The Journal of Open Source Software

audiomate: A Python package for working with audio datasets

Matthias Büchi[1] **and Andreas Ahlenstorf**[1]

1 ZHAW, Winterthur, Switzerland

DOI: 10.21105/joss.0XXXX

Software

- Review
- Repository
- Archive

Editor: Editor Name

Submitted: 01 January 1900
Published: 01 January 3030

Summary

Machine Learning has a variety of applications in different fields, including audio-related tasks. But especially with deep learning the need for data has been increased dramatically. Therefore data has to be prepared carefully before the actual problem can be tackled. Audiomate aims to reduce the effort for the preparation of audio datasets, by providing a consistent way to work with them.

Almost every single audio dataset has its own format. So every time a new dataset is used, a piece of code has to be written to load and parse the data. Having the data loaded, often some kind of preprocessing is necessary. Finally, if the data is intended to be used with a machine learning toolkit, it has to be saved in another specific format.

Audiomate is designed to help with all those problems. It provides a uniform way to work with the data without knowing specific details about the file-system representation of the data. This way datasets of different formats can be loaded and used in the same way. Furthermore, the data can be stored in different ways. For example, a machine learning toolkit like Kaldi (Povey et al., 2011) requires a specific format. Audiomate automatically exports the data in the correct format, without the user having to know how that format looks like. Apart from reading and writing data, audiomate also provides tools to work with the loaded data including:

- Access all meta-data, like speakers, labels
- Read audio data (single files, batches of files)
- Retrieve information about the data (e.g. amount of speakers, total duration, ...)
- Merge data of multiple datasets (e.g. Combine two speech datasets)
- Split data into smaller subsets (e.g. Create train/dev/test splits with a reasonable distribution of classes)
- Validate data for specific requirements (e.g Check if all samples have assigned a label)

Use cases

Audiomate can be used for a variety of use cases. Two of them are described here to show possible scenarios where the usage of audiomate might be helpful.

Training ASR Model with Mozilla's DeepSpeech

For training an automatic speech recognition model, for example Mozilla's implementation of DeepSpeech (https://github.com/mozilla/DeepSpeech) can be used. In order to do that, we

first need to convert our dataset to the format of the DeepSpeech implementation. Using audiomate, we can load the given dataset and store it in the format of DeepSpeech. Of course, it requires audiomate to have the specific reader and writer. But, once having them, they can be used in many ways. Assume, we decide to change to Kaldi instead of DeepSpeech, we just change one line to use another audiomate writer that stores the data in Kaldi's format.

Training a Neural Network recognizing Music

We want to train a neural network detecting segments in an audio stream that contains music. To do so, we intend to use the MUSAN dataset (Snyder, Chen, & Povey, 2015) and the GTZAN dataset ("GTZAN music/speech collection," n.d.). For training the DNN, we use for example PyTorch (https://pytorch.org/). With audiomate, we can load both datasets with a single line of code. Furthermore, we can merge them to a single set. In order to test our model, we also can split the data into two subsets with audiomate. Without knowing how the audio is stored on the filesystem, we can load the samples and labels just by iterating over all utterances or using a method to load the samples in batches.

Implementation

Audiomate is implemented with the goal to make it simple to add new data formats. A new format can be added by implementing one or more of three available abstract interfaces.

- Reader: A reader defines the procedure to load data from a specific format. It stores the data in a universal data structure.
- Writer: A writer defines the procedure to store data in a specific format. It gets data from the universal data structure and stores it in the specific format.
- Download: A downloader can be used to download a dataset. It downloads all required files automatically.

One certain format rarely defines interfaces for all of those tasks. Mostly, readers and downloaders are implemented for datasets, while writers are implemented for machine learning toolkits. Audiomate already provides a bunch of implementations for various datasets and toolkits.

Acknowledgements

Parts of audiomate were developed during the project Keyword Spider (KWS) funded by INNOSUISSE (26567.1 PFES-ES).

References

GTZAN music/speech collection. (n.d.). http://marsyas.info/downloads/datasets.html.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. K., Hannemann, M., et al. (2011). The kaldi speech recognition toolkit. In.

Snyder, D., Chen, G., & Povey, D. (2015). MUSAN: A Music, Speech, and Noise Corpus.