The Journal of Open Source Software

SpeechPy - A Library for Speech Processing and Recognition

Amirsina Torfi[1]

1 Virginia Tech, Department of Computer Science

DOI: 10.21105/joss.00749

Software

- Review
- Repository
- Archive

Submitted: 14 May 2018
Published: 23 July 2018

Abstract

SpeechPy is an open source Python package that contains speech preprocessing techniques, speech features, and important post-processing operations. It provides most frequent used speech features including MFCCs and filterbank energies alongside with the log-energy of filter-banks. The aim of the package is to provide researchers with a simple tool for speech feature extraction and processing purposes in applications such as Automatic Speech Recognition and Speaker Verification.

Introduction

Automatic Speech Recognition (ASR) requires three main components for further analysis: Preprocessing, feature extraction, and post-processing. Feature extraction, in an abstract meaning, is extracting descriptive features from raw signal for speech classification purposes. Due to the high dimensionality, the raw signal can be less informative compared to extracted higher level features. Feature extraction comes to our rescue for turning the high dimensional signal to a lower dimensional and yet a more informative version of that for sound recognition and classification (Furui 1986; Guyon et al. 2008; Hirsch and Pearce 2000).

Feature extraction, in essence, should be done considering the specific application at hand. For example, in ASR applications, the linguistic characteristics of the raw signal are of great importance and the other characteristics must be ignored (D. Yu and Deng 2016; Rabiner and Juang 1993). On the other hand, in Speaker Recognition (SR) task, solely voice-associated information must be contained in the extracted feature (Campbell 1997). So the feature extraction goal is to extract the relevant feature from the raw signal and

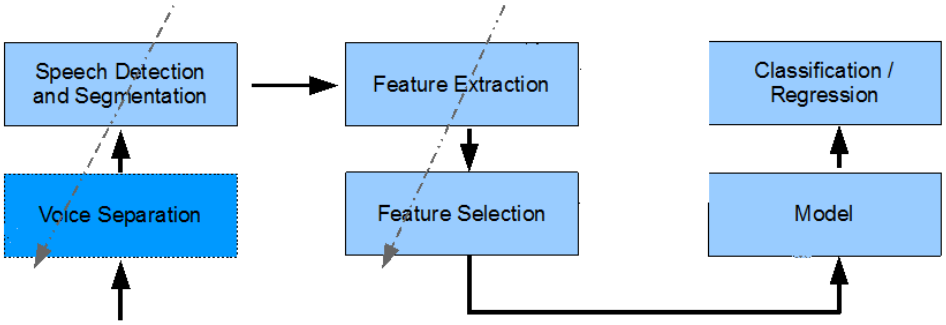

Figure 1: Scheme of speech recognition

map it to a lower dimensional feature space. The problem of feature extraction has been investigated in pattern classification aimed at preventing the curse of dimensionality. There are some feature extraction approaches based on information theory (Amisina Torfi, Soleymani, and Vakili 2017; Shannon 2001) applied to multimodal signals and demonstrated promising results (Gurban and Thiran 2009).

The speech features can be categorized into two general types of acoustic and linguistic features. The former one is mainly related to non-verbal sounds and the later one is associated with ASR and SR systems for which verbal part has the major role. Perhaps one of the most famous linguistic feature which is hard to beat is the Mel-Frequency Cepstral Coefficients (MFCC). It uses speech raw frames in the range from 20ms to 40ms for having stationary characteristics (Rabiner and Juang 1993). MFCC is widely used for both ASR and SR tasks and more recently in the associated deep learning applications as the input to the network rather than directly feeding the signal (Deng et al. 2013; Lee et al. 2009; D. Yu and Seltzer 2011). With the advent of deep learning (LeCun, Bengio, and Hinton 2015; Amirsina Torfi and Shirvani 2018), major improvements have been achieved by using deep neural networks rather than traditional methods for speech recognition applications (Variani et al. 2014; Hinton et al. 2012; Liu et al. 2015).

With the availability of free software for speech recognition such as VOICEBOX, most of these softwares are Matlab-based which limits their reproducibility due to commercial issues. Another great package is PyAudioAnalysis (Giannakopoulos 2015), which is a the comprehensive package developed in Python. However, the issue with PyAudioAnalysis is that its complexity and being too verbose for extracting simple features and it also lacks some important preprocessing and post-processing operations for its current version.

Considering the recent advent of deep learning in ASR and SR and the importance of the accurate speech feature extraction, here are the motivations behind SpeechPy package:

- Developing a free open source package which covers important preprocessing techniques, speech features, and post-processing operations required for ASR and SR applications.
- A simple package with a minimum degree of complexity should be available for beginners.
- A well-tested and continuously integrated package for future developments should be developed.

SpeechPy has been developed to satisfy the aforementioned needs. It contains the most important preprocessing and post-processing operations and a selection of frequently used speech features. The package is free and released as an open source software. Continuous integration using for instant error check and validity of changes has been deployed for SpeechPy. Moreover, prior to the latest official release of SpeechPy, the package has successfully been utilized for research purposes (Amirsina Torfi et al. 2017; Amirsina Torfi, Nasrabadi, and Dawson 2017).

Package Eco-system

SpeechPy has been developed using Python language for its interface and backed as well. An empirical study demonstrated that Python as a scripting language, is more effective and productive than conventional languages for some programming problems and memory consumption is often “better than Java and not much worse than C or C++” (Prechelt 2000). We chose Python due to its simplicity and popularity. Third-party libraries are avoided except *Numpy* and *Scipy* for handling data and numeric computations.

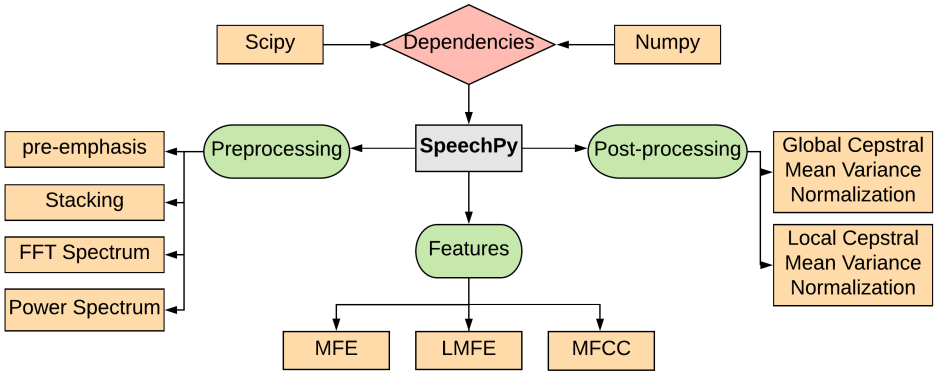

Figure 2: A general view of the package

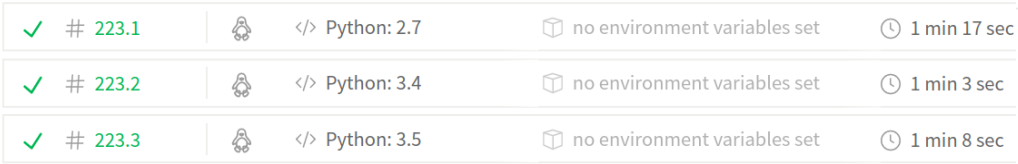

Figure 3: Travic CI web interface after testing SpeechPy against a new change

Complexity

As the user should not and does not even need to manipulate the internal package structure, object-oriented programming is mostly used for package development which provides an easier interface for the user with a sacrifice to the simplicity of the code. However, the internal code complexity of the package does not affect the user experience since the modules can easily be called with the associated arguments. SpeechPy is a library with a collection of sub-modules.

Code Style and Documentation

SpeechPy is constructed based on PEP 8 style guide for Python codes. Moreover, it is extensively documented using the formatted docstrings and Sphinx for further automatic modifications to the document in case of changing internal modules. The full documentation of the project will be generated in HTML and PDF format using Sphinx and is hosted online. The official releases of the project are hosted on the Zenodo as well (Amirsina Torfi 2017).

Continuous Testing and Extensibility

The output of each function has been evaluated as well as using different tests as opposed to the other existing standard packages. For continuous testing, the code is hosted on GitHub and integrated with Travis CI. Each modification to the code must pass the unit tests defined for the continuous integration. This will ensure the package does not break with unadapted code scripts. However, the validity of the modifications should always be investigated with the owner or authorized collaborators of the project. The code will be tested at each time of modification for Python versions *"2.7"*, *"3.4"* and *"3.5"*. In the future, these versions are subject to change.

Availability

Operating system

Tested on Ubuntu 14.04 and 16.04 LTS Linux, Apple Mac OS X 10.9.5 , and Microsoft Windows 7 & 10. We expect that SpeechPy works on any distribution as long as Python and the package dependencies are installed.

Programming language

The package has been tested with Python 2.7, 3.4 and 3.5. However, using Python 3.5 is suggested.

Additional system requirements & dependencies

SpeechPy is a light package and small computational power would be enough for running it. Although the speed of the execution is totally dependent on the system architecture. The dependencies are as follows:

- Numpy
- SciPy

Acknowledgement

This work has been completed with computational resources provided by the West Virginia University and Virginia Tech and is based upon a work supported by the Center for Identification Technology Research (CITeR) and the National Science Foundation (NSF) under Grant #1650474. I would like to thank professor Nasser Nasrabadi for supporting me through this project and for his valuable supervision regarding my research in speech technology.

References

Campbell, Joseph P. 1997. "Speaker Recognition: A Tutorial." *Proceedings of the IEEE* 85 (9). IEEE:1437–62.

Deng, Li, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, et al. 2013. "Recent Advances in Deep Learning for Speech Research at Microsoft." In *Acoustics, Speech and Signal Processing (Icassp), 2013 Ieee International Conference on*, 8604–8. IEEE.

Furui, Sadaoki. 1986. "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34 (1). IEEE:52–59.

Giannakopoulos, Theodoros. 2015. "PyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis." *PloS One* 10 (12). Public Library of Science.

Gurban, Mihai, and Jean-Philippe Thiran. 2009. "Information Theoretic Feature Extraction for Audio-Visual Speech Recognition." *IEEE Transactions on Signal Processing* 57 (12). IEEE:4765–76.

Guyon, Isabelle, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. 2008. *Feature Extraction: Foundations and Applications*. Vol. 207. Springer.

Hinton, Geoffrey, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, et al. 2012. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." *IEEE Signal Processing Magazine* 29 (6). IEEE:82–97.

Hirsch, Hans-Günter, and David Pearce. 2000. "The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems Under Noisy Conditions." In *ASR2000-Automatic Speech Recognition: Challenges for the New Millenium Isca Tutorial and Research Workshop (Itrw)*.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521 (7553). Nature Publishing Group:436.

Lee, Honglak, Peter Pham, Yan Largman, and Andrew Y Ng. 2009. "Unsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks." In *Advances in Neural Information Processing Systems*, 1096–1104.

Liu, Yuan, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu. 2015. "Deep Feature for Text-Dependent Speaker Verification." *Speech Communication* 73. Elsevier:1–13.

Prechelt, Lutz. 2000. "An Empirical Comparison of c, C++, Java, Perl, Python, Rexx and Tcl." *IEEE Computer* 33 (10):23–29.

Rabiner, Lawrence R, and Biing-Hwang Juang. 1993. *Fundamentals of Speech Recognition*. Vol. 14. PTR Prentice Hall Englewood Cliffs.

Shannon, Claude Elwood. 2001. "A Mathematical Theory of Communication." *ACM SIGMOBILE Mobile Computing and Communications Review* 5 (1). ACM:3–55.

Torfi, Amirsina. 2017. "SpeechPy: Speech recognition and feature extraction." https://doi.org/10.5281/zenodo.810391.

Torfi, Amirsina, and Rouzbeh A Shirvani. 2018. "Attention-Based Guided Structured Sparsity of Deep Neural Networks." *arXiv Preprint arXiv:1802.09902*.

Torfi, Amirsina, Seyed Mehdi Iranmanesh, Nasser Nasrabadi, and Jeremy Dawson. 2017. "3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition." *IEEE Access* 5. IEEE:22081–91.

Torfi, Amirsina, Nasser M Nasrabadi, and Jeremy Dawson. 2017. "Text-Independent Speaker Verification Using 3d Convolutional Neural Networks." *arXiv Preprint arXiv:1705.09422*.

Torfi, Amisina, Sobhan Soleymani, and Vahid Tabataba Vakili. 2017. "On the Construction of Polar Codes for Achieving the Capacity of Marginal Channels." *arXiv Preprint arXiv:1707.04512*.

Variani, Ehsan, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. 2014. "Deep Neural Networks for Small Footprint Text-Dependent Speaker Verification." In *Acoustics, Speech and Signal Processing (Icassp), 2014 Ieee International Conference on*, 4052–6. IEEE.

Yu, Dong, and Li Deng. 2016. *AUTOMATIC Speech Recognition*. Springer.

Yu, Dong, and Michael L Seltzer. 2011. "Improved Bottleneck Features Using Pretrained Deep Neural Networks." In *Twelfth Annual Conference of the International Speech Communication Association*.