Avengers, Ensemble! Benefits of ensembling in grapheme-to-phoneme prediction

Vagrant Gautam, Wang Yau Li, Zafarullah Mahmood, Fred Mailhot[*], Shreekantha Nadig[†], Riqiang Wang, Nathan Zhang

Dialpad Canada [†]Dialpad India
vgautam,wangyau.li,zafar,fred.mailhot
shree,riqiang.wang,nzhang@dialpad.com

Abstract

We describe three baseline beating systems for the high-resource English-only sub-task of the SIGMORPHON 2021 Shared Task 1: a small ensemble that Dialpad's[1] speech recognition team uses internally, a well-known off-the-shelf model, and a larger ensemble model comprising these and others. We additionally discuss the challenges related to the provided data, along with the processing steps we took.

1 Introduction

The transduction of sequences of *graphemes* to *phones* or *phonemes*,[2] that is from characters used in orthographic representations to characters used to represent minimal units of speech, is a core component of many tasks in speech science & technology. This *grapheme-to-phoneme* conversion (or *g2p*) may be used, e.g., to automate or scale the creation of digital lexicons or pronunciation dictionaries, which are crucial to FST-based approaches to automatic speech recognition (ASR) and synthesis (Mohri et al., 2002).

The SIGMORPHON 2021 Workshop included a Shared Task on g2p conversion, comprising 3 sub-tasks.[3] The low- and medium-resource tasks were multilingual, while the high-resource task was English-only. This paper provides an overview of the three baseline-beating systems submitted by the Dialpad team for the high-resource sub-task, along with discussion of the challenges posed by the data that was provided.

2 Sub-task 1: high-resource, English-only

The organizers provided 41,680 lines of data in total; 33,344 for training, and 4,168 each for development and test. The data consists of word/pronunciation pairs (*word-pron pairs*, henceforth), where words are sequences of graphemes and pronunciations are sequences of characters from the International Phonetic Alphabet (International Phonetic Association, 1999). The data was derived from the English portion of the WikiPron database (Lee et al., 2020), a massively multilingual resource of word-pron pairs extracted from Wiktionary[4] and subject to some manual QA and post-processing.[5]

The baseline model provided was the 2nd place finisher from the 2020 g2p shared task (Gorman et al., 2020). It is an ensembled neural transition model that operates over edit actions and is trained via imitation learning (Makarov and Clematide, 2020).

Evaluation scripts were provided to compute *word error rate* (WER), the percentage of words for which the output sequence does not match the gold label.

Notwithstanding the baseline's strong prior performance and the amount of data available, the task proved to be challenging; the baseline system achieved development and test set WERs of **45.13** and **41.94**, respectively. We discuss possible reasons for this below.

[*] Corresponding author. Contributing authors are listed alphabetically.

[1] https://www.dialpad.com/

[2] We use these terms interchangeably here to refer to graphical representations of minimal speech sounds, remaining agnostic as to their theoretical or ontological status.

[3] https://github.com/sigmorphon/2021-task1

[4] https://en.wiktionary.org/

[5] See https://github.com/sigmorphon/2021-task1 for fuller details on data formatting and processing.

2.1 Data-related challenges

Wiktionary is an open, collaborative, public effort to create a free dictionary in multiple languages. Anyone can create an account and add or amend words, pronunciations, etymological information, etc. As with most user-generated content, this is a noisy method of data creation and annotation.

Even setting aside the theory-laden question of when or whether a given word should be counted as English,[6] the open nature of Wiktionary means that speakers of different variants or dialects of English may submit varying or conflicting pronunciations for sets of words. For example, some transcriptions indicate that the users who input them had the *cot/caught* merger while others do not; in the training data “cot” is transcribed /k ɑ t/ and “caught” is transcribed /k ɔ t/, indicating a split, but “aughts” is transcribed as /ɑ t s/, indicating merger. There is also variation in the narrowness of transcription. For example, some transcriptions include aspiration on stressed-syllable-initial stops while others do not c.f. “kill” /kʰ ɪ l/ and “killer” /k ɪ l ɚ/. Typically the set of English phonemes is taken to be somewhere between 38-45 depending on variant/dialect (McMahon, 2002). In exploring the training data, we found a total of 124 symbols in the training set transcriptions, many of which only appeared in a small set (1–5) of transcriptions. To reduce the effect of this long tail of infrequent symbols, we normalized the training set.

The main source of symbols in the long tail was the variation in the broadness of transcription—vowels were sometimes but not always transcribed with nasalization before a nasal consonant, aspiration on word-initial voiceless stops was inconsistently indicated, phonetic length was occasionally indicated, etc. There were also some cases of erroneous transcription that we uncovered by looking at the lowest frequency phones and the word-pronunciation pairs where they appeared. For instance, the IPA /j/ was transcribed as /y/ twice, the voiced alveolar approximant /ɹ/ was mistranscribed as the trill /r/ over 200 times, and we found a handful of issues where a phone was transcribed with a Unicode symbol not used in the IPA at all.

Most of these were cases where the rare variant was at least two orders of magnitude less frequent than the common variant of the symbol. There was, however, one class of sounds where the variation was less dramatically skewed; the consonants /m/, /n/, and /l/ appeared in unstressed syllables following schwa (/əm/, /ən/, /əl/) roughly one order of magnitude more frequently than their syllabic counterparts (/m̩/, /n̩/, /l̩/), and we opted not to normalize these. If we had normalized the syllabic variants, it would have resulted in more consistent g2p output but it would likely also have penalized our performance on the uncleaned test set.[7] In the end, our training data contained 47 phones (plus end-of-sequence and UNK symbols for some models).

3 Models

We trained and evaluated several models for this task, both publicly available, in-house, and custom developed, along with various ensembling permutations. In the end, we submitted three sets of baseline beating results. The organizers assigned sequential identifiers to multiple submissions (e.g. *Dialpad-N*); we include these in the discussion of our entries below, for ease of subsequent reference.

3.1 The Dialpad model (Dialpad-2)

Dialpad uses a g2p system internally for scalable generation of novel lexicon additions. We were motivated to enter this shared task as a means of assessing potential areas of improvement for our system; in order to do so we needed to assess our own performance as a baseline.

This model is a simple majority-vote ensemble of 3 existing publicly available g2p systems: *Phonetisaurus* (Novak et al., 2012), a WFST-based model, *Sequitur* (Bisani and Ney, 2008), a joint-sequence model trained via EM, and a neural sequence-to-sequence model developed at CMU as part of the CMUSphinx[8]

[6] E.g., the training data included the arguably French word-pronunciation pair: *embonpoint* /ɑ̃ b ɔ̃ p w ɛ̃/

[7] Although the possibility also exists that one or more of our models would have found and exploited contextual cues that weren’t obvious to us by inspection.

[8] https://cmusphinx.github.io

toolkit (see subsection 3.2). As Dialpad uses a proprietary lexicon and phoneset internally, we retrained all three models on the cleaned version of the shared task training data, retaining default hyperparameters and architectures.

In the end, this ensemble achieved a test set WER of **41.72**, narrowly beating the baseline (results are discussed in more depth in Section 4).

3.2 A strong standalone model: CMUSphinx `g2p-seq2seq` (Dialpad-3)

CMUSphinx is a set of open systems and tools for speech science developed at Carnegie Mellon University, including a g2p system.[9] It is a neural sequence-to-sequence model (Sutskever et al., 2014) that is Transformer-based (Vaswani et al., 2017), written in Tensorflow (Abadi et al., 2015). A pre-trained 3-layer model is available for download, but it is trained on a dictionary that uses ARPABET, a substantially different phoneset from the IPA used in this challenge. For this reason we retrained a model from scratch on the cleaned version of the training data.

This model achieved a test set WER of **41.58**, again narrowly beating the baseline. Interestingly, this outperformed the Dialpad model which incorporates it, suggesting that Phonetisaurus and Sequitur add more noise than signal to predicted outputs, to say nothing of increased computational resources and training time. More generally, this points to the CMUSphinx seq2seq model as a simple and strong baseline against which future g2p research should be assessed.

3.3 A large ensemble (Dialpad-1)

In the interest of seeing what results could be achieved via further naive ensembling, our final submission was a large ensemble, comprising two variations on the baseline model, the Dialpad-2 ensemble discussed above, and two additional seq2seq models, one using LSTMs and the other Transformer-based. The latter additionally incorporated a sub-word extraction method designed to bias a model's input-output mapping toward "good" grapheme-phoneme correspondences.

The method of ensembling for this model is word level majority-vote ensembling. We select the most common prediction when there is a majority prediction (i.e. one prediction has more votes than all of the others). If there is a tie, we pick the prediction that was generated by the best standalone model with respect to each model's performance on the development set.

This collection of models achieved a test set WER of **37.43**, a 10.75% relative reduction in WER over the baseline model. As shown in Table 1, although a majority of the component models did not outperform the baseline, there was sufficient agreement across different examples that a simple majority voting scheme was able to leverage the models' varying strengths effectively. We discuss the components and their individual performance below and in Section 4.

3.3.1 Baseline variations

The "foundation" of our ensemble was the default baseline model (Makarov and Clematide, 2018), which we trained using the raw data and default settings in order to reflect the baseline performance published by the organization. We included this in order to individually assess the effect of additional models on overall performance.

In addition to this default base, we added a larger version of the same model, for which we increased the number of encoder and decoder layers from 1 to 3, and increased the hidden dimensions 200 to 400.

3.3.2 biLSTM + attention seq2seq

We conducted experiments with a RNN seq2seq model, comprising a biLSTM encoder, LSTM decoder, and dot-product attention.[10] We conducted several rounds of hyperparameter optimization over layer sizes, optimizer, and learning rate. Although none of these models outperformed the baseline, a small network (16-d embeddings, 128-d LSTM layers) proved to be efficiently trainable (2 CPU-hours) and improved the ensemble results, so it was included.

[9] https://github.com/cmusphinx/g2p-seq2seq

[10] We used the DyNet toolkit (Neubig et al., 2017) for these experiments.

3.3.3 PAS2P: Pronunciation-assisted sub-words to phonemes

Sub-word segmentation is widely used in ASR and neural machine translation tasks, as it reduces the cardinality of the search space over word-based models, and mitigates the issue of OOVs. Use of sub-words for g2p tasks has been explore, e.g. Reddy and Goldsmith (2010) develop an MDL-based approach to extracting sub-word units for the task of g2p. Recently, a pronunciation-assisted sub-word model (PASM) (Xu et al., 2019) was shown to improve the performance of ASR models. We experimented with pronunciation-assisted sub-words to phonemes (PAS2P), leveraging the training data and a reparameterization of the IBM Model 2 aligner (Brown et al., 1993) dubbed *fast_align* (Dyer et al., 2013).[11]

The alignment model is used to find an alignment of sequences of graphemeres to their corresponding phonemes. We follow a similar process as Xu et al. (2019) to find the consistent grapheme-phoneme pairs and refinement of the pairs for the PASM model. We also collect grapheme sequence statistics and marginalize it by summing up the counts of each type of grapheme sequence over all possible types of phoneme sequences. These counts are the weights of each sub-word sequence.

Given a word and the weights for each sub-word, the segmentation process is a search problem over all possible sub-word segmentation of that word. We solve this search problem by building weighted FSTs[12] of a given word and the sub-word vocabulary, and finding the best path through this lattice. For example, the word "thoughtfulness" would be segmented by PASM as "th_ough_t_f_u_l_n_e_ss", and this would be used as the input in the PAS2P model rather than the full sequence of individual graphemes.

Finally, the PAS2P transducer is a Transformer-based sequence-to-sequence model trained using the ESPnet end-to-end speech processing toolkit (Watanabe et al., 2018), with pronunciation-assisted sub-words as inputs and phones as outputs. The model has 6 layers of encoder and decoder with 2048 units, and 4 attention heads with 256 units. We use dropout with a probability of 0.1 and label smoothing with a weight of 0.1 to regularize the model. This model achieved WERs of **44.84** and **43.40** on the development and test sets, respectively.

4 Results

Our main results are shown in Table 1, where we show both dev and test set WER for each individual model in addition to the submitted ensembles. In particular, we can see that many of the ensemble components do not beat the baseline WER, but nonetheless serve to improve the ensembled models.

Model	dev	test
Dialpad-3	**43.30**	**41.58**
PAS2P	44.84	43.40
Baseline (large)	44.99	41.65
Baseline (organizer)	45.13	41.94
Phonetisaurus	45.44	43.88
Baseline (raw data)	45.92	41.70
Sequitur	46.69	43.86
biLSTM seq2seq	47.89	44.05
Dialpad-2	**43.83**	**41.72**
Dialpad-1	**40.12**	**37.43**

Table 1: Results for components of ensembles, and submitted models/ensembles (bolded).

5 Additional experiments

We experimented with different ensembles and found that incorporating models with different architectures generally improves overall performance. In the standalone results, only the top three models beat the baseline WER, but adding additional models with higher WER than the baseline continues to reduce overall WER. Table 2 shows the effect of this progressive ensembling, from our top-3 models to our top-7 (i.e. the ensemble for the **Dialpad-1** model).

5.1 Edit distance-based voting

In addition to varying our ensemble sizes and components, we investigated a different ensemble voting scheme, in which ties are broken using edit distance when there is no 1-best majority option. That is, in the event of

[11] https://github.com/clab/fast_align

[12] We use Pynini (Gorman, 2016) for this.

Model	dev	test
Ensemble-top3	41.10	39.71
Ensemble-top4	40.74	38.89
Ensemble-top5	40.50	38.12
Ensemble-top6	40.31	37.69
Ensemble-top7 (Dialpad-1)	40.12	37.43

Table 2: Progressive ensembling results, with top-performing components

a tie, instead of selecting the prediction made by the best standalone model (our usual tie-breaking method), we select the prediction that minimizes edit distance to all other predictions that have the same number of votes. The idea of this method is to maximize subword level agreement. Although this method did not show clear improvements on the development set, we found after submission that it narrowly but consistently outperformed the top-N ensembles on the test set (see Table 3).

Model	dev	test
ED-Dialpad-3	43.76	41.70
ED-top3	41.24	39.40
ED-top4	40.62	38.48
ED-top5	40.50	37.69
ED-top6	40.28	37.50
ED-top7	40.21	37.31

Table 3: Results for ensembling with edit-distance tie-breaking

6 Error analysis

We conducted some basic analyses of the **Dialpad-1** submission's patterns of errors, to better understand its performance and identify potential areas of improvement.[13]

6.1 Oracle WER

We began by calculating the *oracle WER*, i.e. the theoretical best WER that the ensemble could have achieved if it had selected the correct/gold prediction every time it was present in the pool of component model predictions for a given input. The Dialpad-1 system's oracle WERs on the dev and test sets were **25.12** and **23.27**, respectively (c.f. 40.12 and 37.43 actual).

These represent massive performance improvements (approx. 15% absolute, or 37% relative, WER reduction), and suggest refinement of our output selection/voting method (perhaps via some kind of confidence weighting) could lead to much-improved results.

6.2 Data-related errors

We also investigated outputs for which none of our component models predicted the correct pronunciation, in hopes of finding some patterns of interest.

Many of the training data-related issues raised in section 2.1 appeared in the dev and test labels as well. In some cases this led to high cross-component agreement, even on incorrect predictions. Our hope that subtle contextual cues might reveal patterns in the distribution of syllabic versus schwa-following liquids and nasals was not borne out, e.g. our ensemble was led astray on words like "warble", which had a labelled pronunciation of /w ɔ ɹ b l̩/, while all 7 of our models predicted /w ɔ ɹ b ə l/, a functionally non-distinct pronunciation. In addition, the previously mentioned issue of /ɹ/ being mistranscribed as /r/ affected our performance, e.g. with the word "unilateral", whose labelled pronunciation was /j u n ɪ l æ t ə r ə l/, instead of /j u n ɪ l æ t ə ɹ ə l/, which was again the pronunciation predicted by all 7 models. Finally, narrowness of transcription was also an issue that affected our performance on the dev and test sets, e.g., for words like "cloudy" /k ɫ a ʊ d i/ and "cry" /k ɹ a ɪ̯/, for which we predicted /k l a ʊ d i/ and /k ɹ a ɪ/, respectively. In the end, it seems that noisiness in the data was a major source of errors for our submissions.[14]

Aside from issues arising due to label noise, our systems also made some genuine errors that are typical of g2p models, mostly related to data distribution or sparsity. For example, our component models overwhelmingly predicted that "irreparate" (/ɪ ɹ ɛ p ə ɹ ə t/) should rhyme instead with "rate" (this "-ate-" /e ɪ t/ correspondence was overwhelmingly present in the training data), that "backache" (/b æ k e ɪ k/) must contain the affricate /t͡ʃ/, that

[13] We are grateful to an anonymous reviewer for suggesting that this would strengthen the paper.

[14] We nonetheless acknowledge the magnitude and challenge of the task of cleaning/normalizing a large quantity of user-generated data, and thank the organizers for the work that they did in this area.

“acres” (e ɪ k ɚ z/) rhymes with “degrees”, and that “beret” has a /t/ sound in it. In each of these cases, there was either not enough samples in the training set to reliably learn the relevant grapheme-phoneme correspondence, or else a conflicting (but correct) correspondence was over-represented in the training data.

7 Conclusion

We presented and discussed three g2p systems submitted for the SIGMORPHON2021 English-only shared sub-task. In addition to finding a strong off-the-shelf contender, we show that naive ensembling remains a strong strategy in supervised learning, of which g2p is a sub-domain, and that simple majority-voting schemes in classification can often leverage the respective strengths of sub-optimal component models, especially when diverse architectures are combined. Additionally, we provided more evidence for the usefulness of linguistically-informed subword modeling as an input transformation on speech-related tasks.

We also discussed additional experiments whose results were not submitted, indicating the benefit of exploring top-N model vs ensemble trade-offs, and demonstrating the potential benefit of an edit-distance based tie-breaking method for ensemble voting.

Future work includes further search for the optimal trade-off between ensemble size and performance, as well as additional exploration of the edit-distance voting scheme, and more sophisticated ensembling/voting methods, e.g. majority voting at the phone level on aligned outputs.

Acknowledgments

We are grateful to Dialpad Inc. for providing the resources, both temporal and computational, to work on this project.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Kyle Gorman. 2016. Pynini: A python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80, Berlin, Germany. Association for Computational Linguistics.

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

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, U.K.

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

Peter Makarov and Simon Clematide. 2018. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882, Brussels, Belgium. Association for Computational Linguistics.

Peter Makarov and Simon Clematide. 2020. CLUZH at 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*, pages 171–176, Online. Association for Computational Linguistics.

April McMahon. 2002. *An Introduction to English Phonology*. Edinburgh University Press, Edinburgh, U.K.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit.

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, Donostia–San Sebastián. Association for Computational Linguistics.

Sravana Reddy and John Goldsmith. 2010. An MDL-based approach to extracting subword units for grapheme-to-phoneme conversion. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 713–716, Los Angeles, California. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. Espnet: End-to-end speech processing toolkit. In *Proc. Interspeech 2018*, pages 2207–2211.

Hainan Xu, Shuoyang Ding, and Shinji Watanabe. 2019. Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7110–7114.