The Journal of Open Source Software

Phonemizer: Text to Phones Transcription for Multiple Languages in Python

Mathieu Bernard[1] **and Hadrien Titeux**[1]

1 LSCP/ENS/CNRS/EHESS/Inria/PSL Research University, Paris, France

DOI: 10.21105/joss.03958

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @henrykironde
- @chrisbrickhouse

Submitted: 25 October 2021
Published: 29 November 2021

Summary

Phones are elementary sounds the speech is made of, on which syllables and words are built. The transcription of texts from their orthographic form into a phonetic alphabet is an important requirement in various applications related to speech and language processing, for instance for text to speech systems. `Phonemizer` is a Python package addressing precisely this issue: it transcribes a text from its orthographic representation into a phonetic one. The package is user-friendly and exposes a single `phonemize` function, also available as a command-line interface. It supports about a hundred different languages and provides end-user functionalities such as punctuation preservation, phones accentuation, tokenization at phone/syllable/word levels, as well as parallel processing of large input texts.

Statement of Need

Whereas the high-level features introduced above are implemented directly by `phonemizer`, the phonetic transcription itself is delegated to third party backends, wrapped in an homogoneous interface by the package. The default backend used by `phonemizer` is eSpeak (Dunn & Vitolins, 2019), a text to speech software built on linguistic expertise and hand written transcription rules. It transcribes text into the International Phonetic Alphabet and supports more than a hundred languages. Using MBROLA voices (Tits & Vitolins, 2019), available for 35 languages, the eSpeak backend transcribes text in the SAMPA computer readable phonetic alphabet. Festival (Black et al., 2014) is another text to speech software used as a backend for `phonemizer`. It is available for American English only, and uses a non standard phoneset for transcription, but this backend is the only one to meet the requirement of some applications by preserving syllable boundaries. The third `phonemizer` backend is Segments (Forkel et al., 2019), a Python package providing Unicode Standard tokenization routines and orthography segmentation. It relies on a grapheme to phone mapping to generate the transcription. This backend is mostly usefull for low-resource languages, for which users with linguistic expertise can write their own mappings. Six languages are provided as exemples with `phonemizer`: Chintang, Cree, Inuktitut, Japanese, Sesotho and Yucatec.

Text to phones transcription is a critical need in different applications related to natural language and speech processing. So far, the `phonemizer` package has been used in the preprocessing pipeline of various deep learning text to speech systems (Ideas Engineering, 2021; Mozilla, 2021; Watanabe et al., 2018). It has also been used as a preprocessing step in word segmentation studies regarding the role of speech prosody in segmentability (Ludusan et al., 2017) and the psychology of child development (Bernard et al., 2020; Cristia et al., 2019). A phonetic transcription generated by the package was used to evaluate a phone discrimination task for the Zero Speech Challenge 2017 (Dunbar et al., 2017). Finally, the `phonemizer` is very suitable to prepare datasets for their use with the Kaldi speech recognition toolkit (Povey et al., 2011), where a phonetic transcription of text is a requirement for various algorithms.

Acknowledgements

We are thankful to Alex Cristia who initiated this project and to Emmanuel Dupoux for his support and advices. We also thank the `phonemizer` users for their bug reports and features requests. This work is funded by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-17-EURE-0017 Frontcog, ANR-10-IDEX-0001-02 PSL, ANR-19-P3IA-0001 PRAIRIE 3IA Institute) and grants from CIFAR (Learning in Machines and Brains), Facebook AI Research (Research Grant), Google (Faculty Research Award), Microsoft Research (Azure Credits and Grant), and Amazon Web Service (AWS Research Credits).

References

Bernard, M., Thiolliere, R., Saksida, A., Loukatou, G. R., Larsen, E., Johnson, M., Fibla, L., Dupoux, E., Daland, R., Cao, X. N., & Alejandrina, C. (2020). WordSeg: Standardizing unsupervised word form segmentation from text. *Behavior Research Methods*, *52*(1), 264–278.

Black, A. W., Clark, R., Richmond, K., Yamagishi, J., Oura, K., & King, S. (2014). *The festival speech synthesis system* (Version 2.4). CSTR, University of Edinburgh. https://www.cstr.ed.ac.uk/projects/festival

Cristia, A., Dupoux, E., Ratner, N. B., & Soderstrom, M. (2019). Segmentability differences between child-directed and adult-directed speech: A systematic test with an ecologically valid corpus. *Open Mind*, *3*, 13–22.

Dunbar, E., Cao, X. N., Benjumea, J., Karadayi, J., Bernard, M., Besacier, L., Anguera, X., & Dupoux, E. (2017). The zero resource speech challenge 2017. *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 323–330.

Dunn, R. H., & Vitolins, V. (2019). eSpeak NG speech synthetizer. In *GitHub repository* (Version 1.50). GitHub. https://github.com/espeak-ng/espeak-ng

Forkel, R., Moran, S., List, J.-M., Greenhill, S. J., Ashby, L., Gorman, K., & Kaiping, G. (2019). *Cldf/segments: Unicode standard tokenization* (Version v2.1.3). Zenodo. https://doi.org/10.5281/zenodo.3549784

Ideas Engineering. (2021). Non-autoregressive transformer based neural network for text-to-speech. In *GitHub repository*. GitHub. https://github.com/as-ideas/TransformerTTS

Ludusan, B., Mazuka, R., Bernard, M., Cristia, A., & Dupoux, E. (2017). The role of prosody and speech register in word segmentation: A computational modelling perspective. *Proceedings of the Association for Computational Linguistics*, 178–183.

Mozilla. (2021). Deep learning for text to speech. In *GitHub repository*. GitHub. https://github.com/mozilla/TTS

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., & Vesely, K. (2011). The kaldi speech recognition toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

Tits, N., & Vitolins, V. (2019). MBROLA. In *GitHub repostory* (Version 3.3). GitHub. https://github.com/numediart/MBROLA

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., & Ochiai, T. (2018). ESPnet: End-to-end speech processing toolkit. *Proceedings of Interspeech*, 2207–2211. https://doi.org/10.21437/Interspeech.2018-1456