The Journal of Open Source Software

Nkululeko: A Python package to predict speaker characteristics with a high-level interface.

Felix Burkhardt [1,2*] **and Bagus Tris Atmaja** [3*]

1 audEERING GmbH, Germany **2** TU Berlin, Germany **3** National Institute of Advanced Industrial Science and Technology (AIST), Japan * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Nkululeko (Burkhardt et al., 2022) is open-source software written in Python and hosted on GitHub. It is predominantly a framework for audio-based machine learning explorations without the need to write Python code, and is strongly based on machine learning packages like sklearn (Pedregosa et al., 2011) and pytorch (Chaudhary et al., 2020). The main features are: training and evaluation of labelled speech databases with state-of-the-art machine learning approach and acoustic feature extractors, a live demonstration interface, and the possibility to store databases with predicted labels. Based on this, the framework can also be used to check on bias in databases by exploring correlations of target labels, like, e.g. depression or diagnosis, with predicted, or additionally given, labels like age, gender, signal distortion ratio or mean opinion score.

Design Choices

The program is intended for **novice** people interested in speaker characteristics detection (e.g., emotion, age, and gender) without proficient in (Python) programming language. Its main target is for **education** and **research** with the main features as follows:

- Finding good combinations of variables, e.g., acoustic features, models (classifier or regressor), feature standardization, augmentation, etc., for speaker characteristics detection (e.g., emotion);
- Characteristics of the database, such as distribution of gender, age, emotion, duration, data size, and so on with their visualization;
- Inference of speaker characteristics from a given audio file or streaming audio (can be said also as "weak" labeling for semi-supervised learning).

Hence, one should be able to use Nkululeko after installing and preparing/downloading their data in the correct format.

```
$ nkululeko.MODULE_NAME --config CONFIG_FILE.ini
```

How does it work?

nkululeko is a command line tool written in Python, best used in conjunction with the Visual Studio code editor (but can be run stand-alone). To use it, a text editor is needed to edit the experiment configuration. You would then run nkululeko like this:

```
$ nkululeko.explore --config conf.ini
```

and inspect the results afterward; they are represented as images, texts, and even a fully automatically compiled PDF report written in latex.

nkululeko's data import format is based on a simple CSV formalism, or alternatively, for a more detailed representation including data schemata, audformat.[1] Basically, to be used by nkululeko, the data format should include the audio file path and a task-specific label. Optionally, speaker ID and gender labels help with speech data. An example of a database labelled with emotion is

```
file, speaker, gender, emotion
x/sample.wav, s1, female, happy
...
```

As the main goal of nkululeko is to avoid the need to learn programming, experiments are specified by means of a configuration file. The functionality is encapsulated by software *modules* (interfaces) that are to be called on the command line. We list the most important ones here:

- **nkululeko**: do machine learning experiments combining features and learners
- **demo**: demo the current best model on the command line
- **explore**: perform data exploration (used mainly in this paper)
- **augment**: augment the current training data. This could also be used to reduce bias in the data, for example, by adding noise to audio samples that belong to a specific category.
- ** **aug_train**: augment the training data and train the model with the augmented data.
- **predict**: predict features like signal distortion ratio, mean opinion score, arousal/valence, age/gender (for databases that miss this information), with deep neural nets models, e.g. as a basis for the *explore* module.
- **segment**: segment a database based on VAD (voice activity detection)
- **ensemble**: ensemble several models to improve performance

The configuration (INI) file consists of a set of key-value pairs that are organised into several sections. Almost all keys have default values, so they do not have to be specified.

Here is a sample listing of an INI file with database section:

```
[EXP]
name = explore-androids
[DATA]
databases = ['androids']
androids = /data/androids/androids.csv
target = depression
labels = ['depressed', 'control']
samples_per_speaker = 20
min_length = 2
[PREDICT]
sample_selection = all
targets = ['pesq', 'sdr', 'stoi', 'mos']
[EXPL]
value_counts = [['gender'], ['age'], ['est_sdr'], ['est_pesq'], ['est_mos']]
[REPORT]
latex = androids-report
```

As can be seen, some of the values simply contain Python data structures like arrays or dictionaries. Within this example, an experiment is specified with the name *explore-androids*, and a result folder with this name will be created, containing all figures and textual results, including an automatically generated Latex and PDF report on the findings.

[1] https://audeering.github.io/audformat/

The *DATA* section sets the location of the database and specifies filters on the sample, in this case limiting the data to 20 samples per speaker at most and at least 2 seconds long. In this section, the split sets (training, development, and test) are also specified. There is a special feature named *balance splits* that lets the user specify criteria that should be used to stratify the splits, for example, based on signal distortion ratio.

With the *predict* module, specific features like, for example, signal distortion ratio or mean opinion score are to be predicted by deep learning models. The results are then used by a following call to the *explore* module to check whether these features, as well as some ground truth features (*age* and *gender*), correlate with the target variable (*depressed* in the given example) in any way.

The `nkululeko` configuration can specify further sections:

- **FEATS** to specify acoustic features (e.g. opensmile (Eyben et al., 2010) or deep learning embeddings; e.g. wav2vec 2.0 (Baevski et al., 2020)) that should be used to represent the audio files.
- **MODEL** to specify statistical models for regression or classification of audio data.

Statement of need

Open-source tools are believed to be one of the reasons for accelerated science and technology. They are more secure, easy to customise and transparent. There are several open-source tools that exist for acoustic, sound, and audio analysis, such as librosa (McFee et al., 2015), TorchAudio (Yang et al., 2021), pyAudioAnalysis (Giannakopoulos, 2015), ESPNET (Watanabe et al., 2018), and SpeechBrain (Ravanelli et al., 2021). However, none of them are specialised in speech analysis with high-level interfaces for novices in the speech processing area.

One exception is Spotlight (Suwelack, 2023), an open-source tool that visualises metadata distributions in audio data. An existing interface between `nkululeko` and Spotlight can be used to combine the visualisations of Spotlight with the functionalities of Nkululeko.

Nkululeko follows these principles:

- *Minimum programming skills*: the only programming skills required are to prepare the data in the correct (CSV) format and to run the command line tool. For AUDFORMAT, no preparation is needed.
- *Standardised data format and label*: the data format is based on CSV and AUFORMAT, which is a widely used format for data exchange. The standard headers are like 'file', 'speaker', 'emotion', 'age', and 'language' but also can be customised.
- *Replicability*: the experiments are specified in a configuration file, which can be shared with others including the splitting of training, development, and test partition. All results are stored in a folder with the same name as the experiment.
- *High-level interface*: the user specifies the experiment in an INI file, which is a simple text file that can be edited with any text editor. The user does not need to write Python code for experiments.
- *Transparency*: as CLI, nkululeko *always output debug*, in which info, warning, and error will be displayed in terminal (and should be easily understood). The results are stored in the experiment folder for further investigations and are represented as images, texts, and even a fully automatically compiled PDF report written in latex.

Usage in Existing Research

Nkululeko has been used in several research projects since its first appearance in 2022 (Burkhardt et al., 2022). The following list gives an overview of the research papers that have used Nkululeko:

- (**?**): this paper reported a database development of synthesized speech for basic emotions and its evaluation using Nkululeko toolkit.
-
-

Acknowledgements

We acknowledge support from these various projects:

- European SHIFT (*MetamorphoSis of cultural Heritage Into augmented hypermedia assets For enhanced accessibiliTy and inclusion*) project (Grant Agreement number: 101060660);
- European EASIER (*Intelligent Automatic Sign Language Translation*) project (Grant Agreement number: 101016982);
- Project JPNP20006 commisioned by the New Energy and Industrial Technology Development Organization (NEDO), Japan;
- Project 24K02967 from the Japan Society for the Promotion of Science (JSPS).

We thank audEERING GmbH for partial funding.

References

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 12449–12460). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf

Burkhardt, F., Wagner, J., Wierstorf, H., Eyben, F., & Schuller, B. (2022). Nkululeko: A tool for rapid speaker characteristics detection. *2022 Language Resources and Evaluation Conference, LREC 2022*, 1925–1932. ISBN: 9791095546726

Chaudhary, A., Chouhan, K. S., Gajrani, J., & Sharma, B. (2020). *Deep learning with PyTorch*. https://doi.org/10.4018/978-1-7998-3095-5.ch003

Eyben, F., Wöllmer, M., & Schuller, B. (2010). openSMILE – the munich versatile and fast open-source audio feature extractor. *MM'10 - Proceedings of the ACM Multimedia 2010 International Conference*, 1459–1462. https://doi.org/10.1145/1873951.1874246

Giannakopoulos, T. (2015). pyAudioAnalysis: An open-source python library for audio signal analysis. *PLoS One*, *10*(12), 1–17. https://doi.org/10.1371/journal.pone.0144610

McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. *Proc. 14th Python Sci. Conf., Scipy*, 18–24. https://doi.org/10.25080/majora-7b98e3ed-003

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,

Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., … Bengio, Y. (2021). *SpeechBrain: A general-purpose speech toolkit*. https://arxiv.org/abs/2106.04624

Suwelack, S. (2023). Spotlight. In *GitHub repository*. https://github.com/Renumics/spotlight/; GitHub.

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N. E. Y., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., & Ochiai, T. (2018). ESPNet: End-to-end speech processing toolkit. *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, *2018-Septe*(September), 2207–2211. https://doi.org/10.21437/Interspeech.2018-1456

Yang, S., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., Lee, K., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H. (2021). SUPERB: Speech Processing Universal PERformance Benchmark. *Interspeech 2021*, 1194–1198. https://doi.org/10.21437/Interspeech.2021-1775