Multilingual grapheme-to-phoneme conversion using a neural network and phonetic features

Overview of the state-of-the-art

by Steve Moran, comments by Tanja Samardzic

The current state-of-the-art with regard to multilingual grapheme-to-phoneme (g2p) conversion is described here in the SIGMORPHON 2020 shared task:

- https://www.aclweb.org/anthology/2020.sigmorphon-1.2/ (https://www.aclweb.org/anthology/2020.sigmorphon-1.2/)

TS: new task published: https://github.com/sigmorphon/2021-task1 (https://github.com/sigmorphon/2021-task1)

The task was led by Kyle Gorman:

- https://research.google/people/KyleGorman/ (https://research.google/people/KyleGorman/)

who summarizes the results here:

- http://www.wellformedness.com/blog/ (http://www.wellformedness.com/blog/)

Taken straight from his blog, highlights of the results include:

- Unsurprisingly, the best systems all used some form of ensembling.
- Many of the best teams performed self-training and/or data augmentation experiments, but most of these experiments were performance-negative except in simulated low-resource conditions. Maybe we'll do a low-resource challenge in a future year.
- LSTMs and transformers are roughly neck-and-neck; one strong submission used a variant of hard monotonic attention.
- Many of the best teams used some kind of pre-processing romanization strategy for Korean, the language with the worst baseline accuracy. We speculate why this helps in the task paper.
- There were some concerns about data quality for three languages (Bulgarian, Georgian, and Lithuanian). We know how to fix them and will do so this summer, if time allows. We may also "re-issue"”" the challenge data with these fixes.

The g2p task is described in detail here, including the data, its format, baselines, and the official results:

- https://sigmorphon.github.io/sharedtasks/2020/task1/ (https://sigmorphon.github.io/sharedtasks/2020/task1/)

A view important points summarized from the task. The data include data for 15 languages (note the different writing systems):

- Adyghe (ady)
- Armenian (arm)
- Bulgarian (bul)
- Dutch (dut)
- French (fre)
- Georgian (geo)
- Hindi (hin)
- Hungarian (hun)
- Icelandic (ice)
- Japanese hiragana (jpn)
- Korean (kor)
- Lithuanian (lit)
- Modern Greek (gre)
- Romanian (rum)
- Vietnamese (vie)

The data include:

- 3600 training data examples and
- 450 development and test data examples for each language

The data format is in NFC Unicode codepoints in UTF-8-encoded tab-separated values files, e.g. from Romanian:

- antonim a n t o n i m
- ploaie p l^{w} a j e
- pornește p o r n e ʃ t e

The data are available here:

- https://github.com/sigmorphon/2020/tree/master/task1/data (https://github.com/sigmorphon/2020/tree/master/task1/data)

and were generated using Wikipron:

- https://github.com/kylebgorman/wikipron (https://github.com/kylebgorman/wikipron)

as described in this paper:

https://www.aclweb.org/anthology/2020.lrec-1.521/ (https://www.aclweb.org/anthology/2020.lrec-1.521/)

and they were segmented with my `segments` library:

- https://github.com/cldf/segments (https://github.com/cldf/segments)

as described in my book on Unicode:

- https://langsci-press.org/catalog/book/176 (https://langsci-press.org/catalog/book/176)

and converted into what Gorman et al call a "rough" IPA:

- https://en.wikipedia.org/wiki/International_Phonetic_Alphabet (https://en.wikipedia.org/wiki/International_Phonetic_Alphabet)

The taks baselines include:

- a pair n-gram model (Novak et al. 2016) implemented using the OpenGrm toolkit (Roark et al. 2012, Gorman 2016), and
- a bidirectional LSTM encoder-decoder sequence model implemented using the Fairseq toolkit (Ott et al. 2019).

Note there are also some restrictions on the challenge, e.g.:

- Participants are not permitted to use any form of pronunciation data derived from Wiktionary, except for the provided training and test data; they are also not permitted to use external pronunciation dictionaries for any of the targeted languages.

However, participants are permitted to use external sources including:

- open-source databases of phoneme inventories and features such as Phoible (Moran & McCloy 2019),
- open-source pronunciation data for languages not targeted in this challenge, and
- open-source morphological analyzers and lexicons such as UDLexicons (Sagot 2018).

And in fact, Gorman et al 2020 note:

`As mentioned above, top submissions make use of techniques such as preprocessing, data augmentation, ensembling, multi-task learning (e.g., phoneme-to-grapheme conversion), and self-training. These techniques are commonly used in shared tasks and are essentially task-agnostic.`

And that:

> However, we were surprised that few teams made use of task-specific resources such as the Phoible (Moran and McCloy 2019) phonemic inventories and feature specifications or rule-based G2P systems like Epitran (Mortensen et al. 2018). Nor do any of the submissions make use of morphological analyzers or lexicons, which were found to be helpful in earlier work (e.g., Coker et al. 1990, Demberg et al. 2007). We speculate that such resources might further improve performance. Finally we note that submissions make use of unsupervised tokenization techniques such as byte-pair encoding (Schuster and Nakajima 2012).

I think this second point is worth exploring in more detail, i.e. using task-specific resources such as PHOIBLE and their feature specification within a neural network framework.

An approach to g2p with neural networks and phonetic feature specifications

The general idea would be to develop a neural network pipeline for phonetic feature specifications along the lines a character level neural machine translation, e.g.

- Fully Character-Level Neural Machine Translation without Explicit Segmentation (Lee_etal2017.pdf)

to test whether the model increases accuracy in the g2p task.

But whereas the character embeddings in character MT are characters, e.g.:

- "T h e i n p u t . . ."

we would instead go "below" the character level and to the level of phonetic features, as described in the PHOIBLE database of phonological inventories:

- https://phoible.org/ (https://phoible.org/)

For example, above we give an example of the data's format for the g2p task for Romanian, including the words and their segmented (and IPA tokenized) forms (in tab-delimited format):

- antonim a n t o n i m
- ploaie p l^{w} a j e
- porneşte p o r n e ʃ t e

In PHOIBLE, the phonological inventories (phonemes) of Romanian are described in two sources:

- https://phoible.org/inventories/view/527 (https://phoible.org/inventories/view/527)
- https://phoible.org/inventories/view/2443 (https://phoible.org/inventories/view/2443)

Each segment in phoible is associated with a set of 37 phonetic/phonological features. Most features are binary (present or not present) and some have "0", which standard for not applicable.

For example, see the report I create here about the g2p task's language coverage in PHOIBLE (long story short, we have phonological inventories for all languages currently in the task):

- https://github.com/uzling/100LC/blob/master/Reports/graphemes/g2p_exploration.md (https://github.com/uzling/100LC/blob/master/Reports/graphemes/g2p_exploration.md)

I show here how we extract those phonemes and their feature vectors as potential input into an NN model.

For example, for the Romanian example, we can extract the phonetic/phonological features for this word:

- antonim (a n t o n i m)

This is a subset for illustrative purposes:

Phoneme consonantal sonorant continuant delayedRelease nasal labial round

m + + - 0 + + -

n + + - 0 + - 0

t + - - - - - 0

a - + + 0 - - 0

i - + + 0 - - 0

o - + + 0

These features can also expand into natural classes if we want to add even more data, e.g. vowels are [+syllabic, -consonantal], fricatives are [-sonorant, +continuant], etc.

All in all, we can potentially use these data as input to a "character" level neural network model for g2p conversion.

TS: Since these are features, probably the best way to encode them is to concatenate them to the character embeddings, but we can explore other options too.

References

I've added some relevant and potentially relevant papers to the the OpenBIS, including:

- What Kind of Language Is Hard to Language-Model? (Mielke_etal2019.pdf)
- Morphological Inflection Generation with Hard Monotonic Attention (AharoniGoldberg2004.pdf)
- The SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion (Gorman_etal2020.pdf)
- Massively Multilingual Pronunciation Mining with WikiPron (Lee_etal2020.pdf)
- Orthographic Codes and the Neighborhood Effect: Lessons from Information Theory (Tulkens_etal2020.pdf)
- Applying the Transformer to Character-level Transduction (Wu_etal2020.pdf)
- ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS (Lan_etal2020.pdf)
- FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling (Ott_etal2019.pdf)
- From Phonology to Syntax: Unsupervised Linguistic Typology at Different Levels with Language Embeddings (BjervaAugenstein2018.pdf)
- Google Crowdsourced Speech Corpora and Related Open-Source Resources for Low-Resource Languages and Dialects: An Overview (Butryna_etal2019.pdf)
- Hidden Markov Models for Grapheme to Phoneme Conversion (Taylor2005.pdf)