Low-resource grapheme-to-phoneme mapping with phonetically-conditioned transfer

Michael Hammond
Dept. of Linguistics
U. of Arizona
Tucson, AZ, USA
hammond@u.arizona.edu

Abstract

In this paper we explore a very simple non-neural approach to mapping orthography to phonetic transcription in a low-resource context with transfer data from a related language. We start from a baseline system and focus our efforts on data augmentation. We make three principal moves. First, we start with an HMM-based system (Novak et al., 2012). Second, we augment our basic system by recombining legal substrings in restricted fashion (Ryan and Hulden, 2020). Finally, we limit our transfer data by only using training pairs where the phonetic form shares all bigrams with the target language.

1 Introduction

This paper describes the submission by our team to the 2022 version of the SIGMORPHON grapheme-to-phoneme conversion challenge (McCarthy et al., 2022). Here we describe our efforts to improve grapheme-to-phoneme mapping for low-resource languages in a non-neural context using only data augmentation techniques.

The problem in the low-resource condition was to map from graphemes to phonetic segments with extremely limited data. Specifically, there were 10 languages with 100 training pairs and 100 development pairs. Each pair was a word in its orthographic representation and a phonetic transcription of that word. In addition, for each language, there were up to 1000 additional training pairs in a "related" language. Systems were then tested on 100 additional pairs for each language. The 10 languages are given in Table 1 along with their codes and the number of additional training pairs.

In addition, there was a higher-resource condition where each language had 1000 pairs without transfer data; our focus was the low-resource condition.

2 Initial neural approaches

We started with a fairly generic transformer system inspired by one of the 2020 baseline systems (Gorman et al., 2020). The system we used is adapted from the OpenNMT base (Klein et al., 2017) and is similar to the one used by Hammond (2021) in the 2021 challenge. There is a 512-element embedding layer in both encoder and decoder. There are six layers in both encoder and decoder and each layer also has 512 nodes. The systems are connected by a 8-head attention mechanism (Luong et al., 2015). Training proceeds in 1,000 steps and the decay method is Noam. Optimization is Adam, the batch size is 8, and the dropout rate is 0.1.[1]

Using this system and running 1000 steps, performance on validation data is terrible as seen in Table 2. In column 1 we give the language codes; column 2 has performance for the 100-pair condition; column 3 gives the results for the 1000-pair condition; and column 4 gives the results with transfer data included.

To get a sense of how much data might be required to get decent performance, we ran a similar transformer configuration over subsets of the CMU pronouncing dictionary (Weide, 1998) for 5 epochs and got the performance in Table 3. The point of this chart is that 100 data pairs is orders of magnitude less than what is needed.

3 An HMM-based approach

Based on how poorly our neural approaches performed with such limited data, we went back to classical HMM-based approaches, specifically selecting the *Phonetisaurus* system (Novak et al., 2012).

This system is based on OpenFST and uses weighted finite-state transducers and expectation-

[1]Full configuration files for this and the experiments below are available at https://github.com/hammondm/g2p2022.

Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 245–248

Target language	Code	Transfer language	Code	Number
Bengali	ben	Assamese	asm	1000
Burmese	bur	Shan	shn	841
German	ger	Dutch	dut	1000
Irish	gle	Welsh	wel	1000
Italian	ita	Romanian	rum	1000
Persian	per	Pashto	pus	721
Swedish	swe	Norwegian Nynorsk	nno	1000
Tagalog	tgl	Cebuano	ceb	126
Thai	tha	Eastern Lawa	lwl	253
Ukrainian	ukr	Belarusian	bel	1000

Table 1: Languages, codes, and the number of additional training pairs in the transfer language

Lang	100	1000	all
ben	100.00	93.15	98.63
ger	99.00	93.00	98.00
ita	99.00	92.00	97.00
per	98.21	94.64	100.00
swe	100.00	93.00	92.00
tgl	99.00	92.00	98.00
tha	98.00	78.00	99.00
ukr	100.00	91.00	100.00
gle	100.00	94.00	100.00
bur	100.00	81.00	99.00
mean	99.32	90.17	98.16

Table 2: Validation WER for all languages with encoder-decoder after 1000 steps

Data	WER
1000	100.00
5000	100.00
10000	83.00
20000	69.00
30000	65.00
133802	53.55

Table 3: Validation WER for CMU with a transformer for 5 epochs with different amounts of data

Lang	100/2	100/3	1000/2	1000/3
ben	91.78	91.78	65.75	68.49
ger	88.00	86.00	57.00	61.00
ita	54.00	54.00	33.00	25.00
per	87.50	89.29	76.79	67.86
swe	83.00	82.00	65.00	55.00
tgl	34.00	34.00	19.00	18.00
tha	97.00	95.00	74.00	72.00
ukr	86.00	89.00	57.00	50.00
gle	93.00	95.00	57.00	51.00
bur	98.00	98.00	49.00	48.00
mean	81.22	81.4	55.35	51.63

Table 4: Validation WER for Phonetisaurus without augmentation

maximization to compute the best many-to-many alignment of letters and phonetic symbols. The system offers a number of different options for alignment and decoding, but we ran it in its most "generic" form.

In Table 4 we give WER for 100 pairs and for 1000 pairs. We can use bigrams or trigrams for the alignment and both are given. The point is that, out of the box, the HMM system performs much better than the neural systems. Compare Table 4 with Table 2.

4 Augmentation steps

We tried several kinds of augmentations. The first was the substring approach developed by Ryan and Hulden (2020). In this approach plausible alignments from the beginnings and ends of words are recombined. In the original approach, techniques were used to increase the likelihood that the alignment point occurred at a plausible C-V or V-C juncture. We found that this did not work for all

languages in our test set, presumably due to how limited the data were. We therefore disabled this feature.

The other augmentation we used applied to the transfer data. If one looks at the training pairs, it's apparent that in a number of cases, the languages are not terribly similar.

For example, Irish and Welsh are indeed related and the diligent linguist can easily find cognates. For example, the Welsh word for 'book' is *llyfr* [ɬɨvɨr] and the Irish word is *leabhar* [lʲəurˠ]. The Welsh word for 'man' is *dyn* [dɨːn]; the Irish word for 'person' is *duine* [dˠɪnʲə]. There are also similar grammatical features. For example, both languages use initial consonant mutation as a grammatical mechanism, both have VSO word order, and both have inflected prepositions.

On the other hand, the orthographic conventions of the two languages are extremely divergent, as are the phonetic inventories. For example, Irish has a contrast between palatalized and plain consonants that is completely absent in Welsh. This contrast is reflected in the orthography where adjacent front vowel letters *i* and *e* indicate that a consonant is palatalized. This orthographic practice applies on both sides of a consonant. Thus, if a consonant is intervocalic and palatalized, it will have front vowels on both sides; if it's not palatalized, it will have back vowels. On the other hand, unlike Irish, Welsh strikingly can use *w* and *y* as vowels giving rise to words that seem quite unpronouncable, e.g. *tywydd* [tʰəwɨð] 'weather' or *gŵr* [guːr] 'husband'.

With this in mind, we tried approaches that would limit the transfer data to just those pairs that were most like the target language.

We tried three approaches in this vein. First, we only took pairs where the phonetic segments of the transfer language were in the inventory of the target language. Second, we further restricted the pairs to only those where the phonetic bigrams of the transfer language all occurred in the target language. Finally, we only included pairs where all orthographic characters in the transfer language occurred in the target language.

Different combinations of these options appear in Table 5. The second column gives validation WER for all 100 training pairs plus all transfer data (all). In the third column we have results when only transfer pairs with shared phonetic elements are included (phon). In column 4, we only include transfer pairs where the phonetic and orthographic elements are shared with the target language (phonorth). In column 5, we further restrict that so all phonetic bigrams must be shared (phorth+bg). In column 6, we leave the orthographic relationship unrestricted, but require shared phonetic bigrams (phbg). Finally, in column 7, we have shared bigrams and we add 1900 forms created with shared legal prefixes and suffixes from the target language (phbg+1900).

Looking at Table 5, we see that adding all transfer data diminishes performance. If we restrict the phonetic relationship between the transfer data and the target language, we get some improvement. We also get improvement if we restrict the relationship further with either phonetic bigrams or orthographic overlap, but curiously those two criteria do not help simultaneously. Finally, we see that we get still further improvement with the substring recombination technique.

Performance on the test data of course varies slightly from what we saw with the validation data so we give those results in Table 6 for the full 1000 pairs, the small 100-pair set, and our final system with phonetically-restricted transfer data (using phonetic bigrams) plus substring recombined forms.

5 Conclusion

In conclusion, we've seen several effects. First, a simple encoder-decoder or transformer does not perform well with so few data. Second, an HMM-based approach does better, and does better still when we restrict the kind of transfer data that is used. Specifically, transfer data should be restricted based on how similar it is to the target language. Similarity in terms of phonetics is clearly beneficial, but similarity in terms of orthography seems to help as well. Finally, we saw that the substring recombination technique of Ryan and Hulden (2020) can be added on top of these moves for an additional benefit.

Acknowledgments

Thanks to Sayed Issa and Diane Ohala for useful discussion. All errors are my own.

References

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya McCarthy, Shijie Wu, and Daniel You. 2020. The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings*

Lang	all	phon	phonorth	phorth+bg	phbg	phbg+1900
ben	80.82	79.45	83.56	84.93	84.93	86.30
ger	92.00	85.00	85.00	80.00	80.00	81.00
ita	38.00	38.00	35.00	38.00	38.00	36.00
per	94.64	91.07	91.07	89.29	89.29	89.29
swe	78.00	74.00	71.00	69.00	68.00	71.00
tgl	66.00	37.00	37.00	35.00	35.00	47.00
tha	96.00	93.00	93.00	95.00	95.00	96.00
ukr	96.00	88.00	88.00	95.00	94.00	80.00
gle	98.00	97.00	97.00	96.00	96.00	87.00
bur	97.00	98.00	98.00	98.00	98.00	93.00
mean	83.64	78.05	77.86	78.02	77.82	76.65

Table 5: Validation WER for target + transfer data: a) all data, b) overlapping phonetic segments, c) overlapping and orthographic segments, d) overlapping orthographic segments and phonetic bigrams, e) phonetic bigrams, f) phonetic bigrams and recombined substrings

Lang	1000	100	100+transfer
ben	71.23	91.78	79.45
ger	48.00	90.00	85.00
ita	29.00	50.00	41.00
per	59.65	80.70	82.46
swe	62.00	82.00	81.00
tgl	16.00	24.00	37.00
tha	71.00	95.00	91.00
ukr	53.00	96.00	86.00
gle	56.00	93.00	85.00
bur	46.00	93.00	89.00
mean	51.19	79.55	75.69

Table 6: Test WER for for the full 1000 pairs, the small 100-pair set, and our final system with phonetically-restricted transfer data (using phonetic bigrams) plus substring recombined forms

of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 40–50. Association for Computational Linguistics.

Michael Hammond. 2021. Data augmentation for low-resource grapheme-to-phoneme mapping. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 126–130. Association for Computational Linguistics.

G. Klein, Y. Kim, Y. Y. Deng, J. Senellart, and A.M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. *ArXiv e-prints*. 1701.02810.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Arya D. McCarthy, Jackson L. Lee, Alexandra DeLucia, Winston Wu, Travis M. Bartley, Milind Agarwal, Lucas F.E. Ashby, Luca Del Signore, and Cameron Gibson. 2022. Results of the third SIGMORPHON shared task on cross-lingual and low-resource grapheme-to-phoneme conversion. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Seattle, USA. Association for Computational Linguistics.

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 *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49.

Zach Ryan and Mans Hulden. 2020. Data augmentation for transformer-based G2P. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 184–188. Association for Computational Linguistics.

Robert L. Weide. 1998. The CMU pronouncing dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.