Grapheme-to-Phoneme Conversion with a Multilingual Transformer Model

Omnia ElSaadany
Department of Informatics
University of Zurich, Switzerland
omnia.elsaadany@uzh.ch

Benjamin Suter
Department of Informatics
University of Zurich, Switzerland
benjamin.suter@uzh.ch

Abstract

In this paper, we describe our three submissions to the SIGMORPHON 2020 shared task 1 on grapheme-to-phoneme conversion for 15 languages. We experimented with a single multilingual Transformer model. We observed that the multilingual model achieves results on par with our separately trained monolingual models and is even able to avoid a few of the errors made by the monolingual models.

1 Introduction

Grapheme-to-phoneme conversion is the task of predicting the phonemic representation for a given orthographic word, where a phoneme is the smallest unit of sound which can distinguish one word from another. In many languages, some phonemes have different realizations depending on their context, and these variants are called allophones. While the task is about predicting phonemes and not allophones, in fact most datasets (e.g., the datasets for Hungarian, Bulgarian, and Armenian) also contain allophones. However, since the distribution of allophones conditioned on the context is learnable, this is not an issue.

The shared task training data consists of 15 languages which have diverse phonologies, ranging from tonal languages to languages with glottalized consonants, and they are written in eight different writing systems. The data comes from the English version of Wiktionary. Each training set contains 3600 words, and each development and test set contains 450 words. The official metrics for the task are Word Error Rate (WER) and Phoneme Error Rate (PER).

A multilingual approach for grapheme-to-phoneme conversion has been explored by Milde et al. (2017). They propose a sequence-to-sequence multilingual model that benefits from training on additional phonetic representations for the same language (which was not permitted in our shared task).

The Transformer (Vaswani et al. 2017) with its attention mechanism has been applied very successfully to machine translation tasks, and it was also used for grapheme-to-phoneme conversion. Yolchuyeva et al. (2019) suggested using a Transformer-based approach for grapheme-to-phoneme conversion and Yu et al. (2020) proposed a multilingual Transformer model for languages with different writing systems by employing byte-level input representation.

In our submission to the shared task, we explore the performance of a multilingual Transformer model with augmented input representation which can transduce a word from any language present in the training data into its IPA representation.

2 Linguistic Background

2.1 IPA

The phonemic representation in this task uses the International Phonetic Alphabet (IPA). Interestingly, there is an issue with IPA which is lack of "orthography". This might seem surprising given that the IPA aims at representing the pronunciation of words with more rigor than typical orthographies. However, different levels of depth of analysis are possible with IPA, and this makes inconsistent use of symbols among annotators unavoidable. To give an example, Bulgarian exhibits a voiceless coronal plosive /t/~/t̪/. The phoneme is articulated as a dental plosive in Bulgarian. Somewhat randomly, the IPA provides an atomic symbol for the voiceless alveolar plosive (/t/), but only a composed symbol for the voiceless dental plosive (/t̪/). In principle, /t̪/ would be the correct representation for the phoneme in question, but since there is no phonemic contrast between den-

tal and alveolar articulation in Bulgarian, a simple /t/ suffices to represent the voiceless coronal plosive phoneme in Bulgarian. Hence, as is expected, the phoneme is not transcribed consistently in the training data; while /t̪/ is used 1588 times,/t/ is applied 681 times. Similar issues are found frequently for other phonemes, and for other languages.

2.2 Languages

In our monolingual baseline models trained with the Transformer baseline published by the task organizers, the WER (PER) ranged from only 3.78 (0.66) for Hungarian up to 40.00 (16.38) for Korean. Seeing these huge differences in performance, it seemed worth analyzing the difficulties faced by the model for the three languages with the worst WER, viz. Korean (40.00), Bulgarian (30.67), and Georgian (28.44).

2.2.1 Georgian

We were particularly surprised to see Georgian among the seemingly most difficult languages. Georgian has a fully phonemic alphabet; each character represents exactly one phoneme, and each phoneme is represented by exactly one character (Hewitt 1995). Grapheme-to-phoneme conversion (and phoneme-to-grapheme conversion) for Georgian is thus a trivial task and can be done in principle with 100% accuracy using a simple 1-to-1 look-up table.

We actually implemented this look-up table, and this allowed us to identify and quantify the issues in the Georgian dataset. We found that there are three phonemes that are each inconsistently represented by two IPA symbols (and distributed roughly 50/50): i~ɪ; x~χ; ɣ~ʁ. The difference between these symbols is neither phonemic nor allophonic. Rather, it is caused by different annotators using different representation for a given phoneme, in line with the orthographic weakness of the IPA outlined above in Section 2.1.

We reported these data inconsistencies,[1] and we prepared a consistent dataset produced with our look-up table. Together with the organizers, we planned to update the Georgian data directly on Wiktionary and then re-retrieve the training data from there. Unfortunately, bulk uploading to Wiktionary is not trivial, and it was not possible for us to update the data before the task deadline. For the current task, it means that the WER cannot be substantially reduced for Georgian due to these inconsistencies.

2.2.2 Bulgarian

Bulgarian exhibits vowel reduction in unstressed syllables (similar phenomena are found, for instance, in English, German, and Russian), which leads to many allophones for vowels in unstressed positions (Leafgren 2020). These allophones should not be present in a purely phonemic transcription, however they are in the given training set. Furthermore, the pronunciation of a vowel in Bulgarian depends on the position of stress, yet Bulgarian word stress can fall on any syllable and is not completely predictable. We experimented with a self-written tool which predicts the stress position in Bulgarian based on heuristics, however the WER could only be decreased marginally using a stress-annotated training set, which is why we abandoned this approach. Similar issues like the ones discussed above for Georgian are present in the Bulgarian training data, and these were also discussed on GitHub.[2] However, these issues are somewhat more difficult to solve automatically compared to Georgian.

2.2.3 Korean

Korean uses an alphabet that provides a symbol for each consonant and for each vowel, yet it groups symbols into square syllable blocks, which makes it look somewhat close to Chinese and Japanese writing, although it is much simpler. By default, Unicode encodes Korean in syllable blocks and not as single sounds, which results in a character set comprising thousands of characters. Luckily, Unicode also provides code points for the single-sound characters (called Jamo), and syllable characters can easily be decomposed to single-sound characters.[3] We used `hangul-jamo`[4] for this decomposition. To give an example of the decomposition, 가감 /k a̠ g a̠ m/, is decomposed to ㄱ ㅏ ㄱ ㅏ ㅁ . With this approach, we were able to decrease the WER and PER of our Korean baseline Transformer model considerably: the WER was reduced from 40.00 to 21.50, and the PER from 16.38 to 3.86. We use this preprocessing step for Korean for all our submitted models.

[1] https://github.com/sigmorphon/2020/issues/8

[2] https://github.com/sigmorphon/2020/issues/9

[3] http://www.unicode.org/versions/Unicode8.0.0/ch03.pdf

[4] https://github.com/jonghwanhyeon/hangul-jamo

3 Approach

We trained a multilingual model which can transduce a word in any of the 15 source languages into its IPA representation. Multilingual models can be of the types many-to-one, one-to-many, or many-to-many. In our case, there are obviously multiple languages on the source side. On the target side, there is usually exactly one desired phoneme sequence for a given source word. Superficially, we thus have a many-to-one problem. However, many character sequences exist in more than one language. For instance, the character sequence <transformation> without further context can be read as an English word or as a French word, and its pronunciation depends on the choice of language (/tɹæns.fɔɹ.meɪ.ʃən/ vs. /tʁɑ̃s.fɔʁ.ma.sjɔ̃/). This makes it a many-to-many problem for a subset of the data.

The possibility of multiple desired sequences on the target side for a given source word makes it necessary to annotate the source words with the desired language. In our approach, we prefix each source word with its two-letter ISO language code, followed by an underscore, e.g. 'fr_maison', or 'ka_ავტორი'. This is similar to the approach in Johnson et al. (2017).

A side effect of our multilingual approach is that the size of the training data is increased from 3600 to 54000 (15 x 3600) samples. Ideally, a model might profit from this enlarged dataset, and languages can learn from each other. Given the various source-side writing systems and differences in phoneme sets across languages, we expect cross-language learning to be somewhat limited.

The multilingual approach proposed here allows for language-specific preprocessing where needed. In our case, we only used a preprocessing step for Korean, as outlined above in Section 2.2.3.

3.1 Model UZH-1

For our first submission, we used the Transformer baseline[5] provided by the organizers and experimented with different hyperparameters. The Transformer (Vaswani et al. 2017) is implemented in Fairseq (Ott et al. 2019) and uses Adam (Kingma and Ba 2015) for optimization and ReLU as an activation function. It has 4 encoder and decoder layers with 4 attention heads each.

[5] https://github.com/sigmorphon/2020/tree/master/task1/baselines/transformer

In our hyperparameter tuning, we experimented with the following values: embedding dimension {128, 256} and hidden size {512, 1024} for both the encoder and the decoder, batch size {256, 512, 1024}, and dropout probability {0.1, 0.2, 0.3}. The number of epochs is limited to 400.

Our submitted model has the largest possible values for all tuned hyperparameters: embedding dimensions of 256, hidden sizes of 1024, a batch size of 1024, and a dropout probability of 0.3. Due to limitations in available computation power, further tuning with even larger hyperparameter values was not feasible for us.

3.2 Model UZH-2

For our second submission, we added extra language data from 6 languages not addressed in the task, viz. English, Italian, Portuguese, Czech, Danish, and Macedonian. Some of these languages have rather small data sets available on Wiktionary, therefore we added only 2400 training samples per language, and 300 development samples each, which is two thirds of the data for the other languages.

We selected the additional languages based on our intuition regarding whether a language might be useful for one or more of the 15 languages in the task. An additional restriction was the fact that large enough data sets are available mainly for European languages. Of the selected additional languages, some are closely related to another one from the official training set (e.g., Macedonian to Bulgarian, or, to a lesser degree, Danish to Dutch). Others have similar phonologies (e.g., Spanish and Greek, or Czech and Hungarian). In addition, some training sets (e.g., the one for French) contain English loanwords whose irregular pronunciation might be learned from additional English data.

The data was retrieved from Wiktionary using WikiPron (Lee et al. 2020) and sampled randomly. We used the same model architecture and the same hyperparameter search space for this experiment as in UZH-1, and the final model has the same hyperparameter values as UZH-1.

3.3 Model UZH-3

Our third submission is an ensemble model. It uses the predictions of UZH-1 and UZH-2, and for each word it takes the higher probability prediction from the two models.

4 Results

	UZH-1		UZH-2		UZH-3	
	WER	PER	WER	PER	WER	PER
arm	15.56	3.29	15.78	3.52	**14.89**	**3.17**
bul	32.89	6.48	**30.00**	**5.59**	30.22	5.77
fre	7.78	1.88	8.00	1.80	**6.89**	**1.64**
geo	26.44	5.00	28.00	5.11	**26.22**	**4.97**
gre	**18.00**	**2.97**	21.33	3.41	18.89	3.03
hin	6.89	1.58	7.78	2.16	**6.00**	**1.43**
hun	**5.78**	**1.15**	7.11	1.54	6.00	1.18
ice	**11.78**	**2.39**	12.89	2.78	**11.78**	2.46
kor	28.67	4.99	29.11	4.99	**28.44**	**4.88**
lit	27.33	4.69	28.44	4.84	**27.11**	**4.61**
ady	26.00	6.05	28.00	6.35	**25.78**	**5.94**
dut	**17.78**	**3.27**	21.56	3.94	18.67	3.42
jpn	9.33	2.46	**6.00**	1.58	**6.00**	**1.54**
rum	13.33	2.96	13.78	3.11	**12.00**	**2.59**
vie	8.44	2.91	6.67	2.62	**6.22**	**2.46**
macro avg	17.07	3.47	17.63	3.56	**16.34**	**3.27**

Table 1: WER and PER of our 3 models for each language and as macro-average on the official test set.

As can be seen from Table 1, our basic multilingual system (UZH-1) achieved a macro-average WER of 17.07 and a PER of 3.47 on the official test set.

For the multilingual model with additional data from six extra languages (UZH-2), we achieved a macro-average WER of 17.63 and a PER of 3.56. While performance did not increase with this approach, it also did not decrease dramatically, which indicates that it would be possible to have an even larger multilingual model for more than 15 languages without major performance loss.

More interestingly, even though the performance of UZH-2 was slightly worse, the model was able to resolve some of the errors made by UZH-1, while at the same time introducing others. We assume that there is indeed a cross-language interference which can influence the result both positively and negatively. We observed similar behavior on the development set during our experiments, which brought us to the idea of combining the results of both systems to get the best of both. Indeed, our ensemble model (UZH-3), which takes the prediction with the higher probability from UZH-1 and UZH-2, was the best-performing model among our submissions with a macro-average WER of 16.34 and PER of 3.27.

5 Conclusion

While other submissions outperformed our models, our PER for UZH-3 is only 0.51 points higher than that of the winning model (IMS). The difference in WER is slightly higher, with an increase of 2.53 points compared to the winning model. Overall, this shows that a single multilingual model can achieve competitive results even in a setting with highly unrelated languages, by simply prefixing each word with its language code. In future work, we like to explore further how cross-language interference in a multilingual model influences performance both positively and negatively.

References

Brian G. Hewitt. 1995. Georgian: A Structural Reference Grammar. John Benjamins Publishing, Amsterdam, Netherlands.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations.

John Leafgren. 2020. A Concise Bulgarian Grammar. (to be published).

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, page 4216–4221.

Benjamin Milde, Christoph Schmidt, and Joachim Köhler. 2017. Multitask sequence-to-sequence models for grapheme-to-phoneme conversion. In Proceedings of Interspeech 2017, page 2536–2540.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of NAACL-HLT 2019: Demonstrations.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, page 6000–6010.

Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. 2019. Transformer based grapheme-to-phoneme conversion. In Proceedings of Interspeech 2019.

Mingzhi Yu, Hieu Duy Nguyen, Alex Sokolov, Jack Lepird, Kanthashree Mysore Sathyendra, Samridhi Choudhary, Athanasios Mouchtaris, and Siegfried Kunzmann. 2020. Multilingual grapheme-to-phoneme conversion with byte representation. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), page 8234–8238.