Massively Multilingual Pronunciation Mining with WikiPron

Jackson L. Lee, Lucas F. E. Ashby[*], M. Elizabeth Garza[*], Yeonju Lee-Sikka[*], Sean Miller[*], Alan Wong[*], Arya D. McCarthy[†], Kyle Gorman[*]

[*]Graduate Center, City University of New York

[†]Center for Language and Speech Processing, Johns Hopkins University

Abstract

We introduce WikiPron, an open-source command-line tool for extracting pronunciation data from Wiktionary, a collaborative multilingual online dictionary. We first describe the design and use of WikiPron. We then discuss the challenges faced scaling this tool to create an automatically-generated database of 1.7 million pronunciations from 165 languages. Finally, we validate the pronunciation database by using it to train and evaluating a collection of generic grapheme-to-phoneme models. The software, pronunciation data, and models are all made available under permissive open-source licenses.

Keywords: speech, pronunciation, grapheme-to-phoneme, g2p

1. Introduction

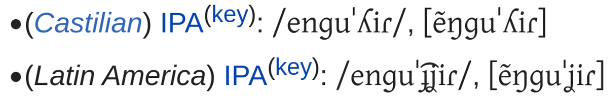
Nearly all speech technologies depend on explicit mappings between the orthographic forms of words and their pronunciations, represented as a sequence of phones. These mappings are constructed using digital pronunciation dictionaries, and for out-of-vocabulary words, grapheme-to-phoneme conversion models trained on such dictionaries. Like many language resources, pronunciation dictionaries are expensive to create and maintain, and free, large, high-quality dictionaries are only available for a small number of languages.

1.1. Prior work

Given the importance of pronunciation modeling to speech technology and the dearth of freely available data, some researchers have exploited crowd-sourced pronunciation data (Ghoshal et al., 2009). One obvious source of data is Wiktionary, a collaborative multilingual online dictionary. Wiktionary has been mined for many natural language resources, including UniMorph, a multilingual database of morphological paradigms (Kirov et al., 2018). Schlippe et al. (2010) extract Wiktionary pronunciation data for English, French, German, and Spanish. They report that this data is both abundant and improves automatic speech recognizer performance. However, they do not release any software or data. Deri and Knight (2016) release a collection of 650,000 word-pronunciation pairs extracted from Wiktionary; once again, they do not release the associated software.

1.2. Contributions

In this paper we introduce WikiPron, an open-source tool for mining pronunciation data from Wiktionary. We then describe a database of 1.7 million word/pronunciation pairs in 165 languages, both living and dead, natural and constructed, that we mined using this tool. Finally, we use this database to perform experiments in grapheme-to-phoneme modeling. WikiPron and the full pronunciation database are hosted at a public open-source repository.[1] Materials for the modeling experiments are hosted at a separate open-source repository.[2] While we target a smaller number of languages than the 531-language data set provided by Deri and Knight (2016)—we omit languages with fewer than 100 word-pronunciation pairs, and do not perform any sort of cross-lingual projection—our database contains more than twice as many word-pronunciation pairs. Furthermore, we release our mining software so that users no longer depend on ossified snapshots of an ever-growing, ever-changing collaborative resource.



- (*Castilian*) IPA(key): /enguˈʎiɾ/, [ẽŋguˈʎiɾ]
- (*Latin America*) IPA(key): /enguˈɟ͡ʝiɾ/, [ẽŋguˈʝiɾ]

Figure 1: Pronunciation of the Spanish word *enguillar* ‘to wolf down’ as it appears on Wiktionary. The entry gives phonemic and phonetic transcriptions for two dialects.

1.3. Wiktionary pronunciation data

At the time of writing, the English edition of Wiktionary has pronunciation entries for over 900 languages.[3] An example of this data is shown in Figure 1. Among them are living, ancient (e.g., Egyptian), constructed (e.g., Esperanto), and even reconstructed (e.g., Proto-Austronesian) languages. Of these, nearly 200 languages have 100 or more entries. Pronunciations are given in the International Phonetic Alphabet (IPA), and many languages provide transcription guidelines for Wiktionary contributors.

2. Using WikiPron

WikiPron is implemented as a Python package hosted by the Python Package Index (PyPI). In a Python 3.6+ environment, WikiPron can be conveniently downloaded and installed by executing the terminal command

```
pip install wikipron
```

To scrape pronunciation data for, say, French (ISO 639 code: `fra`), the terminal command

```
wikipron fra
```

[1] `https://github.com/kylebgorman/wikipron`

[2] `https://github.com/kylebgorman/wikipron-modeling`

[3] `https://en.wiktionary.org/wiki/Category:Terms_with_IPA_pronunciation_by_language`

Word	Pronunciation
accrémentitielle	a k ʁ e m ɑ̃ t i t j ɛ
accrescent	a k ʁ ɛ s ɑ̃
accrétion	a k ʁ e s j ɔ̃
accrétions	a k ʁ e s j ɔ̃

Table 1: Sample French "phonemic" pronunciation data scraped by WikiPron; the pronunciations have been segmented, and stress and syllable boundary markers removed.

initiates the scraping run and prints the data to standard output. Optional command-line arguments used for further customization are discussed in section 3.2 below. The output of WikiPron is a list of UTF-8 encoded word/pronunciation pairs, with each pair on its own line and word and pronunciation separated by a tab character. Sample output is shown in Table 1. This simple output format is intended to be sufficiently generic to be used in a wide variety of circumstances. WikiPron also has a Python API, which allows one to build more sophisticated workflows, such as the massively multilingual mining tool we now discuss.

3. The massively multilingual database

The vast majority of prior work on grapheme-to-phoneme modeling is limited to a handful of high-resource languages for which large pronunciation databases are publicly available. Or, in other cases, such as the recent study of multilingual G2P by van Esch et al. (2016), modeling experiments are conducted using proprietary resources and thus these results are not replicable by the larger research community. Furthermore, researchers interested in multilingual G2P are limited to a single static snapshot (Deri and Knight, 2016) of this unique and dynamic resource for pronunciation data. To remedy this limitation, the WikiPron repository hosts a database of pronunciations from the 165 Wiktionary languages for which at least 100 pronunciations are available. It also contains code used to automatically generate and update this database. This design allows us to quickly produce versioned releases of the database on an annual basis.

3.1. Summary statistics

Table 2 gives the number of pronunciation entries for the 165 languages, dialects, and scripts currently supported. In all, these comprise 1.7 million pronunciations.

3.2. Challenges

We faced a number of challenges in developing WikiPron to support hundreds of Wiktionary languages. Below, we describe some major linguistic and technical challenges, and the solutions pursued by WikiPron.

Phonetic versus phonemic transcription Wiktionary entries (both within and across languages) vary in terms of whether phonetic or phonemic transcription is given. For consistency, we decided it was desirable to separate phonemic and phonetic transcriptions. Fortunately, the distinction is indicated by the use of square brackets (for phonetic transcription) or slashes (for phonemic transcription) as is standard in linguistic literature.[4] Therefore, WikiPron allows users to select either phonemic or phonetic transcriptions via a command-line flag.

Dialect specification Many Wiktionary pronunciations are paired with dialectal specifications, as exemplified by Figure 1. If these specifications were simply ignored, we would obtain a large number of pronunciation variants for each word. Therefore, WikiPron allows users to limit their query to certain dialect specifications via a command-line flag. At the time of writing, there are four languages each split into two separate dialects in the massively multilingual database; the two registers of Norwegian—Bokmål and Nynorsk—are treated as separate languages by Wiktionary.

IPA segmentation For modeling purposes, it is highly desirable to have pronunciations segmented in a way that properly recognizes IPA diacritics, e.g., that keeps combining and modifier diacritics with their host phonetic symbols or preserves the transcription of contour segments indicated using tie bars. For example, consider [k^{h}æt], a phonetic transcription of the English word *cat*. A naïve segmentation separating out each Unicode codepoint would separate [k] and its aspirated release, giving ⟨k, ^{h}, æ, t⟩. WikiPron uses the `segments` library (Moran and Cysouw, 2018) to segment IPA strings. This correctly segments [k^{h}æt] as ⟨k^{h}, æ, t⟩. One known limitation of the segments library is that it does not yet properly segment diacritics that precede the phone they are meant to modify. For example, the Faroese word *kokusnøt* /k^{h}oː^{h}kʊsnøː^{h}t/ 'coconut', which contains two pre-aspirated stops, is segmented as ⟨k^{h}, oː^{h}, k, ʊ, s, n, øː^{h}, t⟩ with the aspiration incorrectly attached to the preceding vowels. Finally, IPA segmentation can also be disabled using a command-line flag.

Suprasegmentals WikiPron also has command-line flags allowing users to optionally remove word stress marks or syllable boundaries. These options are enabled for the massively multilingual database because stress and syllable boundaries are often omitted in G2P modeling tasks.

Special extraction The vast majority of Wiktionary languages use the same underlying HTML structure for their entries, a key feature which enables massively multilingual pronunciation mining. However, some languages require special treatment, and targeting the IPA transcription or the correct orthographic form can be technically challenging. Certain languages, such as Khmer and Thai, require bespoke extraction functions to target their pronunciations, while other languages like Japanese require special extraction functions to target their pronunciations and orthographic forms. Wiktionary entries in Japanese have headwords written in kanji, hiragana, or katakana. Kanji entries also provide their corresponding katakana form and all entries list their correspond-

[4] It is important to note that the distinction between "phonemic" and "phonetic" transcriptions on Wiktionary does not necessarily correspond to the linguistic notions of this distinction. In particular, "phonemic" transcriptions for some languages include predictable allophones; for example, German "phonemic" transcriptions contain both [ç] versus [x], despite the fact that these have long been regarded as allophones of a single phoneme (Bloomfield, 1930). Wiktionary's "phonemic" and "phonetic" transcriptions are more accurately described as "broad" and "narrow", respectively.

Language	# entries	Language	# entries	Language	# entries
Adyghe	4,620	Hindi	8,218	Old Tupi	147
Afrikaans	897	Hungarian	44,670	Oriya	211
Albanian	1,149	Hunsrik	812	Ottoman Turkish	116
Alemannic German	300	Icelandic	9,614	Pashto	1,210
Aleut	104	Ido	5,110	Persian	3,300
Ancient Greek	68,783	Indonesian	1,182	Piedmontese	281
Arabic	5,036	Interlingua	264	Pipil	262
Aramaic	2,330	Irish	6,117	Pitjantjatjara	125
Armenian	13,568	Italian	9,612	Polish	118,947
Assamese	4,384	Japanese (Hiragana)	14,494	Portuguese (Brazil)	9,315
Asturian	130	Japanese (Katakana)	4,549	Portuguese (Portugal)	9,539
Azerbaijani	1,985	Kabardian	824	Punjabi	132
Balinese	172	Khmer	2,950	Romanian	4,300
Bashkir	1,932	Kikuyu	1,010	Russian	388,999
Basque	222	Korean	12,623	Sanskrit	4,577
Belarusian	1,168	Kurdish	1,152	Sardinian	107
Bengali	663	Lao	299	Scots	869
Breton	480	Latin	34,017	Scottish Gaelic	904
Brunei Malay	339	Latvian	1,269	Skolt Sami	77
Bulgarian	34,355	Libyan Arabic	154	Serbo-Croatian (Cyrillic)	22,683
Burmese	3,998	Ligurian	753	Serbo-Croatian (Latin)	23,685
Carrier	175	Limburgish	125	Sicilian	736
Catalan	46,948	Lithuanian	12,603	Slovak	3,742
Cebuano	266	Livonian	353	Slovene	4,360
Chichewa	734	Low German	189	Spanish (Castilian)	47,597
Choctaw	109	Lower Sorbian	1,930	Spanish (Latin America)	38,184
Classical Nahuatl	1,182	Luxembourgish	3,980	Sranan Tongo	153
Classical Syriac	5,924	Macedonian	4,760	Swedish	2,826
Coptic	105	Malay	2,486	Sylheti	224
Cornish	401	Maltese	2,118	Tagalog	1,391
Czech	20,328	Manx	195	Tajik	132
Dalmatian	176	Marshallese	321	Tamil	1,351
Danish	4,119	Mauritian Creole	184	Taos	135
Dongxiang	117	Mecayapan Nahuatl	111	Telugu	441
Dutch	22,175	Mi'kmaq	134	Thai	14,095
Dzongkha	190	Middle Dutch	210	Tibetan	1,569
Egyptian	2,684	Middle English	6,293	Tongan	154
English (UK, R.P.)	52,425	Middle Low German	171	Turkish	2,009
English (US, Gen. Am.)	48,556	Middle Welsh	144	Ukrainian	1,655
Esperanto	14,086	Mongolian	982	Urdu	700
Estonian	283	Navajo	146	Uyghur	207
Faroese	1,639	Neapolitan	238	Vietnamese	10,975
Finnish	38,613	Northern Sami	3,344	Volapük	562
French	53,655	Norwegian (Bokmål)	878	Wauja	146
Galician	4,645	Norwegian (Nynorsk)	1,106	Welsh (North Wales)	4,271
Gamilaraay	444	Norwegian	2,081	Welsh (South Wales)	5,203
Georgian	14,037	Occitan	290	West Frisian	720
German	26,887	Okinawan	152	Western Apache	147
Gothic	623	Old English	6,280	White Hmong	214
Greek	7,842	Old French	334	Xhosa	367
Gulf Arabic	417	Old High German	120	Yakut	134
Hadza	273	Old Irish	1,710	Yiddish	319
Hawaiian	484	Old Norse	160	Zazaki	178
Hebrew	1,161	Old Saxon	178	Zhuang	360
Hijazi Arabic	762	Old Spanish	270	Zulu	907

Table 2: Number of entries per language; if both phonemic and phonetic entries are present for a given language, only the larger of the two is shown. Counting both phonetic and phonemic pronunciations, there are 1,667,526 entries in all.

ing rōmaji elsewhere on the page. For modeling purposes, we extract both hiragana and katakana forms, and then separate hiragana and katakana data as a post-processing step. A similar issue arises in Serbo-Croat. Wiktionary entries for this language include both "Serbian" headwords written in Cyrillic and "Croatian" headwords written in Latin script. We therefore separate the Serbo-Croat data into the two constituent scripts as a post-processing step. A final challenge is posed by Latin. Modern Latin scholarship uses the macron diacritic to indicate long monophthongs, but macrons are not present in Wiktionary headwords. This creates numerous instances of "homographs": for example, the headword *malus* can be pronounced either as *malus* [malus] 'unpleasant' or *mālus* [maːlus] 'apple tree'.[5] This also requires language-specific HTML parsing to extract the proper graphemic form.

4. Experiments

To validate the WikiPron data, we perform a series of grapheme-to-phoneme modeling experiments. We first construct a sample of WikiPron data from fifteen languages. No two languages in this sample are closely related, and and the majority use a non-Latin script. For each language, we remove entries consisting of a single grapheme or a single phone and words with multiple pronunciations. We then randomly partition the data into disjoint training (80%), development (10%), and test (10%) sets.[6]

4.1. Models

We experiment with two types of model, described below.

4.1.1. Pair n-gram model

Our baseline is a form of the *pair n-gram* model (Novak et al., 2016). This approach is closely related to the hidden Markov model approach proposed for G2P by Taylor (2005) but allows for much faster training of higher-order models. Our implementation uses libraries from the OpenGrm collection, including Pynini (Gorman, 2016), Baum-Welch, and NGram (Roark et al., 2012).

Training Let G be the set of graphemes, P the set of phones, and ϵ the empty string. We first construct a *unigram aligner* finite-state transducer

$$C^* = [(G \cup \{\epsilon\}) \times (P \cup \{\epsilon\})]^*$$

where $\times$ is the cross-product operator and * is the Kleene star. The resulting tranducer has the topology of a unigram model of grapheme-to-phone alignment, one which permits any grapheme to align to any one phone, and any phone to align to any one grapheme, and both graphemes and phones can align to nothing, symbolized by ϵ. Next, we use Viterbi training (Brown et al., 1993, 293) to maximize the probability of the training data until convergence. We use 25 random initializations, run in parallel, and select the model which minimizes training data perplexity. Then, we compute the best-probability alignments for the training data using the Viterbi algorithm. We then "encode" the alignments so that each alignment is a finite-state acceptor in which each transition matches a $(G \cup \epsilon, P \cup \epsilon)$ pair. Using these encoded alignments, we compute a higher-order n-gram model over these pairs. This model is smoothed using the Kneser-Ney method (Ney et al., 1994), shrunken to 1 million n-grams using relative entropy pruning (Stolcke, 1998), and encoded as a weighted finite-state acceptor. Finally, we then "decode" the acceptor arcs so that each transition accepts a grapheme or the null ϵ and each transition emits a phone or a null. The resulting weighted finite-state transducer, a weighted relation over $G^* \times P^*$, is our final model. For further details and alternatives, see Novak et al. (2016).

Tuning The development set is used to select the order of the n-gram model; we sweep values in the range 2–9.

Decoding To compute the best path, we compose the grapheme sequence with this weighted transducer, producing a weighted lattice of possible phone sequences. We then compute the highest probability phone sequence through the lattice using the Viterbi algorithm.

4.1.2. Neural sequence model

Neural network sequence-to-sequence models have also been used for G2P. Rao et al. (2015) and Yao and Zweig (2015) report that these models outperform pair n-gram models on CMUDict, a large database of American English pronunciations, and van Esch et al. (2016) apply these models to a large, proprietary 20-language database. Here, we provide a simple proof of concept using the `fairseq` toolkit (Ott et al., 2019).

Training The model consists of a single bidirectional LSTM encoder layer and a single LSTM decoder layer connected by a standard attention mechanism. The two embeddings share parameters, a simple form of regularization. We train using up to fifty epochs of stochastic gradient descent with a fixed learning rate.

Tuning Given that many of the data sets are far smaller than the ones used in prior work on neural network G2P, we limit ourselves to a simple hyperparameter search. We use the development set to perform *early stopping*; that is, we generate a checkpoint each epoch, saving the checkpoint that minimizes development set perplexity. We also use the development set to select the dimensionality of the encoder and decoder, and source and target embeddings, sweeping in lockstep over values in {64, 128, 256, 512, 1024}.

Decoding During decoding we use the early-stopping checkpoints and search using a beam of width five.

4.2. Metrics

Our primary evaluation is word error rate (WER), which is the percentage of words for which the hypothesized transcription sequence does not match the gold transcription. We also report phone error rate (PER), the micro-averaged edit distance between hypotheses and gold transcriptions. This is computed by computing the sum of the edit distances between each hypothesis and gold transcription, and dividing by the summed length of the gold transcriptions. As

[5] Gorman et al. (2019) report that this issue afflicted the Latin data in the CoNLL-SIGMORPHON 2017 shared task on morphological reinflection, which also used data mined from Wiktionary.

[6] A similar evaluation setups is used by Galescu and Allen (2001), Chen (2003), Taylor (2005), Bisani and Ney (2008), and Novak et al. (2016), among others.

is common practice, we multiply both metrics by 100 and express them as percentages.

4.3. Results

Summary statistics and results for the fifteen-language sample are given in Table 3. We observe that the neural sequence model outperforms the pair n-gram model on most—but not all—languages. Error rates are lowest for Hungarian, which has a relatively consistent, "shallow" orthography in the sense of Sproat (2000, 6f.) and one of the larger training sets. The language with the highest error rates overall is English. While this is one of the larger data sets, English orthography is conservative and highly abstract. And, while English is an enthusiastic borrower, it rarely adapts the spelling of words borrowed from other Latin scripts (Kessler and Treiman, 2003). Finally, we note that the 153 unique phonemes in the English WikiPron sample far exceed any reasonable estimate for the number of phonemes in any variety of English, implying the presence of inconsistent—or perhaps overly narrow—transcriptions.

4.4. Error analysis

We also performed a brief manual error analysis for several languages. In Romanian, for example, the largest category of errors involves incorrect prediction of vowel length. Several other errors involve a true ambiguity in the orthography: word-final *i* is read as [i] in some words and as the offglide [ʲ] in others. Finally, a few errors result from incorrect transcriptions in the gold data, and in the case of the neural model, there is at least one "silly" error resisting a proper linguistic characterization: *Transnistria* [transnistria] 'id.' incorrectly transcribed as *[transnist**rist**ria]. Whereas Romanian has a relatively shallow orthography, the French and Korean orthographies are highly abstract. French is written in a Latin alphabet and Korean in the *hangul* syllabary, but in both languages most of the observed errors consist of under- or over-application of phonological rules not indicated in spelling. In French, for example, many errors involve the incorrect deletion or retention of final consonants, such as *plouc* [pluk] 'redneck' incorrectly transcribed as *[plu]. In many other cases final nasalized vowels are also deleted, as in *truculent* 'id.'—transcribed as *[trycyl] rather than [trycylɑ̃]—likely due confusion with the silent third person plural verbal suffix *-ent*. In Korean, such errors often involve the failure to apply phonological rules (e.g., nasalization) across syllable—and thus, grapheme—boundaries. For instance, 익명 [i**ŋ**mjʌ̹ŋ] 'anonymity' is incorrectly transcribed as *[i**k**mjʌ̹ŋ]. We set aside a more systematic error analysis for future work.

5. Conclusion

We describe software for mining pronunciation data from Wiktionary. This software allows us to automatically generate, and regenerate, a database of pronunciations for 165 languages. We hope that these resources will be used to build and evaluate speech technologies, particularly grapheme-to-phoneme conversion engines, for less-resourced and less-studied languages. In future work, we intend to exploit external resources—including Phoible (Moran and McCloy, 2019), a multilingual database of phonemic inventories—to vet this data. Ultimately, we hope that such efforts will improve the quality and consistency of Wiktionary itself. We also will continue to enhance the library to support additional languages, dialects, and scripts, in particular the logographic scripts of East Asia.

6. Acknowledgements

We thank the countless Wiktionary contributors and editors without whom this work would have been impossible.

7. Bibliographical References

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Bloomfield, L. (1930). German ç and x. *Le Maître Phonétique*, 20:27–28.

Brown, P. F., Della Pietra, V. J., Della Pietra, S. A., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–312.

Chen, S. F. (2003). Conditional and joint models for grapheme-to-phoneme conversion. In *EUROSPEECH 2003–INTERSPEECH 2003: 8th European Conference on Speech Communication and Technology*, pages 2033–2036, Geneva.

Deri, A. and Knight, K. (2016). 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)*, pages 399–408, Berlin. Association for Computational Linguistics.

Galescu, L. and Allen, J. F. (2001). Bi-directional conversion between graphemes and phonemes using a joint n-gram model. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, page 131, Perthshire, Scotland.

Ghoshal, A., Jansche, M., Khudanpur, S., Riley, M., and Ulinski, M. (2009). Web-derived pronunciations. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*, pages 4289–4292, Taipei.

Gorman, K., McCarthy, A. D., Cotterell, R., Vylomova, E., Silfverberg, M., and Markowska, M. (2019). Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151, Hong Kong. Association for Computational Linguistics.

Gorman, K. (2016). Pynini: a Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80, Berlin. Association for Computational Linguistics.

Kessler, B. and Treiman, R. (2003). Is English spelling chaotic? Misconceptions concerning its iregularity. *Reading Psychology*, 24:267–289.

Kirov, C., Cotterell, R., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Mielke, S., McCarthy, A., Kübler, S., Yarowsky, D., Eisner, J., and Hulden,

Language	# entries	# graphemes	# phones	Pair n-gram		Neural seq2seq	
				WER (%)	PER (%)	WER (%)	PER (%)
Adyghe	3,538	31	92	30.0	6.9	29.4	7.2
Bulgarian	25,608	30	73	5.6	1.0	5.3	0.9
Burmese	3,129	59	75	28.6	7.3	30.9	8.0
English (UK, R.P.)	31,604	67	153	48.6	13.2	48.2	13.0
French	40,999	50	50	6.0	1.2	6.0	1.2
Georgian	11,215	34	36	23.9	4.0	22.6	3.9
Modern Greek	6,108	51	46	12.8	2.2	14.5	2.4
Hindi	5,678	62	86	12.1	2.4	9.0	2.0
Hungarian	35,460	37	82	2.6	0.5	2.0	0.4
Icelandic	7,296	37	79	16.9	3.0	16.9	3.8
Japanese (Hiragana)	10,968	81	100	11.0	3.1	10.2	3.0
Korean	9,369	1,271	65	39.7	9.4	28.8	6.2
Lithuanian	9,854	32	110	8.4	1.5	8.7	1.4
Romanian	3,256	29	69	12.8	2.7	11.3	2.6
Welsh (South Wales)	1,797	37	47	28.0	6.6	17.3	4.2

Table 3: Results for G2P modeling experiments; WER: word error rate; PER: phone error rate.

M. (2018). UniMorph 2.0: Universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Moran, S. and Cysouw, M. (2018). *The Unicode cookbook for linguists: managing writing systems using orthography profiles*. Language Science Press, Berlin.

Moran, S. and McCloy, D. (2019). *PHOIBLE 2.0.* Max Planck Institute for the Science of Human History, Jena.

Ney, H., Essen, U., and Kneser, R. (1994). On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38.

Novak, J. R., Minematsu, N., and Hirose, K. (2016). Phonetisaurus: exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. *Natural Language Engineering*, 22(6):907–938.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: a fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis. Association for Computational Linguistics.

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

Roark, B., Sproat, R., Allauzen, C., Riley, M., Sorensen, J., and Tai, T. (2012). The OpenGrm open-source finite-state grammar software libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66, Jeju Island, Korea. Association for Computational Linguistics.

Schlippe, T., Ochs, S., and Schultz, T. (2010). Wiktionary as a source for automatic pronunciation extraction. In *INTERSPEECH 2010: 11th Annual Conference of the International Speech Communication Association*, pages 2290–2293, Makuhari, Japan.

Sproat, R. (2000). *A computational theory of writing systems*. Cambridge University Press, Cambridge.

Stolcke, A. (1998). Entropy-based pruning of backoff language models. In *Proceedings of the DARPA Broadcast News And Understanding Workshop*, pages 270–274, Lansdowne, VA.

Taylor, P. (2005). Hidden Markov models for grapheme to phoneme conversion. In *INTERSPEECH 2005–EUROSPEECH 2005: 9th European Conference on Speech Communication and Technology*, pages 1973–1976, Lisbon.

van Esch, D., Chua, M., and Rao, K. (2016). Predicting pronunciations with syllabification and stress with recurrent neural networks. In *INTERSPEECH 2016: 17th Annual Conference of the International Speech Communication Association*, pages 2841–2845, San Francisco.

Yao, K. and Zweig, G. (2015). Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *INTERSPEECH 2015: 16th Annual Conference of the International Speech Communication Association*, pages 3330–3334, Dresden.