPA using LanguageNet G2Ps. In [58], data were used to train three sets of speech recognizers per language: monolingual (trained and tested on the train and test subsets of the same language), multilingual (trained on all languages), and cross-lingual (trained on all languages except the test language). In [36], monolingual and multilingual systems were trained on only three tonal languages (Mandarin, Cantonese, and Vietnamese), and the cross-lingual setting used one hour of transcribed data from the test language (Lao) in order to adapt each recognizer.

An end-to-end phone recognizer, such as those trained and tested in [36,58], generates a sequence of IPA phone characters, with no further distinction between simple phones, diacritics, or complex phones. The reported error rate is therefore a new metric, which was named "phonetic token error rate" (PTER) in [58]: the string edit distance between the reference and hypothesis IPA character strings, counting the number of substitutions, deletions, and insertions of unicode IPA characters. PTER varied considerably among the 13 languages studied by [58], but in all 13 cases, the multilingual ASR was better than the monolingual ASR, and the cross-lingual ASR was worse than either. Multilingual PTER ranged from 8.1% (Czech) to 41% (Javanese). Cross-lingual PTER ranged from 61.7% (French) to 99.7% (again, Javanese).

In general, IPA tone symbols caused problems for the cross-lingual system in [58]. Mandarin had only 17.2% PTER in the multilingual setting, but had 85.9% PTER in the cross-lingual setting, because of incorrectly generated tones. Javanese is not a tone language, but most of its errors, in the cross-lingual setting, came from the incorrect insertion of IPA tone symbols. The problem of IPA tone symbols was studied in more depth by [36]. Four different models were considered. In the first model, the neural net was trained to output both phones and tones on the same output tier, as in [58]. In the second model, tones and phones were split into separate training and sequences, and the net learned to generate them on separate output tiers. The third model used all three of the tiers from models 1 and 2; the fourth model added a fourth tier, containing voice quality features. The 1-tier system was most successful if phones and tones were recombined into one stream prior to scoring. If phones and tones were scored separately, then the 4-tier model gave lowest error rates multilingually, but the 2-tier model was superior cross-lingually, suggesting that the simpler model might generalize better across language boundaries.

## 5 Conclusions

Creating ASR for all 7000 languages of the world requires methods that rapidly create a G2P for any new language. The methods proposed here create a G2P in about an hour, based on data from a standard alphabet table from Wikipedia or from a one-hour interview with a native speaker. Methods have also been developed to incorporate larger data sources into the G2P training pipeline, and have been applied for this purpose in 60 of the languages in the current distribution. Agglomerative clustering of the resulting G2Ps, using a novel G2P distance metric proposed here, results in clusters that tend to group together members of the same language family, with some exceptions. Cross-language ASRs using the LanguageNet G2Ps have been tested on 19 different languages: 7 using acoustic models trained on only one source language, and 13 using acoustic models each trained on 3 to 12 source languages (Zulu is in both sets). Word error rates and phonetic token error rates of cross-language ASR are high; ongoing research seeks methods that will reduce them.

## References

1. Adda, G., et al.: Breaking the unwritten language barrier: the BULB project. In: Proceedings of the SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced Languages (2016)
2. Aker, A., Paramita, M.L., Pinnis, M., Gaizauskas, R.J.: Bilingual dictionaries for all EU languages. In: Proceedings of the Conference on Language Resources and Evaluation (LREC), pp. 2839–2845 (2014)
3. Allauzen, C., Mohri, M., Roark, B.: Generalized algorithms for constructing statistical language models. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 40–47 (2003)
4. Baayen, R., Piepenbrock, R., Gulikers, L.: CELEX2. Technical report, LDC96L14, Linguistic Data Consortium (1996)
5. Bahl, L.R., Brown, P.F., de Souza, P.V., Picheny, M.A.: Acoustic Markov models used in the Tangora speech recognition system. In: Proceedings ICASSP, pp. 497–500 (1988)
6. Bisani, M., Ney, H.: Joint-sequence models for grapheme-to-phoneme conversion. Speech Commun. **50**(5), 434–451 (2008)
7. Blench, R., Nebel, A.: Dinka-English and English-Dinka dictionary (2005)
8. Bond, F., Paik, K.: A survey of wordnets and their licenses. Small **8**(4), 5 (2012)
9. Bouckaert, R., et al.: Mapping the origins and expansion of the Indo-European language family. Science **337**(6097), 957–960 (2012)
10. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In: Proceedings ICASSP, pp. 4960–4964 (2016). https://doi.org/10.1109/ICASSP.2016.7472621

11. Dâna, A.: Sözlük (2006). www.denizyuret.com/2006/11/turkish-resources.html. Accessed 20 July 2020
12. Davis, K., Biddulph, R., Balashek, S.: Automatic recognition of spoken digits. J. Acoust. Soc. Am. **24**(6), 637–642 (1952)
13. Deng, L.: Integrated-multilingual speech recognition using universal phonological features in a functional speech production model. In: Proceedings ICASSP (1997). https://doi.org/10.1109/ICASSP.1997.596110
14. Deri, A., Knight, K.: Grapheme-to-phoneme models for (almost) any language. In: Proceedings 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 399–408 (2016). https://doi.org/10.18653/v1/P16-1038
15. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2001)
16. Dudley, H., Balashek, S.: Automatic recognition of phonetic patterns in speech. J. Acoust. Soc. Am. **30**, 721–732 (1958)
17. Eberhard, D.M., Simons, G.F., Fennig, C.D. (eds.): Ethnologue: Languages of the World. 23rd edn. SIL International, Dallas (2020). www.ethnologue.com
18. Elmahdy, M., Hasegawa-Johnson, M., Mustafawi, E.: Development of a TV broadcasts speech recognition system for Qatari Arabic. In: Proceedings of the Conference on Language Resources and Evaluation (LREC), pp. 3057–3061 (2014)
19. Garrett, J., Lastowka, G., et al.: Turkmen-English dictionary: a SPA project of Peace Corps Turkmenistan (1996)
20. Gilloux, M.: Automatic learning of word transducers from examples. In: Proceedings EUROSPEECH, pp. 107–112 (1991)
21. Grézl, F., Karafiaát, M., Veselý, K.: Adaptation of multilingual stacked bottle-neck neural network structure for new language. In: Proceedings ICASSP, pp. 7704–7708 (2014)
22. Hasegawa-Johnson, M., Goudeseune, C., Levow, G.A.: Fast transcription of speech in low-resource languages (2019). https://arxiv.org/abs/1909.07285
23. Hock, H.H.: Principles of Historical Linguistics. Mouton de Gruyter, Berlin (1991)
24. Howard, D.A.: The History of Turkey. Greenwood, Santa Barbara (2016)
25. Hughes, G.W.: The Recognition of Speech by Machine. Ph.D. Thesis, MIT (1961)
26. Hwang, M.Y., Huang, X.: Subphonetic modeling for speech recognition. In: Human Language Technology (HLT), pp. 174–179 (1992)
27. IATE: Interactive terminology for Europe (2020). https://iate.europa.eu. Accessed 26 July 2020
28. International Phonetic Association: Handbook of the International Phonetic Association, Cambridge (1999)
29. Kamholz, D., Pool, J., Colowick, S.M.: PanLex: building a resource for panlingual lexical translation. In: Proceedings of the Conference on Language Resources and Evaluation (LREC), pp. 3145–3150 (2014)
30. Kneser, R., Ney, H.: Improved backing-off for M-gram language modeling. In: Proc. ICASSP, pp. 181–184 (1995)
31. Köhler, J.: Comparing three methods to create multilingual phone models for vocabulary independent speech recognition tasks. In: Multi-Lingual Interoperability in Speech Technology (1999)
32. Kroeber, P.D.: The Salish Language Family: Reconstructing Syntax. University of Nebraska Press (1999)
33. Kučera, H.: Mechanical phonemic transcription and phoneme frequency count in Czech. Int. J. Slavic Linguist. Poetics **6**, 36–50 (1963)

34. Ladefoged, P.: The revised international phonetic alphabet. Language **66**(3), 550–552 (1990)
35. Lee, F.F.: Automatic grapheme-to-phoneme translation of English. J. Acoust. Soc. Am. **41**(6), 1594 (1969). https://doi.org/10.1121/1.2143635
36. Li, J., Hasegawa-Johnson, M.: Autosegmental neural nets: should phones and tones be synchronous or asynchronous? In: Proceedings Interspeech (2020)
37. Marcantonio, A.: The Uralic language family: facts, myths and statistics. Sapienza Università di Roma (2002)
38. Millward, J.: Eurasian Crossroads: A History of Xinjiang. Columbia University Press (1982)
39. Moran, S., McCloy, D. (eds.): PHOIBLE 2.0. Jena: Max Planck Institute for the Science of Human History (2019)
40. Mortensen, D.R., Dalmia, S., Littell, P.: Epitran: precision G2P for many languages. In: Proceedings of the Conference on Language Resources and Evaluation (LREC), pp. 2710–2714 (2018)
41. Neubig, G., et al.: DyNet: the dynamic neural network toolkit (2017). https://arxiv.org/pdf/1701.03980.pdf. Accessed 14 Sept 2017
42. Novak, J.R., Minematsu, N., Hirose, K.: Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. Natural Lang. Eng. **22**(6), 907–938 (2015)
43. Omar, A.H.: The Malay spelling reform. J. Simplified Spelling Soc. **1989**(2), 9–13 (1989)
44. Peters, B., Dehdari, J., van Genabith, J.: Massively multilingual neural grapheme-to-phoneme conversion. In: EMNLP 2017 Workshop on Building Linguisically Generalizable NLP Systems (2017)
45. Peterson, G.E.: Automatic speech recognition procedures. Lang. Speech **4**(4), 200–219 (1961). https://doi.org/10.1177/002383096100400403
46. Povey, D., et al.: The Kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB
47. Rentzepopoulos, P.A., Kokkinakis, G.K.: Efficient multilingual phoneme-to-grapheme conversion based on HMM. Comput. Linguist. **22**(3), 351–376 (1996)
48. Ritchie, M., Comrie, B. (eds.): The Intercontinental Dictionary Series. Max Planck Institute for Evolutionary Anthropology, Leipzig (2015). http://ids.clld.org. Accessed 26 July 2020
49. Rolston, L., Kirchhoff, K.: Collection of bilingual data for lexicon transfer learning. Technical report, UWEETR-2016-0000, University of Washington Department of Electrical Engineering (2016)
50. Schultz, T.: GlobalPhone: a multilingual speech and text database developed at Karlsruhe University. In: Seventh International Conference on Spoken Language Processing (2002)
51. Schultz, T., Waibel, A.: Multilingual and crosslingual speech recognition. In: Proceedings International Conference Spoken Language Processing (ICSLP), pp. 0577:1–4 (1998)
52. Uzman, M.: Romanisation in Uzbekistan past and present. J. Roy. Asiatic Soc. **20**(1), 49–60 (2010)
53. van Rijnsoever, P.: A multilingual text-to-speech system. In: IPO Annual Progress Report, pp. 34–41. Institute for Perception Research, Eindhoven (1988)
54. Varga, K.: Kaldi ASR: Extending the ASpIRE model (2017). chrisearch.wordpress.com/2017/03/11/speech-recognition-using-kaldi-extending-and-using-the-aspire-model

55. Vasu, S.C.: The Ashtádhyáyí of Páṅini. Translated into English, Sindhu Charan Bose (1897)
56. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. Curran Associates, Inc. (2017). http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
57. Watanabe, S., et al.: ESPnet: end-to-end speech processing toolkit. In: Proceedings Interspeech, pp. 2207–2211 (2018). https://doi.org/10.21437/Interspeech.2018-1456
58. Żelasko, P., Moro-Velázquez, L., Hasegawa-Johnson, M., Scharenborg, O., Dehak, N.: That sounds familiar: an analysis of phonetic representations transfer across languages. In: Proceedings Interspeech (2020)