Multi-source grapheme-to-phoneme conversion

1 Introduction

Nearly all speech technologies, including automatic speech recognition, text-to-speech synthesis, and virtual assistants like Alexa, Cortana, the Google Assistant, and Siri require mappings between words as they are written—*graphemic* form—and a broad transcription of how they are pronounced—*phonemic* form. However, experience has shown that no digital pronunciation dictionary is ever complete; new words are constantly coined, and in some languages, the set of words speakers can produce is seemingly infinite [1]. Therefore, speech technologies also require a *grapheme-to-phoneme conversion* (or G2P) model, a system which predicts the pronunciation of out-of-vocabulary (OOV) words—i.e., words not found in the pronunciation dictionary—on the basis of their spelling. Very few writing systems exhibit a completely regular mapping between grapheme and phoneme, but G2P prediction is feasible in nearly all languages and scripts [2], though performance varies substantially from language to language and script to script [3–6]. This is true even in English with its notoriously "chaotic" spelling conventions [7].

In a few writing systems, this mapping between grapheme and phoneme is sufficiently consistent that a literate, linguistically-sophisticated speaker can simply enumerate all the necessary rules. However, such rule-based systems are difficult to design and maintain. Therefore, most speech engines use machine learning technologies for G2P. Earlier work uses variants of hidden Markov models [8–11], and more recent work employs sequence-to-sequence neural networks [3–6, 12, 13]. While these neural sequence-to-sequence models outperform earlier methods, they still make non-trivial errors. For instance, it would not be at all surprising for a state-of-the-art English G2P to provide incorrect transcriptions for 10–20% of OOV words. Such errors are likely to propagate, yielding speech recognition errors or unnatural computer-generated speech that negatively impact the usability of virtual assistants. Thus our primary goal is to continue to improve the quality and robustness of G2P systems in general. Our second goal in the proposed study is to develop methods for mapping between pronunciations from different pronunciation dictionaries from the same language. Pronunciation resources for a language may target different dialects and they also may use rather different transcription systems. At present there are no proven methods for mapping between pronunciations in different dialects or transcription systems, making it difficult to exploit multiple resources. Such a method would be useful, for example, for mapping pronunciation data in one dialect to other, less-resourced dialects.

We propose to combine these two goals into a single method, which we term **G&P2P**. In this method, we condition our predicted pronunciations not only the on graphemes, but also on *side-pronunciations*, pronunciations of the same word from other pronunciation dictionaries in different dialects or transcriptions. A pilot study, described below, shows that even a simple method for fusing the grapheme sequence and a side-pronunciation results in a substantial improvement over baseline neural sequence-to-sequence models, and we hypothesize that more sophisticated fusion techniques will result in further improvements, satisfying our first goal. Secondly, G&P2P

CELEX	`# ' k 1 d j @ n`
PronLex	`aa0 r k ey1 d iy0 ih0 n`
WikiPron-UK	`ɑː ' k e ɪ d i ə n`
WikiPron-US	`ɑ ɹ ' k e ɪ d i ə n`

Table 1: Transcriptions of the word *Arcadian* as it appears in the four databases.

CELEX	73,351
PronLex	99,981
WikiPron-UK	52,995
WikiPron-US	49,132

Table 2: Number of pronunciations found in each of the four English lexicons.

can also be used to map between pronunciations in different dialects or transcription systems, satisfying our second goal.

2 Data sources

Our primary experiments will target four well-known English pronunciation lexicons:[1]

- CELEX2 [14]: DISC transcriptions of (Received Pronunciation) British English from the Collins COBUILD Advanced Dictionary
- PronLex [15]: ARPAbet transcriptions of Mainstream American English from the CALLHOME project
- WikiPron-UK [11]: IPA transcriptions of British English extracted from Wiktionary, a free online dictionary
- WikiPron-US [11]: IPA transcriptions of American English from Wiktionary

Sample transcriptions from these four sources are shown in Table 1, and sizes of these sources are shown in Table 2. Initial experiments will focus on predicting held-out words from one of these lexicon, using side-pronunciation data from another lexicon in a different dialect, transcription system, or both. Table 3 provides a matrix describing the proposed single side-pronunciation experiments; the column labeled $|L_1 \cap L_2|$ gives the number of G&P2P examples available for training and evaluation. If time allows, we will conduct additional experiments in French (using WikiPron and Lexique 2 [16]) and Bangla (using WikiPron and the Google lexicon [17]).

[1] We exclude from consideration the well-known CMU Pronouncing Dictionary because our pilot studies raised serious concerns about its consistency and quality.

L_1	L_2	cross-dialect	cross-transcription	$\lvert L_1 \cap L_2 \rvert$	$\lvert L_1 - L_2 \rvert$	$\lvert L_2 - L_1 \rvert$
CELEX	PronLex	✓	✓	57,277	15,333	33,710
CELEX	WikiPron-UK	✗	✓	23,447	49,164	22,213
PronLex	WikiPron-US	✗	✓	24,979	66,008	16,359
WikiPron-UK	WikiPron-US	✓	✗	37,880	7,780	3,458

Table 3: Overlap for single side-pronunciation experiments; $|L_1 \cap L_2|$: # of entries in both L_1 and L_2; $|L_1 - L_2|$: # of entries in L_1 but not in L_2; $|L_2 - L_1|$: # of entries in L_2 but not in L_1.

3 Methods

By exploring different combinations of computational models and fusion techniques we will be able to determine the best practices for incorporating side-pronunciations in G2P and for converting between different standards.

3.1 Models

Following much recent work in G2P [3–6, 11–13] we will use attentive LSTM [18] and transformer [19] neural network sequence-to-sequence models. If time allows, additional experiments will be conducted with pointer-generator networks with either LSTM [20] and transformer [21] encoder and decoder layers. We will use random search to optimize hyperparameters.

3.2 Fusion

We propose several different techniques for combining the grapheme source with various sources of side-pronunciations.

String concatenation with disjoint vocabularies One simple method is to concatenate the source grapheme sequence with with one or more pronunciations from other lexicons, keeping the vocabularies disjoint with "subscripts". For instance, for the word *dog*, and using side-transcriptions from PronLex, the input sequence might be:

```
d_grapheme o_grapheme g_grapheme d_pronlex ao1_pronlex g_pronlex
```

Table 4 gives the results of a pilot study using this method.

String concatenation with control symbols Alternatively, one can concatenate grapheme and side-transcriptions using control symbols indicating the source of each substring [4]. Adapting the previous example, the input sequence for *dog* might be:

```
[grapheme] d o g [pronlex] d ao1 g
```

	LSTM	Transformer
CELEX2	7.88	7.73
CELEX2 (+ PronLex)	**5.56**	5.73
PronLex	9.03	9.32
PronLex (+ CELEX2)	6.49	**6.41**

Table 4: Word error rates for source concatenation pilot experiment. By concatenating PronLex and CELEX2 (respectively) side-transcriptions to the grapheme sequence we obtain a 2.17 absolute (28% relative) and 2.62 absolute (29% relative) reduction in WER for CELEX2 and PronLex.

String concatenation with source embeddings Another alternative is to compute learned embeddings for each of the different sources, and concatenate these to the character embeddings [22]. Let e be the size of the character embeddings and n the length of the concatenated sequence. Then we will learn, for each of the t different sources, a source embedding of size f, and the concatenated embedding matrix fed to the encoder will be of size $(e + f) \times n$.

Hidden state concatenation Yet another alternative is to use separate encoders for each source—grapheme sequence or side-pronunciation—and concatenate their hidden states [23].

Multi-target fusion Finally, we will experiment with a single multi-source, multi-target model using all available side-pronunciations and a special target control symbol to indicate the desired output dialect and transcription system. This method, a variant of one commonly used by multilingual machine translation systems [24], can be combined with any of the above fusion methods.

3.3 Evaluation

We will use word error rate (the percentage of words whose pronunciations are incorrectly predicted) as our primary evaluation metric, with phoneme error rate as a secondary metric.

4 Outcomes

We will first discover and disseminate information about best practices for multi-source G2P. In addition, we will combine the predicted pronunciations produced by the best model into a single automatically-generated multi-dialect, multi-transcription system English pronunciation lexicon; this will released freely to the speech technology community under a Creative Commons license.

5 Approach

Experiments will be conducted by one or more graduate RAs working out of the PI's lab during fall 2023. All requested funds will be used to provide RA wages. Spring 2024 will be used to prepare a paper to be submitted to a conference or workshop sponsored by the Association for Computational Linguistics.

References

[1] J. Hankamer. Morphological parsing and the lexicon. In *Lexical Representation and Process.* Ed. by W. D. Marslen-Wilson. MIT Press, 1989. [2] R. Sproat. *A Computational Theory of Writing Systems.* Cambridge University Press, 2000. [3] D. van Esch et al. Predicting pronunciations with syllabification and stress with recurrent neural networks. In *INTERSPEECH 2016: 17th Annual Conference of the International Speech Communication Association.* 2016. [4] B. Peters et al. Massively multilingual neural grapheme-to-phoneme conversion. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems.* 2017. [5] K. Gorman et al. The SIGMORPHON 2020 shared task in multilingual grapheme-to-phoneme conversion. In *17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology.* 2020. [6] L. F. Ashby et al. Results of the second SIGMORPHON shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology.* 2021. [7] B. Kessler and R. Treiman. Is English spelling chaotic? Misconceptions concerning its iregularity. *Reading Psychology* 24 (2003). [8] P. Taylor. Hidden Markov models for grapheme to phoneme conversion. In *INTERSPEECH 2005–EUROSPEECH 2005: 9th European Conference on Speech Communication and Technology.* 2005. [9] M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication* 50.5 (2008). [10] J. R. Novak et al. Phonetisaurus: exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. *Natural Language Engineering* 22.6 (2016). [11] J. L. Lee et al. Massively multilingual pronunciation mining with WikiPron. In *Proceedings of the 12th Language Resources and Evaluation Conference.* 2020. [12] K. Rao et al. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2015. [13] K. Yao and G. Zweig. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *INTERSPEECH 2015: 16th Annual Conference of the International Speech Communication Association.* 2015. [14] R. H. Baayen et al. CELEX2. LDC96L14. 1996. [15] P. Kingsbury et al. CALLHOME American English Lexicon (PRONLEX). LDC97L20. 1997. [16] B. New et al. Lexique 2: a new French lexical database. *Behavior Research Methods, Instruments, & Computers* 36.3 (2004). [17] A. Gutkin et al. TTS for low resource languages: A Bangla synthesizer. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation.* 2016. [18] M.-T. Luong et al. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* 2015. [19] A. Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems 30.* 2017. [20] A. See et al. Get to the point: summarization with pointer-generator networks. In *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 2017. [21] A. Singer and K. Kann. The NYU-CUBoulder Systems for SIGMORPHON 2020 Task 0 and Task 2. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology.* 2020. [22] S. Wu et al. Applying the transformer to character-level transductions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume.* 2021. [23] B. Zoph and K. Knight. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2016. [24] M. Johnson et al. Google's multilingual neural machine translation system: enabling zero-shot translation. arXiv preprint 1611.04558. 2016.