One Model to Pronounce Them All: Multilingual Grapheme-to-Phoneme Conversion With a Transformer Ensemble

Kaili Vesik[1,2], **Muhammad Abdul-Mageed**[1,2,3], **Miikka Silfverberg**[2]
[1] Natural Language Processing Lab
[2] Department of Linguistics
[3] School of Information
The University of British Columbia
[1] {kaili.vesik,muhammad.mageed,miikka.silfverberg}@ubc.ca

Abstract

The task of grapheme-to-phoneme (G2P) conversion is important for both speech recognition and synthesis. Similar to other speech and language processing tasks, in a scenario where only small-sized training data are available, learning G2P models is challenging. We describe a simple approach of exploiting model ensembles, based on multilingual Transformers and self-training, to develop a highly effective G2P solution for 15 languages. Our models are developed as part of our participation in the SIGMORPHON 2020 Shared Task 1 focused at G2P. Our best models achieve 14.99 word error rate (WER) and 3.30 phoneme error rate (PER), a sizeable improvement over the shared task competitive baselines.

1 Introduction

Speech technologies are becoming increasingly pervasive in our lives. The task of *grapheme-to-phoneme (G2P)* conversion is an important component of both speech recognition and synthesis. In G2P conversion, sequences of graphemes (the symbols used to write words) are mapped to corresponding phonemes (pronunciation symbols, e.g., symbols of the International Phonetic Alphabet). Members of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON) have proposed a G2P shared task (SIGMORPHON 2020 Shared Task 1) [1] involving multiple languages. In this paper, we describe our submissions to the shared task. Organizers provide an overview of the task and submitted systems in Gorman et al. (2020) (this volume).

The task was introduced with data from 10 languages, with an additional 5 'surprise' languages released during the task timeline. Our goal was to develop an effective system based on modern deep learning methods as a solution. However, deep learning technologies work best with sufficiently large training data. Hence, a clear challenge we came across is the limited size of the shared task training data for each of the 15 individual languages. To ease this bottleneck, we decided to view the task through a *multilingual machine translation lens* where we build a single model mapping from input to output across all the languages simultaneously. In this, we hypothesized that a multilingual model would allow for shared representations across the various languages that may be more powerful than individual representations of monolingual models. Abundant evidence now exists for approaching machine translation tasks from a multilingual perspective (Johnson et al., 2017a; Dong et al., 2015; Firat et al., 2016), which inspired our choice.

In order to make use of unlabeled data, we also explore a straightforward *self-training approach*. In particular, we employ our trained models to convert sequences of multilingual unlabeled graphemes, taken from Wikipedia data, into multilingual phonemes. We then select sequences of phonemes predicted with our models above a certain confidence threshold to augment the shared task training data, thus re-training our models with larger (gold and silver) training data from scratch. Our models are based on the Transformer architecture which exploits effective self-attention. We show that both our multilingual model and the self-trained variation outperform the results of the competitive baseline monolingual

[1] The shared task webpage is accessible at: https://sigmorphon.github.io/sharedtasks/2020/task1.

Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 146–152

https://doi.org/10.18653/v1/P17

models provided by the task organizers. Ultimately, we demonstrate how our simple modeling choices enable us to provide an effective solution to the problem in spite of the low-resource challenge. Intrinsically, our approach also enjoys the simplicity of a single model rather than 15 different models.

The rest of the paper is organized as follows: Section 2 is a description of the shared task data, evaluation metrics, and baselines. Section 3 introduces both our fully supervised, multilingual models (Section 3.1) and self-trained model (Section 3.2). We present our results in Section 4. We provide an analysis of results and report on an ablation study in Section 5. We overview related work in Section 6, and conclude in Section 7.

2 Task Data, Evaluation, and Baselines

The data provided by the organizers of the shared task are extracted from Wiktionary [2] using the WikiPron library (Lee et al., 2020), and consist of 4,050 gold labeled grapheme-phoneme pairs for each of 15 languages, split into a `training` set (3,600 per language) and a `development` set (450 per language). The blind `test` data comprise 450 sources for each language. The data involves languages in the set *{Adyghe (ady), Armenian (arm), Bulgarian (bul), Dutch (dut), French (fre), Georgian (geo), Modern Greek (gre), Hindi (hin), Hungarian (hun), Icelandic (ice), Japanese hiragana (jpn), Korean (kor), Lithuanian (lit), Romanian (rum), Vietnamese (vie)}*. [3] This set of languages employ a variety of writing systems: *alphabets* (e.g. French), *alphasyllabary* (e.g. Hindi), and *syllabary* (e.g. Japanese hiragana), thus introducing enough diversity and modelling challenge. Table 1 shows sample pairs from training data across 5 languages.

Evaluation. For evaluation, the task organizers use both Word Error Rate (WER) and Phoneme Error Rate (PER). WER is the percentage of words whose predicted transcription does not match the gold transcription; PER is the micro-averaged edit distance between predicted and gold transcriptions. We follow this set-up in evaluating our models on the development data as well, as reported in this paper.

Language	Source	Target (IPA)
Alphabet:		
arm	ահեղ	ɑ h ɛ ʁ
	լիարժեք	l j ɑ ɾ ʒ ɛ kʰ
fre	front	f ʁ ɔ̃
	vêtu	v e t y
Alphasyllabary:		
hin	दिखावा	d ɪ kʰ ɑː ʊ ɑː
	हटना	ɦ ə ʈ n ɑː
kor	개벽	k ɐ̟ b j ʌ̹ kˀ
	오빠	o̞ p̬ a̠
Syllabary:		
jpn	いなり	i n a̠ ɾʲ i
	やせん	j a̠ s ɐ̟̃ ɴ

Table 1: Sample pairs from training data

Baselines. Organizers provide a number of monolingual baselines. The first is a pair n-gram model encoded as a weighted finite-state transducer (FST), implemented using the OpenGRMtoolkit [4]. The second is a bi-LSTM encoder-decoder sequence model implemented using the Fairseq toolkit [5]. The third is a Transformer model also implemented using the Fairseq toolkit. Organizer-provided shared task baselines are shown in Table 2 as WER and PER averages over the 15 languages. We now introduce our models.

	Avg over 15 langs	
Model	WER	PER
FST	22.00	4.92
Bi-LSTM	16.84	3.99
Transformer	17.51	4.30

Table 2: Baseline performance as avg. WER and PER over the 15 languages as provided by task organizers. Baselines exploit monolingual models.

3 Models

As explained, our models are based on Transformers and we offer two primary types of models, depending on how we supervise each. We first introduce *fully supervised* multilin-

[2] https://www.wiktionary.org/.

[3] We use three-character ISO-639-2 abbreviations as not all of the task languages have ISO-639-1 codes.

[4] http://www.opengrm.org/twiki/bin/view/GRM.

[5] https://github.com/pytorch/fairseq.

gual models, then we introduce our *semi-supervised* models (also multilingual). Our semi-supervised models follow a self-training set up. We now explain each of these models.

3.1 Supervised, Multilingual Models

We use a multilingual approach where we train a single model on data from all 15 languages. For this purpose, we prepend a token comprising a language code (e.g. `fre`) to each grapheme sequence source. For our implementation, we use the PyTorch Transformer architecture in the OpenNMT Neural Machine Translation Toolkit (Klein et al., 2017). We set the model hyper-parameters as shown in Table 3, which follows those adopted by Vaswani et al. (2017).

Hyper-Parameter	Value
Number of layers	6
Hidden state size	512
Word embedding size	512
Hidden feed-forward size	2,048
Number of self-attention heads	8
Optimizer	Adam
Dropout probability	0.1
Number of training steps	200K

Table 3: Multilingual Transformer hyper-parameters.

We train the model with 3 different random seeds, and at inference we employ an ensemble consisting of the models from 4 training checkpoints (at 50k, 100k, 150k, and 200k steps) for each of the 3 models generated by the random seeds. We note that OpenNMT averages individual models' prediction distributions, which is how we deploy our ensemble. We use beam search with the OpenNMT default beam width of 5. [6]

3.2 Self-Trained Model

3.2.1 Wikipedia Data Augmentation

One of the models we submitted to the task employs a self-training approach, as a way to augment training data. The additional data is sourced from Wikipedia articles from 12 of the 15 languages (excluding Adyghe, Japanese, and Vietnamese) [7]. We download the Wikipedia dumps from the Wikimedia website [8] and use an off-the-shelf tool [9] for extracting text. Further pre-processing involved removing any remaining XML markup, discarding leading and trailing punctuation and numerals for each word, and ignoring any words with remaining word-internal punctuation or numerals.

Due to time constraints, only one million words from each language were used, and from those only unique entries were submitted to the model for translation and subsequent evaluation as potential candidates for augmenting training data. Table 4 summarizes the size of the Wikipedia data used for each available language. Selection methods and thresholds are discussed in Section 3.2.2.

Language	Translated	Selected
arm	9,947	4,723
bul	9,999	3,197
dut	2,275	860
fre	9,985	2,888
geo	5,038	3,043
gre	9,949	3,419
hin	1,450	727
hun	10,000	3,444
ice	9,839	3,719
kor	4,282	2,681
lit	7,033	3,615
rum	9,785	3,102
Total	89,582	35,418

Table 4: Number of Wikipedia words translated vs. number of words selected for self-training.

3.2.2 Procedure

As explained, self-training data is drawn from the translations of Wikipedia text in 12 languages as predicted by an ensemble model. In order to select pairs to augment the training set, we first calculate the mean per-class softmax value in the development set (which we

[6] We also experimented with beam size 10, but did not obtain improvements on the development set.

[7] We note that there is no Adyghe Wikipedia. Also, the Japenese Wikipedia is not strictly in Hiragana and so we exclude it. By mistake, we did not include Vietnamese either. Clearly, we average results from the self-training models only on the languages for which we augment the data.

[8] https://dumps.wikimedia.org/.

[9] https://github.com/attardi/wikiextractor

find to be at 0.11). [10] Comparatively, the average per-class softmax value for the predicted Wikipedia targets for each language ranges from 0.12 to 0.30. Based on this analysis, we select only those Wikipedia pairs whose predicted targets have a probability greater than 0.2. [11] The selected data are combined with the original (i.e., from official task) training set and the models are re-trained using the same hyper-parameters as the fully-supervised setting.

4 Results

	Multilingual		Self-trained	
Lang	WER	PER	WER	PER
ady	25.56	6.40	25.11	6.47
arm	16.67	3.37	16.89	3.37
bul	28.44	7.30	27.33	7.12
dut	16.00	2.84	15.33	2.84
fre	8.22	1.96	8.44	1.92
geo	24.44	4.92	26.22	5.22
gre	15.11	2.72	16.22	3.00
hin	6.44	1.66	6.89	1.89
hun	2.89	0.54	3.56	0.66
ice	9.56	1.88	10.89	2.23
jpn	7.33	2.18	7.11	2.11
kor	24.22	6.54	26.00	6.50
lit	20.00	4.11	21.11	3.96
rum	12.00	2.94	11.78	2.97
vie	5.56	1.77	5.56	1.91
avg	14.83	3.41	15.23	3.48

Table 5: Development set results for *fully-supervised multilingual* and *self-trained multilingual* models.

Both models demonstrate lower word error rates (WER) and phoneme error rates (PER), averaged across languages, than the baseline monolingual models provided by the task organizers (see Table 2 in Section 2). Error rates per language are shown in Table 5 for the development set and Table 6 for the blind test set (results published by organizers). Table 7

[10] As is known, the softmax function produces a probability distribution over the classes.

[11] There could be different ways to select predicted data for augmentation. For example, one can arbitrarily choose the top $n\%$ most confidently predicted points (with n being a hyper-parameter).

	Multilingual		Self-trained	
Lang	WER	PER	WER	PER
ady	28.44	6.46	29.11	6.46
arm	13.11	2.98	12.89	3.07
bul	27.11	5.91	30.89	6.92
dut	15.78	2.98	16.89	3.07
fre	5.33	1.24	5.78	1.36
geo	26.00	5.25	26.67	5.23
gre	16.67	2.68	15.78	2.60
hin	6.44	1.58	6.67	1.66
hun	4.67	1.05	4.22	0.98
ice	9.56	2.11	9.11	1.83
jpn	6.00	1.44	6.00	1.40
kor	32.22	8.54	32.44	8.86
lit	19.33	3.63	20.00	3.68
rum	9.33	1.96	10.44	2.23
vie	4.89	1.66	4.00	1.28
avg	14.99	3.30	15.39	3.37

Table 6: Blind test set results for *fully-supervised multilingual* and *self-trained multilingual* models.

shows examples of prediction errors, which demonstrate some of the typical minor errors in phenomena such as voicing (e.g. k vs. g), epenthesis and elision (e.g. p ʁ u vs. p ʁ u l), and coarticulation (e.g. b^j vs. b).

On average, the fully-supervised models performed slightly better than the self-trained model. We expected that the self-trained model would see (at least slightly) better performance than the fully supervised; however, due to time constraints, we were not able to augment the training data to such a degree that this hypothesized improvement would be tangible. We leave it as a question for the future whether, and if so to what extent, self-training can improve our models. We now provide an analysis of our findings and report on an ablation study under a number of settings.

5 Analysis & Ablation Study

We suspected that languages with shared writing systems (in our multilingual models) would benefit from the shared representation and hence see better results, posing a challenges to those languages with unique orthography (i.e., orthography not shared by o=any of the other languages considered). However, our results do not support this hypothesis; there did not

Lang	Source	Target	Prediction
arm	զուգարան	z u kʰ ɑ ɾ ɑ n	z u g ɑ ɾ ɑ n
	անխնա	ɑ ŋ χ ə n ɑ	ɑ ŋ χ n ɑ
fre	full	f u l	f y l
	proulx	p ʁ u	p ʁ u l
hin	धन्य	dʱ ə n j ə	dʱ ə n j
	मेहरबानी	m ɛːʱ ɾ b ɑː n iː	m eː ɦ ə ɾ b ɑː n iː
jpn	こたま	k o̞ d a̠ m a̠	k o̞ t a̠ m a̠
	ひぞう	ç i z oː	ç i z o̞ː
rum	ceri	t ʃ e rʲ	t͡ʃ e rʲ
	iubeau	j u bʲ æ u	j u b e̯ a w

Table 7: Sample prediction errors from development data.

appear to be a significant correlation between writing system and results on G2P conversion. For example, a total of 7 of the languages (i.e., dut, fre, hun, ice, lit, rum, vie) use the Roman alphabet, but the WERs for these languages cover a reasonably wide range (from first- to eleventh-best) of the results. It is worth noting, however, that the two languages that use the Cyrillic alphabet (ady, bul) were the two worst-performing languages of the set.

Both prior and subsequent to the task deadline, we performed several ablations in order to assess the effectiveness of our approach. First, we compare results based on single models vs. those based on the ensemble. Table 8 shows the error rates of development set translation by the four training checkpoints used in the ensemble, in this case trained with the default (random) seed. Given that each of these results is poorer than our ensemble results for the multilingual model (WER 14.83 / PER 3.41), it is clear that the ensemble approach is superior. Clearly, the ensemble has the advantage of exploiting multiple predictions for each word. This does result in reduced error rates as compared to individual models.

We also compare our multilingual model's error rates on a given language to those acquired by the respective monolingual models. We note that each of the monolingual models is otherwise initialized with the same parameters as the multilingual model described in Section 3.1. Results for the 15 monolingual models are shown in Table 9. The average WER across all languages is almost twice as big as that of our multilingual model (whether individual or ensemble), and the per-language results are worse across the board as well. The monolingual Georgian WER (25.33) was the only result to approach its multilingual counterpart (24.44). ***Our multilingual approach is clearly a significant improvement over otherwise equivalent monolingually-trained models.***

	Avg over 15 langs	
Checkpoint	WER	PER
50k of 200k steps	16.70	3.93
100k of 200k steps	16.04	3.69
150k of 200k steps	16.25	3.78
200k of 200k steps	15.73	3.65
Ensemble	14.83	3.41

Table 8: Development set results for individual models vs. our ensemble

6 Related Work

Various data-driven models have been successfully applied to G2P conversion. In terms of English conversion, Bisani and Ney (2008) use co-segmentation and joint sequence models for early data-driven G2P. Novak et al. (2016) employ a joint multigram approach to generate weighted finite-state transducers for G2P. Recently, neural sequence-to-sequence models based on CNN and RNN architectures have been proposed for the G2P task delivering superior results compared to earlier non-neural approaches (Chae et al., 2018; Yolchuyeva et al., 2019a). Similar to our approach, Yolchuyeva et al. (2019b) use transformers (Vaswani et al., 2017) to perform English G2P conversion.

	Monolingual	
Lang	WER	PER
ady	33.56	9.31
arm	24.00	5.65
bul	41.33	12.07
dut	30.89	7.73
fre	34.89	12.69
geo	25.33	5.19
gre	24.00	5.13
hin	22.67	6.76
hun	20.89	5.30
ice	30.22	11.12
jpn	11.78	3.73
kor	30.67	9.17
lit	26.00	7.75
rum	20.00	5.52
vie	32.00	13.75
avg	27.22	8.06

Table 9: Development set results for monolingual models.

Multilingual training is a crucial component in our system. Our approach is closely related to multilingual neural machine translation (Johnson et al., 2017b), where a single model is trained to translate between multiple source and target languages. Others have also explored multilingual approaches to G2P. Deri and Knight (2016) use multilingual G2P conversion for the purpose of adapting models from high-resource languages to train weighted finite-state transducers for related low-resource languages. Ni et al. (2018) experiment with multilingual training for deep learning models. They use pretrained character embeddings with LSTM encoder-decoders in order to train multilingual G2P models for Chinese, Japanese, Korean and Thai. In contrast to Ni et al. (2018), we inspect multilingual training in the context of transformer models.

For our second model, whose training data is augmented from Wikipedia, we use a self-taining method. Sun et al. (2019) investigate self-training together with ensemble distillation for English G2P conversion, using transformer models. Their setting resembles ours: A teacher model is first trained using a gold standard labeled G2P training set. The teacher model is then used to label additional grapheme data, producing a silver standard training set. Subsequently, a model ensemble is trained on the combination of the gold and silver data. Sun et al. (2019) train on nearly 200k gold standard examples and 2M silver standard examples and report small improvements. In contrast, we do not observe improvements from self-training. This might be a consequence of the small size of the shared task datasets and our silver standard Wikipedia data.

7 Conclusion

We introduced a multilingual approach to G2P conversion, exploiting Transformers in a fully supervised multilingual setting. Strikingly, our choice to model all languages in a shared, nultilingual space reduces error rates (in WER and PER) by almost one half. We also showed how an ensemble of individually-trained multilingual Transformers, is an improvement over non-ensemble models. We also leveraged multilingual Wikipedia data via a self-training strategy, though due to time constraints we were not able to incorporate enough silver labeled data into training to see the results we had hoped for[12]. Nevertheless, the multilingual models successfully surpassed all organizer-provided baselines on the task and compared favorably to several other submitted models. Our future work includes scaling up our self-training with larger Wikipedia data and choosing fully-trained models (e.g., in our case ones at 200K steps) to include in the ensemble.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Social Sciences Research Council of Canada (SSHRC), and Compute Canada (www.computecanada.ca).

References

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

[12]Training on all available Wikipedia data is in progress at the time of this paper's submission

Moon-jung Chae, Kyubyong Park, Jinhyun Bang, Soobin Suh, Jonghyuk Park, Namju Kim, and Longhun Park. 2018. Convolutional sequence to sequence model with non-sequential greedy decoding for grapheme to phoneme conversion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2486–2490. IEEE.

Aliya Deri and Kevin Knight. 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.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya D. McCarthy, Shijie Wu, and Daniel You. 2020. The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017a. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017b. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. Massively multilingual pronunciation mining with WikiPron. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4216–4221, Marseille.

Jinfu Ni, Yoshinori Shiga, and Hisashi Kawai. 2018. Multilingual grapheme-to-phoneme conversion with global character vectors. In *Interspeech*, pages 2823–2827.

Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose. 2016. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938.

Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2019. Token-level ensemble distillation for grapheme-to-phoneme conversion. *arXiv preprint arXiv:1904.03446*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019a. Grapheme-to-phoneme conversion with convolutional neural networks. *Applied Sciences*, 9(6):1143.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019b. Transformer based grapheme-to-phoneme conversion. *Interspeech 2019*.