phonet Documentation

Release 0.2

Camilo Vasquez

Jan 15, 2020

CONTENTS

This toolkit compute posteriors probabilities of phonological classes from audio files for several groups of phonemes according to the mode and manner of articulation.

If you are not sure about what phonological classes are, have a look at this Phonological classes tutorial

The code for this project is available at https://github.com/jcvasquezc/phonet .

The list of the phonological classes available and the phonemes that are activated for each phonological class are observed in the following Table

Phonological class	Phonemes
vocalic	/a/, /e/, /i/, /o/, /u/
consonantal	/b/, /tS/, /d/, /f/, /g/, /x/, /k/, /l/, //, /m/, /n/, /p/, //, /r/, /s/, /t/
back	/a/, /o/, /u/
anterior	/e/, /i/
open	/a/, /e/, /o/
close	/i/, /u/
nasal	/m/, /n/
stop	/p/, /b/, /t/, /k/, /g/, /tS/, /d/
continuant	/f/, /b/, /tS/, /d/, /s/, /g/, //, /x/
lateral	/l/
flap	//
trill	/r/
voiced	/a/, /e/, /i/, /o/, /u/, /b/, /d/, /l/, /m/, /n/, /r/, /g/, //
strident	/f/, /s/, /tS/
labial	/m/, /p/, /b/, /f/
dental	/t/, /d/
velar	/k/, /g/, /x/
pause	/sil/

NEED HELP?

If you have trouble with Phonet, please write to Camilo Vasquez at: juan.vasquez@fau.de

REFERENCES

If you use Phonet for research purposes, please cite the following paper:

Vásquez-Correa, J. C., Klumpp, P., Orozco-Arroyave, J. R., & Nöth, E. (2019). Phonet: a Tool Based on Gated Recurrent Neural Networks to Extract Phonological Posteriors from Speech. Proc. Interspeech 2019, 549-553.

Download paper

Supported features:

- `phonet.model()` - This is the architecture used for the estimation of the phonological classes using a multi-task learning strategy. It consists of a 2 Bidirectional GRU layers, followed by a time-distributed dense layer

- `phonet.get_phon_wav()` - Estimate the phonological classes using the BGRU models for an audio file (.wav)

- `phonet.get_phon_path()` - Estimate the phonological classes using the BGRU models for all the (.wav) audio files included inside a directory.

- `phonet.get_posteriorgram()` - Estimate the posteriorgram for an audio file (.wav) sampled at 16kHz.

- `phonet.get_PLLR()` - Estimate the phonological log-likelihood ratio (PLLR) features for an audio file (.wav) sampled at 16kHz.

INSTALLATION

From the source file:

```
git clone https://github.com/jcvasquezc/phonet
cd phonet
python setup.py install
```

METHODS

Compute posteriors probabilities of phonological classes from audio files for several groups of phonemes according to the mode and manner of articulation. @author: J. C. Vasquez-Correa

> Pattern recognition Lab, University of Erlangen-Nuremberg Faculty of Engineering, University of Antioquia, juan.vasquez@fau.de

class phonet.**Phonet** (*phonological_classes*)

Phonet computes posteriors probabilities of phonological classes from audio files for several groups of phonemes.

> **Parameters phonological_classes** – phonological class to be evaluated ("consonantal", "back", "anterior", "open", "close", "nasal", "stop", "continuant", "lateral", "flap", "trill", "voice", "strident", "labial", "dental", "velar", "pause", "vocalic", "all").
>
> **Returns** Phonet Object (see Examples).

phonological_classes=='all' computes the phonological posterior for the complete list of phonological classes.

get_PLLR (*audio_file*, *feat_file=''*, *projected=True*, *plot_flag=False*)

Estimate the phonological log-likelihood ratio (PLLR) features for an audio file (.wav) sampled at 16kHz

> **Parameters**
>
> - **audio_file** – file audio (.wav) sampled at 16 kHz
>
> - **feat_file** – .csv file to save the PLLR features for the phonological classes. Deafult="" does not save the csv file
>
> **Projected** whether to make a projection of the feature space of the PLLR according to [1], in order to avoid the bounding effect.
>
> **Plot_flag** True or False. Plot distributions of the feature space
>
> **Returns** Pandas dataFrame with the PLLR features

```
>>> phon=Phonet(["all"])
>>> file_audio=PATH+"/audios/sentence.wav"
>>> phon.get_PLLR(file_audio)
```

References:

[1] Diez, M., Varona, A., Penagarikano, M., Rodriguez-Fuentes, L. J., & Bordel, G. (2014). On the projection of PLLRs for unbounded feature distributions in spoken language recognition. IEEE Signal Processing Letters, 21(9), 1073-1077.

[2] Abad, A., Ribeiro, E., Kepler, F., Astudillo, R. F., & Trancoso, I. (2016). Exploiting Phone Log-Likelihood Ratio Features for the Detection of the Native Language of Non-Native English Speakers. In INTERSPEECH (pp. 2413-2417).

get_feat (*signal*, *fs*)

> This method extracts log-Mel-filterbank energies used as inputs of the model.
>
> > **Parameters**
> >
> > > - **signal** – the audio signal from which to compute features. Should be an N array.
> > >
> > > - **fs** – the sample rate of the signal we are working with, in Hz.
> >
> > **Returns** A numpy array of size (NUMFRAMES by 33 log-Mel-filterbank energies) containing features. Each row holds 1 feature vector.

get_phon_path (*audio_path*, *feat_path*, *plot_flag=False*)

> Estimate the phonological classes using the BGRU models for all the (.wav) audio files included inside a directory
>
> > **Parameters**
> >
> > > - **audio_path** – directory with (.wav) audio files inside, sampled at 16 kHz
> > >
> > > - **feat_path** – directory were the computed phonological posteriros will be stores as a (.csv) file per (.wav) file from the input directory
> > >
> > > - **plot_flag** – True or False, whether you want plots of phonological classes or not
> >
> > **Returns** A directory with csv files created with the posterior probabilities for the phonological classes.

```
>>> phon=Phonet(["vocalic", "strident", "nasal", "back", "stop", "pause"])
>>> phon.get_phon_path(PATH+"/audios/", PATH+"/phonclasses2/")
```

get_phon_wav (*audio_file*, *feat_file=''*, *plot_flag=True*)

> Estimate the phonological classes using the BGRU models for an audio file (.wav)
>
> > **Parameters**
> >
> > > - **audio_file** – file audio (.wav) sampled at 16 kHz
> > >
> > > - **feat_file** – . File (.csv) to save the posteriors for the phonological classes. Deafult="" does not save the csv file
> > >
> > > - **plot_flag** – True or False, whether you want plots of phonological classes or not
> >
> > **Returns** A pandas dataFrame with the posterior probabilities for the phonological classes.

```
>>> phon=Phonet(["stop"]) # get the "stop" phonological posterior from a
→single file
>>> file_audio=PATH+"/audios/pataka.wav"
>>> file_feat=PATH+"/phonclasses/pataka"
>>> phon.get_phon_wav(file_audio, file_feat, True)
```

```
>>> file_audio=PATH+"/audios/sentence.wav"
>>> file_feat=PATH+"/phonclasses/sentence_nasal"
>>> phon=Phonet(["nasal"]) # get the "nasal" phonological posterior from a
→single file
>>> phon.get_phon_wav(file_audio, file_feat, True)
```

```
>>> file_audio=PATH+"/audios/sentence.wav"
>>> file_feat=PATH+"/phonclasses/sentence_nasal"
>>> phon=Phonet(["strident", "nasal", "back"]) # get "strident, nasal, and
→back" phonological posterior from a single file
>>> phon.get_phon_wav(file_audio, file_feat, True)
```

get_posteriorgram (*audio_file*)

Estimate the posteriorgram for an audio file (.wav) sampled at 16kHz

> **Parameters audio_file** – file audio (.wav) sampled at 16 kHz
>
> **Returns** plot of the posteriorgram

```
>>> phon=Phonet(["vocalic", "strident", "nasal", "back", "stop", "pause"])
>>> phon.get_posteriorgram(file_audio)
```

mask_correction (*posterior*, *threshold=0.5*)

Implements a mask for a correction the posterior probabilities

> **Parameters**
>
> • **posterior** – phonological posterior.
>
> • **threshold** – threshold for correction
>
> **Returns** Corrected phonological posterior.

model (*input_size*)

This is the architecture used for the estimation of the phonological classes using a multitask learning strategy It consists of a 2 Bidirectional GRU layers, followed by a time-distributed dense layer

> **Parameters input_size** – size of input for the BGRU layers (number of features x sequence length).
>
> **Returns** A Keras model of a 2-layer BGRU neural network.

modelp (*input_size*)

This is the architecture used for phoneme recognition It consists of a 2 Bidirectional GRU layers, followed by a time-distributed dense layer

> **Parameters input_size** – size of input for the BGRU layers (number of features x sequence length).
>
> **Returns** A Keras model of a 2-layer BGRU neural network.

number2phoneme (*seq*)

Converts the prediction of the neural network for phoneme recognition to a list of phonemes.

> **Parameters seq** – sequence of integers obtained from the preiction of the neural network for phoneme recognition.
>
> **Returns** A list of strings of the phonemes recognized for each time-frame.

INDICES AND TABLES

- genindex
- modindex
- search

HELP

If you have trouble with Phonet, please write to Camilo Vasquez at: juan.vasquez@fau.de

PYTHON MODULE INDEX

p

phonet, **??**