Low-Resource G2P and P2G Conversion with Synthetic Training Data

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Arnob Mallik, Grzegorz Kondrak
Department of Computing Science
University of Alberta, Edmonton, Canada
{bmhauer,amirahmad,yixing1,amallik,gkondrak}@ualberta.ca

Abstract

This paper presents the University of Alberta systems and results in the SIGMORPHON 2020 Task 1: Multilingual Grapheme-to-Phoneme Conversion. Following previous SIGMORPHON shared tasks, we define a low-resource setting with 100 training instances. We experiment with three transduction approaches in both standard and low-resource settings, as well as on the related task of phoneme-to-grapheme conversion. We propose a method for synthesizing training data using a combination of diverse models.

1 Introduction

In this system paper, we discuss the participation of the University of Alberta team in the SIGMORPHON 2020 Task 1: Multilingual Grapheme-to-Phoneme Conversion (Gorman et al., 2020). This is a sequence-to-sequence transduction task, in which a word, represented by a sequence of graphemes, must be converted into the sequence of phonemes representing its pronunciation. For example, given the French word *connaissent* the correct output is the phoneme sequence [k ɔ n ɛ s].

Following previous SIGMORPHON shared tasks, in addition to the standard setting with 3600 training examples for each language (which we refer to as the high-resource setting), we define a low-resource setting in which training data is limited to 100 examples. This emulates a plausible scenario of working with a low-resource language for which only a small quantity of reliable phonological data is available. For example, a typical IPA description of the phonological inventory of a single language contains about a hundred phonetic transcriptions of individual words (IPA, 1999). We analyze the relative performance of different systems depending on the size of the training data.

The task of phoneme-to-grapheme (P2G) conversion is the inverse of grapheme-to-phoneme Conversion (G2P), in which the goal is to predict the spelling of a word given its phonetic transcription (Rentzepopoulos and Kokkinakis, 1996). While G2P reflects the difficulty of reading, P2G may indicate the complexity of writing in a given language. Training instances for one of the two tasks can easily be applied to the other one by simply reversing the input and output. We use the shared task datasets to investigate how systems designed for G2P perform on P2G. We also leverage raw text corpora to improve the accuracy on P2G, which indirectly leads to improvements on G2P as well.

We develop a novel method of mitigating resource limitations by synthesizing additional training data using a combination of multiple G2P and P2G models. The underlying intuition is that a P2G model should be the inverse of the corresponding G2P model. Since models trained on a small number of instances tend to have limited accuracy, we attempt to distinguish between the correct and incorrect predictions by ensuring that P2G model output matches the corresponding G2P model input. The precision of this approach is further improved by comparing predictions of different systems. Figure 1 illustrates this idea.

The principal contributions of this paper include a novel G2P data augmentation method that leverages multiple systems and text corpora, as well as a thorough comparison of several G2P and P2G systems in both low-resource and high-resource settings.

2 Prior Work

Our methods build upon the prior work of the University of Alberta teams on string transduction. DirecTL, a feature-based discriminative transducer, was originally designed for the G2P task (Jiampojamarn et al., 2008). In DirecTL+ (Jiampojamarn et al., 2010), the feature set was augmented with

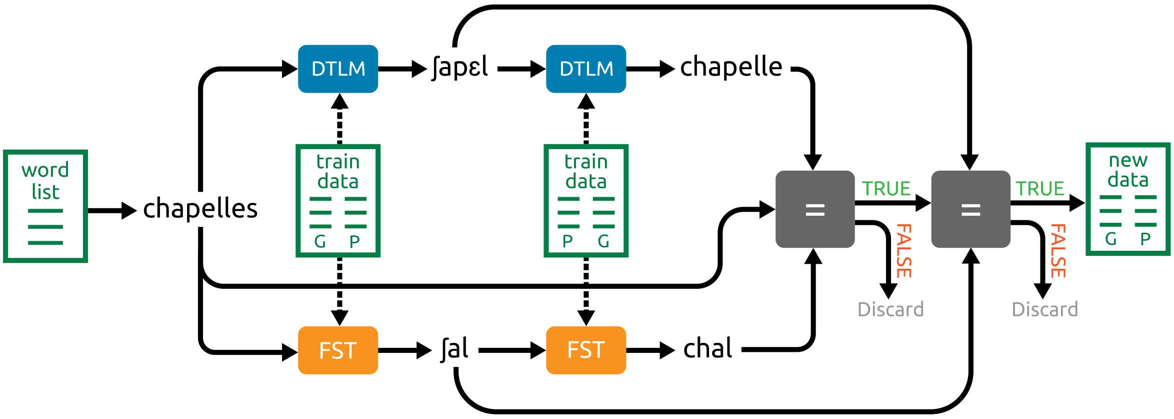

Figure 1: Our approach to synthesizing additional G2P training data.

joint n-grams defined on both source and target substrings. The system was applied to related tasks such as transliteration (Jiampojamarn et al., 2009), morphological inflection (Nicolai et al., 2015), stemming (Nicolai and Kondrak, 2016), and cognate projection (Hauer et al., 2019), proving to be particularly competitive in low-resource settings. DTLM (Nicolai et al., 2018), our principal tool in this work, is a successor of DirecTL+, which incorporates target-side language models and a high-precision alignment. DTLM achieved state-of-the-art results on several tasks in which plain word types constitute the transduction target strings. Finally, our data augmentation approach is inspired by the self-training approach of Hauer et al. (2017).

3 Methods

In this section, we first describe DTLM, a multi-purpose string discriminative transduction system which we apply to both G2P and P2G tasks. We then introduce our approach to synthesizing additional training data from unannotated texts.

3.1 Discriminative String Transduction

The core of DTLM, adapted from DirecTL+, is a dynamic programming algorithm which uses a set of feature templates to transduce multiple characters in a single operation. The feature set includes context features (n-grams on the source side), transition features (target side bigrams), linear-chain features (conjunction of context and transition features), and joint n-gram features (on both source and target).

The transduction quality of DTLM depends on a high precision one-to-many alignment, which is performed with M2M+ aligner (Jiampojamarn et al., 2007) in a two-step process. In the first step, M2M+ induces a one-to-one alignment in which null symbols may be inserted on either side. In the second step, the null links on the source side are removed by merging adjacent target symbols.

The accuracy of DTLM can be enhanced by leveraging target character and word language models. A 4-gram character languages model, which is induced from a set of word types extracted from a text corpus, encourages the prediction of high-probability letter sequences. A unigram word language model (which we also refer to as *word counts*) biases DTLM toward the production of known word-forms, with more common words and prefixes being preferred. Thus, DTLM is able to take advantage of existing multi-lingual text corpora, such as Wikipedia, to improve its accuracy on P2G. Since we have no access to any corpora of phonetic transcriptions, the language model component is not used for G2P.

3.2 Data Augmentation

Inspired by the data hallucination technique for neural model training (Silfverberg et al., 2017; Anastasopoulos and Neubig, 2019), we introduce a method to synthesize additional training instances from unannotated texts. For each language under consideration, we train base transduction models on the available training data, and extract a list of words from a text corpus. A naive self-training approach would be to simply apply a base G2P model to the words in the list to produce new training instances. However, without some mechanism to filter out incorrect predictions, a model trained on the augmented data would learn to replicate many of the errors made by the base model. Instead, we reduce the noise by cross-checking the predictions of the independent base transduction systems applied in both directions.

Figure 1 illustrates the data augmentation process. For each word in the word list, we perform multiple sanity checks before accepting a new training instance. First, both G2P models (in this case, DTLM and FST) must agree on their phoneme predictions. Second, when applied to the common G2P prediction, the corresponding base P2G models must not only agree, but also output the original orthographic word. If both G2P models predict the same phoneme sequence, and both P2G models recover the original grapheme sequence, that grapheme–phoneme pair is added to the synthetic training data. The final augmented model is trained on the combined original and synthetic data.

4 Development

In this section, we describe our development experiments on both G2P and P2G with three different transduction systems and the synthetic training data.

4.1 Datasets

We created low-resource datasets of 100 instances from each standard (high-resource) training set of 3600 instances (Lee et al., 2020). We extracted every 36th instance, starting from the first instance, in a deterministic manner, to ensure replicability. The P2G datasets were created by swapping the grapheme and phoneme strings in the task datasets. The official development sets of 450 instances were used for model tuning only.

4.2 Task Baselines

The task organizers provided implementations of three baseline systems, which are referred to as FST, LSTM, and TRANSFORMER. These are not baselines in the traditional sense of “the simplest possible algorithm” (Manning and Schutze, 2001, page 234), but rather sophisticated systems capable of achieving state-of-the-art results on related tasks. Rather than develop a novel competitive approach, our goal was to combine the unmodified baselines and DTLM to achieve a relative improvement with respect to the individual systems.

As our neural base system, we selected TRANSFORMER, an encoder-decoder architecture with fully-connected layers and self-attention mechanism, which was originally developed for machine translation (Vaswani et al., 2017). Our choice of TRANSFORMER over LSTM was based on initial development experiments.[1] The system is implemented using the Fairseq toolkit (Ott et al., 2019).

Language	DTLM	-LM	-WC	-LM -WC
Dutch	21.6	25.6	25.1	29.8
French	28.2	28.4	48.4	52.2
Greek	33.1	40.9	52.0	59.6

Table 1: WER for variants of DTLM on P2G development sets in the standard (high-resource) setting.

Unlike FST, which only needs to be tuned on the size of n-grams, TRANSFORMER requires extensive tuning which may take several days to complete. We attempted to follow the tuning guidelines as they became available. We kept the hyperparameters as specified in the source code, with the maximum number of training epochs set to 400. The tuning was performed separately for each language in terms of word error rate (WER). We trained the models on two Nvidia Titan RTX GPUs, using Adam optimizer. We varied dropout probability between 0.1, 0.2, and 0.3. and batch size between 256, 512, and 1024 in the high-resource setting, and 64 in the low-resource setting. Due to the underspecification in the guidelines, instead of tuning the number of epochs, we took the model checkpoint of the last epoch.

Unfortunately, we were ultimately unsuccessful in replicating the official results of TRANSFORMER. The implementation used for producing the official results was not available at the system submission time, and used different hyperparameter settings.[2]

4.3 DTLM and P2G

DTLM was our principal system for both G2P and P2G. The models were tuned on the official development sets separately for each task (G2P and P2G), language, and setting (high-resource and low-resource). The context size was varied from 1 to 3 in low-resource, and from 2 to 7 in high-resource settings. We also varied joint n-gram features from 1 to 6, and Markov order from 0 to 2, with and without linear chain features.

For P2G models, we extracted word frequency lists for each language from the first one million

[1] However, the official baseline results, show LSTM as more accurate than TRANSFORMER on most languages. The model results and predictions were not available at the system submission time.

[2] Unlike the earlier implementation that we used, it tuned the number of training epochs without a fixed maximum.

lines of Wikipedia[3], excluding words with frequency less than 10, shorter than 4 characters, or containing non-alphabetic characters. From the word lists, we generated 4-gram character language models using the CMU Toolkit[4]. Target language models are not used for the G2P task because of the lack of phonetic transcription corpora.

Table 1 demonstrates the impact of word counts (WC) and character language models (LM) on P2G accuracy. The results on three challenging languages suggest that most of the DTLM advantage comes from leveraging monolingual text corpora. Furthermore, word counts help more than character LMs. Without those two components, DTLM results on P2G in the standard (high-resource) setting were in the same range as FST and TRANSFORMER.

4.4 Synthetic Training Data

For our data augmentation approach outlined in Section 3.2, we required base G2P and P2G transduction systems. We preferred FST and DTLM over TRANSFORMER, as they performed better on small training datasets in terms of both accuracy and speed. Although data augmentation could also be applied to P2G, we used it exclusively for G2P, which is the primary focus of this shared task.

The starting point for generating the synthetic training data were the word lists extracted from Wikipedia, as described in Section 4.3. We applied the base models to the lists, and filtered out the instances on which the models disagreed or failed to recover the original spelling from their own phonetic predictions. We further limited the number of synthetic training instances to 20,000 per language. This process failed to produce a substantial number of new instances for Vietnamese and Korean, which we attribute to the unusual characteristics of the two scripts.

The data augmentation approach was successful in our development experiments on the standard high-resource datasets, reducing the average WER with respect to base TRANSFORMER from 17.0% to 16.0%, We obtained improvements on 13 out of 15 languages, with the exception of Bulgarian and Korean.[5]

	High Resource			Low Resource		
Language	DTLM	FST	TF	DTLM	FST	TF
Adyghe	18.2	16.7	21.3	53.1	56.0	87.8
Armenian	4.9	5.1	8.0	14.0	27.3	80.7
Bulgarian	6.0	6.4	8.4	20.9	28.7	83.8
Dutch	23.8	27.3	21.1	34.0	66.7	90.4
French	28.7	50.4	51.3	51.6	72.4	94.0
Georgian	1.1	0.7	1.1	4.4	6.4	74.7
Greek	32.9	59.6	56.9	41.3	89.1	97.6
Hindi	3.8	12.0	15.1	18.0	45.8	86.9
Hungarian	4.0	6.9	8.0	14.9	28.7	81.8
Icelandic	13.6	12.0	15.6	28.0	45.6	82.4
Japanese	4.4	9.8	3.6	61.1	59.3	97.8
Korean	39.1	50.0	32.7	96.7	97.3	100
Lithuanian	4.0	3.6	3.3	15.1	25.8	75.1
Romanian	1.8	1.3	2.9	17.8	15.6	57.3
Vietnamese	16.2	18.4	16.2	71.8	85.6	96.9
Average	**13.5**	**18.7**	**17.7**	**36.2**	**50.0**	**85.8**

Table 2: WER on P2G test sets.

5 Test Results

Table 2 shows the P2G results on the test sets. All models are trained on the same training sets, without any synthesized instances. TRANSFORMER (TF) completely fails with only 100 training instances (low resource), but outperforms FST with 3600 training instances (high resource).[6] DTLM is substantially more accurate on average than the other two systems in both settings. Although DTLM benefits from information extracted from freely-available unannotated text corpora, the results of the three systems are directly comparable because they all use the same annotated training material. This further confirms the claim of Nicolai et al. (2018) that DTLM achieves state-of-the-art results on the task of phoneme-to-grapheme conversion.

Table 3 shows the G2P results on the test sets. The DTLM models were trained without any synthetic data or target language models. Although DTLM results are generally lower than on P2G, it outperforms FST in both settings.[7] TRANSFORMER again fails in the low resource setting, In the standard (high resource) setting, DTLM is about 6% worse on average than TRANSFORMER in terms of WER, but 10% better in terms of PER (3.9% vs 4.3% according to the official results). In addition, DTLM is much easier and faster to train.

The TRANSFORMER models trained on the data

[3] *https://dumps.wikimedia.org*

[4] *http://www.speech.cs.cmu.edu/SLM*

[5] Only 36% of the graphemes in the Korean test set are observed in the low-resource train set. The corresponding number in Japanese is 90%.

[6] We note that the P2G accuracy is particularly high on Georgian, which, unlike French, seems to be easier to write than to read.

[7] FST, which is not included in Table 3, obtains 22.0% WER average in the standard setting according to the official results, and 58.1% WER average in the low-resource setting, as our submission with RunID=5.

	High Resource			Low Resource		
Language	DTLM	TF	TF+	DTLM	TF	TF+
RunID	1	2	3	4	6	-
Adyghe	29.8	28.9	28.2	54.4	92.9	58.4
Armenian	16.9	13.1	16.0	36.4	82.9	36.2
Bulgarian	35.8	30.0	36.7	67.6	93.3	66.4
Dutch	19.6	19.3	16.9	58.7	93.6	57.6
French	7.6	6.4	6.4	53.3	94.9	44.9
Georgian	28.2	25.8	27.1	39.6	84.4	42.2
Greek	15.8	17.1	17.3	39.1	88.0	44.0
Hindi	12.2	10.7	8.7	48.2	89.8	43.1
Hungarian	5.3	6.0	5.3	27.6	87.6	22.7
Icelandic	13.1	10.2	11.3	61.6	90.9	62.0
Japanese	8.7	6.7	6.7	57.8	98.0	53.1
Korean	45.3	45.1	45.1	95.1	100	100
Lithuanian	21.8	22.7	24.4	62.7	90.7	64.0
Romanian	11.3	12.7	10.7	30.2	69.3	28.9
Vietnamese	7.8	7.3	8.7	75.3	95.3	87.3
Average	**18.6**	**17.5**	**18.0**	**53.8**	**90.1**	**54.1**

Table 3: WER on G2P test sets.

augmented with synthesized instances (labeled as TF+ in Table 3) achieved consistently higher results in our development experiments in the standard (high resource) setting (Section 4.4). Unfortunately, a corresponding improvement is not seen in the official test results. Possible explanations include the limit of 400 on the number of epochs made by the task organizers, as well as the suboptimal tuning procedure, which might have accidentally resulted in the overfitting of the augmented model. This is also suggested by the fact that the results of our TRANSFORMER models are often better than the official results on the test datasets.

On the other hand, the data augmentation approach is remarkably successful in the low-resource setting, yielding an average WER improvement over 35% with respect to base TRANSFORMER. We interpret these results as a strong proof-of-concept of the validity of our data augmentation approach; when training data is limited, it can dramatically improve the accuracy of neural models, without any change to their architecture.

6 Conclusion

We have presented a novel data augmentation method that combines the strengths of multiple string transduction methods. We have also explored both G2P and P2G tasks in both the standard high-resource setting, and a low-resource setting of our own design. The results demonstrate that the weakness of neural systems in low-resource settings can be mitigated through the application of data augmentation.

Acknowledgements

We thank Garrett Nicolai for the assistance with DTLM. We thank the organizers of the shared task for their effort. In particular, we thank Kyle Gorman for promptly answering our questions during the pandemic lockout.

This research was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, and Alberta Advanced Education.

References

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China. 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 *this volume*.

Bradley Hauer, Amir Ahmad Habibi, Yixing Luan, Rashed Rubby Riyadh, and Grzegorz Kondrak. 2019. Cognate projection for low-resource inflection generation. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 6–11, Florence, Italy. Association for Computational Linguistics.

Bradley Hauer, Garrett Nicolai, and Grzegorz Kondrak. 2017. Bootstrapping unsupervised bilingual lexicon induction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 619–624.

IPA, 1999. *Handbook of the International Phonetic Association*. Cambridge University Press.

Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. 2009. DirecTL: a language independent approach to transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 28–31, Suntec, Singapore. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proceedings of ACL-08: HLT*, pages 905–913.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young

Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47, Uppsala, Sweden. Association for Computational Linguistics.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York. Association for Computational Linguistics.

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.

Christopher D. Manning and Hinrich Schutze. 2001. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 922–931.

Garrett Nicolai and Grzegorz Kondrak. 2016. Leveraging inflection tables for stemming and lemmatization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1138–1147.

Garrett Nicolai, Saeed Najafi, and Grzegorz Kondrak. 2018. String transduction with target language models and insertion handling. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 43–53.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Panagiotis A. Rentzepopoulos and George K. Kokkinakis. 1996. Efficient multilingual phoneme-to-grapheme conversion based on HMM. *Computational Linguistics*, 22(3):351–376.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99, Vancouver. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.