HAL
open science

Textless-lib: a Library for Textless Spoken Language Processing

Eugene Kharitonov, Jade Copet, Kushal Lakhotia, Tu Anh Nguyen, Paden Tomasello, Ann Lee, Ali Elkahky, Wei-Ning Hsu, Abdelrahman Mohamed, Emmanuel Dupoux, et al.

To cite this version:

Eugene Kharitonov, Jade Copet, Kushal Lakhotia, Tu Anh Nguyen, Paden Tomasello, et al.. Textless-lib: a Library for Textless Spoken Language Processing. NAACL 2022 - Annual Conference of the North American Chapter of the Association for Computational Linguistics, Jul 2022, Seattle, United States. pp.1-9. hal-03831838

HAL Id: hal-03831838

https://inria.hal.science/hal-03831838v1

Submitted on 15 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

textless-lib: a Library for Textless Spoken Language Processing

Eugene Kharitonov★, Jade Copet★, Kushal Lakhotia▲, Tu Anh Nguyen★, Paden Tomasello★
Ann Lee★, Ali Elkahky★, Wei-Ning Hsu★, Abdelrahman Mohamed★, Emmanuel Dupoux★†, Yossi Adi★
★ Meta AI Research, † EHESS
▲ Outreach
{kharitonov, jadecopet, adiyoss}@fb.com

Abstract

Textless spoken language processing research aims to extend the applicability of standard NLP toolset onto spoken language and languages with few or no textual resources. In this paper, we introduce textless-lib, a PyTorch-based library aimed to facilitate research in this research area. We describe the building blocks that the library provides and demonstrate its usability by discuss three different use-case examples: (i) speaker probing, (ii) speech resynthesis and compression, and (iii) speech continuation. We believe that textless-lib substantially simplifies research the textless setting and will be handful not only for speech researchers but also for the NLP community at large. The code, documentation, and pre-trained models are available at https://github.com/facebookresearch/textlesslib/.

1 Introduction

Textless spoken language modeling (Lakhotia et al., 2021) consists in jointly learning the acoustic and linguistic characteristics of a natural language from raw audio samples without access to textual supervision (e.g. lexicon or transcriptions). This area of research has been made possible by converging progress in self-supervised speech representation learning (Schneider et al., 2019; Baevski et al., 2020; Oord et al., 2018; Hsu et al., 2021; Chorowski et al., 2021; Chen et al., 2021; Chung et al., 2021; Wang et al., 2021; Ao et al., 2021), language modeling (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Brown et al., 2020; Lewis et al., 2020), and speech synthesis (Ren et al., 2019; Kumar et al., 2019; Yamamoto et al., 2020; Ren et al., 2020; Kong et al., 2020; Morrison et al., 2021).

Lakhotia et al. (2021) presented a Generative Spoken Language Modeling (GSLM) pipeline trained from raw audio, consisting in a speech encoder (converting speech to discrete units), a language model (based on units) and a decoder (converting units back to speech). These components enabled the generation of new speech by sampling units from the language model. Polyak et al. (2021) proposed an improved encoder/decoder working from disentangled quantized content and F0 units and showed how such a system could be used for efficient audio compression. Kharitonov et al. (2021a) proposed a modified language model system capable of jointly modelling content units and F0 yielding expressive generations. Lastly, Kreuk et al. (2021) demonstrated that the language model can be replaced by a sequence to sequence model achieving the first high quality speech emotion conversion system (including laughter and yawning).

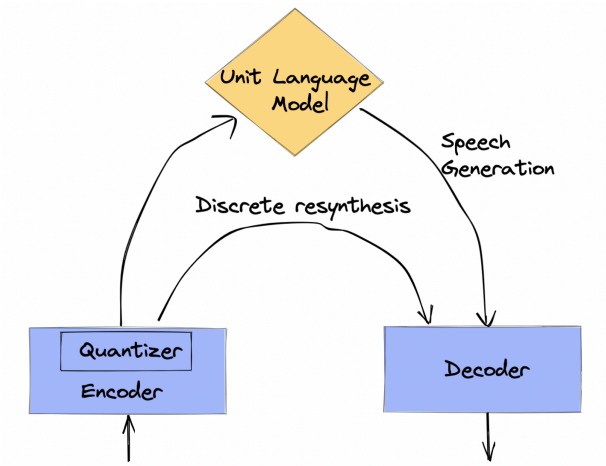

Figure 1: A visual description for textless modeling of spoken language. One can perform language modeling for speech continuations (Lakhotia et al., 2021) or a direct speech resynthesis (Polyak et al., 2021).

The textless approach has several advantages. First, it would be beneficial for the majority of the world's languages that do not have large textual resources or even a widely used standardized orthography (Swiss German, dialectal Arabic, Igbo, etc.). Despite being used by millions of people, these languages have little chance of being served by current

text-based technology. Moreover, "high-resource" languages can benefit from such modeling where the oral and written forms are mismatched in terms of lexicon and syntax. Second, directly modeling spoken language from raw audio allows us to go beyond lexical content and also model linguistically relevant signals such as prosodic features, intonation, non-verbal vocalizations (e.g., laughter, yawning, etc.), speaker identity, etc. All of these are virtually absent in text.

Although great progress has been made in modeling spoken language, it still requires domain expertise and involves a complicated setting. For instance, the official implementation of the GSLM pipeline (Lakhotia et al., 2021) consists of roughly four different launching scripts with a few dozens of checkpoints. Similarly, running the official implementation of Polyak et al. (2021), requires using four scripts from two different repositories.

We present `textless-lib`, a PyTorch library for textless spoken language processing. It makes processing, encoding, modeling, and generating of speech as simple as possible. With a few lines of code, one can perform speech continuation, audiobook compression, representation analysis by probing, speech-to-speech translation, etc. We provide all the necessary building blocks, example pipelines, and example tasks. We believe such a simple to use API will encourage both speech and NLP communities to deepen and extend the research work on modeling spoken language without text and unlock potential future research directions.

2 Background

Below we provide an overview of the common textless spoken language modeling pipeline. In a nutshell, such pipeline is usually comprised of: i) Speech-to-Units (S2U) encoders that automatically discover discrete representations or units which can be used to encode speech into "pseudo-text"; ii) Units-to-Units (U2U) models that are used for units modeling. This can take a form as Unit-Language-Model (uLM) for speech continuation (Lakhotia et al., 2021; Kharitonov et al., 2021a), sequence-to-sequence models for speech emotion conversion (Kreuk et al., 2021) or translation tasks (Lee et al., 2021a,b); iii) Units-to-Speech (U2S) models to reconstruct back the speech signals.

Alternatively, one could drop the U2U component and perform a direct speech resynthesis (Polyak et al., 2021). This can be used for speech compression, voice conversion, or developing a better understanding of the learned representation using probing methods. See Figure 1 for a visual description of the full system. We provide a detailed description for each of the above-mentioned components in the following subsections.

Type	Model	Dataset
Encoders	HuBERT	LS-960
	CPC	LL-6k
Quantizers	k-means	LS-960 w. 50 units
		LS-960 w. 100 units
		LS-960 w. 200 units
		LS-960 w. 500 units
F0 extract.	YAAPT	-
Decoders	Tacotron2	LJ Speech
	WaveGlow	LJ Speech

Table 1: Summary of the pre-trained models provided in `textless-lib`. We denote LibriSpeech and Libri-Light as LS-960 and LL-6k accordingly. All quantizers were trained on "dev-clean" partition of LibriSpeech.

2.1 Speech to Units

Consider the domain of audio samples as $\mathcal{X} \subset \mathbb{R}$. The representation for an audio waveform is therefore a sequence of samples $\boldsymbol{x} = (x^1, \ldots, x^T)$, where each $x^i \in \mathcal{X}$ for all $1 \leq t \leq T$. We denote the S2U encoder as, $E(\boldsymbol{x}) = \mathbf{z}$, where $\mathbf{z} = (\boldsymbol{z}^1, \ldots, \boldsymbol{z}^L)$ is a spectral representation of $\boldsymbol{x}$ sampled at a lower frequency, each $\boldsymbol{z}^i$ for $1 \leq i \leq L$ is a d-dimensional vector, and $L < T$.

Next, as the representations obtained by E are continuous, an additional quantization step is needed. We define a quantization function Q, which gets as input dense representations and outputs a sequence of discrete tokens corresponding to the inputs' quantized version. Formally, $Q(\mathbf{z}) = \boldsymbol{z}_q$, where $\boldsymbol{z}_q = (z_q^1, \ldots, z_q^L)$ such that $z_q^i \in \{1, \ldots, K\}$ and K is the size of the vocabulary. After quantization one can either operate on the original discrete sequences (duped) or collapse repeated units (e.g., $0, 0, 1, 1, 2 \rightarrow 0, 1, 2$), we refer to such sequences as "deduped". Working with the deduped sequences simplifies modeling long sequences, however, the tempo information is lost.

2.2 Units to Speech

Converting a sequence of units to audio is akin to the Text-to-Speech (TTS) problem, where we consider the discrete units as "pseudo-text". This can be solved by adopting a standard TTS architecture. For instance, Lakhotia et al. (2021) trained a Tacotron2 model (Shen et al., 2018) to perform units to mel-spectrogram conversion followed by

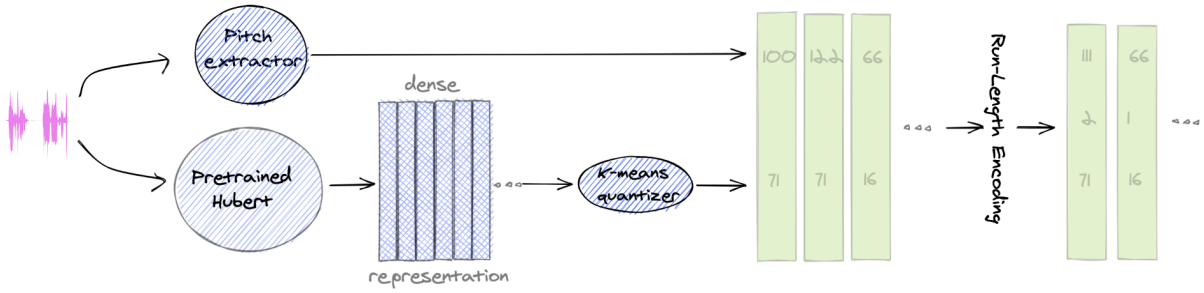

Figure 2: We represent speech as three aligned, synchronised streams: discrete pseudo-units, duration, and pitch.

a WaveGlow (Prenger et al., 2019) neural vocoder for time-domain reconstruction.

Formally, to reconstruct a time-domain speech signal from a sequence of discrete units, $\boldsymbol{z}_q$ we define the composition as, $V(G(\boldsymbol{z}_q)) = \hat{x}$, where G is a mel-spectrogram estimation module (e.g., Tacotron2), and V is a phase vocoder module responsible for time-domain synthesis (e.g., WaveGlow). The input sequence $\boldsymbol{z}_q$ can be either the original sequence or its deduped version.

Interestingly, one can simplify the synthesis process when working with the duped unit sequences. As we have a direct mapping between the duped discrete unit sequence to the time domain signal (e.g., each unit corresponds to a 20ms window) one can remove G, and directly feed $\boldsymbol{z}_q$ to V. This was successfully done in (Polyak et al., 2021) for speech resynthesis using the HiFi-GAN neural vocoder (Kong et al., 2020). Alternatively, as suggested by (Kreuk et al., 2021; Lee et al., 2021b) one can train a unit duration prediction model and use the predicted durations to inflate the sequence and feed the discrete sequence directly to V.

2.3 Unit to Units

Equipped with the models to encode spoken language into discrete unit sequences and convert them back to speech samples, one can conveniently use common NLP architectures to model spoken language. Consider M to be a sequence modeling function that gets as input a discrete unit sequence $\boldsymbol{z}_q$ and outputs another discrete units sequence, denoted as $\hat{\boldsymbol{z}}_q$. Generally, $\hat{\boldsymbol{z}}_q$ can represent different generations, depending on the modeling task. For instance, Lakhotia et al. (2021) and Kharitonov et al. (2021a) set M to be a Transformer and trained a generative spoken language model. Similarly, Kreuk et al. (2021) set M to be a sequence-to-sequence model, hence can cast the emotion conversion problem as a translation task.[1]

[1]Examples are provided at `speechbot.github.io/`.

3 Library Overview

In this section, we present the `textless-lib` library, intending to simplify future research on textless spoken language modeling. Additionally, the proposed package will remove the main barrier of processing and synthesizing speech, which requires domain expertise, for other language researchers (e.g., NLP researchers) who are interested in modeling spoken language, analyzing the learned representations, etc.

To support the above, it is essential to provide the main building blocks described in Section 2, together with pre-trained models, with minimal coupling between them (a list of the supported pre-trained models can be seen on Table 1). This will allow researchers to flexibly use the provided pre-trained building blocks or develop new building blocks and use them anywhere in their pipeline. We decided to exclude both U2U models as well as evaluation metrics from the core functionality of the library as we believe these models should be an example usage. There are plenty of ways to evaluate the overall pipeline (Lakhotia et al., 2021; Dunbar et al., 2019, 2020; Nguyen et al., 2020) and different ways to model the "pseudo-text" units (Shi et al., 2021; Kharitonov et al., 2021a; Polyak et al., 2021; Kreuk et al., 2021; Lee et al., 2021a), hence including them as an integral part of the library will make it overcomplicated.

3.1 Interfaces

The pipeline presented in Figure 1 hints a straightforward way to decouple elements of the library into two principal blocks: (i) encoding speech; and (ii) decoding speech, with the only interdependence being the format of the data in-between (e.g., vocabulary size). Such interfaces enable interesting mix-and-match combinations as well as conducting research on each component independently. We firstly present those two interfaces, then we discuss helpers for dataloading.

```
1  url = "dev-clean"
2  existing_root = "./data"
3  dense_model_name = "hubert-base-ls960"
4  quantizer_name = "kmeans"
5  vocab_size = 100
6
7  encoder = SpeechEncoder.by_name(
8    dense_model_name=dense_model_name,
9    quantizer_model_name=quantizer_name,
10   vocab_size=vocab_size,
11   deduplicate=True,
12 ).cuda()
13
14 quantized_dataset = QuantizedLibriSpeech(
15   root=existing_root, speech_encoder=encoder, url=url)
16
17 datum = quantized_dataset[0]
18 # datum['units'] = tensor([71, 12, 63, ...])
```

Figure 3: `textless-lib` provides an "encoded" view for standard datasets, such as LibriSpeech.

Encoders and Vocoders. We denote the encoders as `SpeechEncoder`. These modules encompass all steps required to represent raw audio as discrete unit sequences (i.e., pseudo-text units and, optionally duration and pitch streams).

`SpeechEncoder` obtains a dense vector representation from a given self-supervised model, discretizes the dense representation into units, extracts pitch, aligns it with the unit streams, and potentially, applies run-length encoding with per-frame pitch averaging. See Fig. 2 for a visual description.

For each sub-model, a user might choose to use a pre-trained model or provide a custom `torch.nn.Module` module instead. An example of the former is demonstrated in lines 7-12 in Figure 3, in which a HuBERT model and a corresponding k-means codebook with a pre-defined K (i.e., vocabulary size) are automatically retrieved.

Conversely, vocoders take as input a discretized sequence and convert it back to the audio domain. As with `SpeechEncoder`, we can retrieve a pre-trained model by setting the expected input specification (model, quantizer, and the size of the codebook), see Figure 4 lines 17-21.

Datasets, Dataloaders, and Preprocessing.

Apart from encoders and vocoders, in the `textless-lib` we provide several components aimed to simplify frequent data loading use-cases. First, we provide a set of standard datasets (e.g., LibriSpeech) wrapped to produce quantized representations (see Fig. 3 lines 14-15). Those datasets are implemented via a `QuantizeDataset` wrapper which can be used to wrap any map-style PyTorch dataset, containing raw waveform data.

```
1  dense_model_name = "hubert-base-ls960"
2  quantizer_name, vocab_size = "kmeans", 100
3  input_file, output_file = "input.wav", "output.wav"
4
5  encoder = SpeechEncoder.by_name(
6    dense_model_name=dense_model_name,
7    quantizer_model_name=quantizer_name,
8    vocab_size=vocab_size,
9    deduplicate=True,
10 ).cuda()
11
12 waveform, sample_rate = torchaudio.load(input_file)
13
14 encoded = encoder(waveform.cuda())
15 units = encoded["units"]  # tensor([71, ...], ...)
16
17 vocoder = TacotronVocoder.by_name(
18   dense_model_name,
19   quantizer_name,
20   vocab_size,
21 ).cuda()
22
23 audio = vocoder(units)
24
25 torchaudio.save(output_file,
26                audio.cpu().float().unsqueeze(0),
27                22_050)
```

Figure 4: Fully functioning code for discrete audio resynthesis. An audio file is loaded, coverted into a sequence of pseudo-units and transformed back into audio with Tacotron2. The model setup code will download required checkpoints and cache them locally.

The `QuantizeDataset` runs an instance of a dense representation model, which can be computationally heavy (e.g., the HuBERT-base model has 7 convolutional layers and 12 Transformer layers). Unfortunately, such heavy preprocessing can starve the training loop. Hence, we provide two possible solutions: (i) as part of the `textless-lib` we provide a way to spread `QuantizeDataset` and `DataLoader` preprocessing workers (each with its copy of a dense model) across multiple GPUs, hence potentially balancing training and preprocessing across different devices; (ii) in cases where on-the-fly preprocessing is not required (e.g., there is no randomized data augmentation (Kharitonov et al., 2021b)), an alternative is to preprocess the entire dataset in advance. `textless-lib` provides a tool for preprocessing arbitrary sets of audio files into a stream of pseudo-unit tokens and, optionally, streams of per-frame tempo and F0 values, aligned to the token stream. The tool uses multi-GPU and multi-node parallelism to speed up the process.

Model	Quantized?	Vocab. size	Accuracy
HuBERT	-	-	0.99
HuBERT	✓	50	0.11
HuBERT	✓	100	0.19
HuBERT	✓	200	0.29
HuBERT	✓	500	0.48
CPC	-	-	0.99
CPC	✓	50	0.19
CPC	✓	100	0.32
CPC	✓	200	0.34
CPC	✓	500	0.40

Table 2: Speaker probing. Test accuracy on predicting speaker based on HuBERT & CPC representations.

3.2 Pre-trained Models

As part of `textless-lib` we provide several pre-trained models that proved to work best in prior work (Lakhotia et al., 2021; Polyak et al., 2021). In future, we will maintain the list of the models to be aligned with state-of-the-art.

Dense representations. We support two dense representation models: (i) HuBERT base-960h model (Hsu et al., 2021) trained on LibriSpeech 960h dataset, with a framerate of 50 Hz; (ii) Contrastive Predictive Coding (CPC) model (Rivière and Dupoux, 2020; Oord et al., 2018) trained on the 6K hours subset from LibriLight (Kahn et al., 2020) with a framerate of 100 Hz. Both models provided the best overall performance according to (Lakhotia et al., 2021; Polyak et al., 2021).

Pitch extraction. Following Polyak et al. (2021) we support F0 extraction using the YAAPT pitch extraction algorithm (Kasi and Zahorian, 2002). We plan to include other F0 extraction models, e.g. CREPE (Kim et al., 2018).

Quantizers. With the `textless-lib` we provide several pre-trained quantization functions for both HuBERT and CPC dense models using a vocabulary sizes $K \in \{50, 100, 200, 500\}$. For the quantization function, we trained a k-means algorithm using the "dev-clean" part in the LibriSpeech dataset (Panayotov et al., 2015).

Pitch normalization. Following Kharitonov et al. (2021a), we applied per-speaker pitch normalization to reduce inter-speaker variability. For single speaker datasets, we do not perform F0 normalization and the span of pitch values is preserved. Under the `textless-lib` we provide two pitch-normalization methods: per-speaker and prefix-based. In the per-speaker normalization, we assume the mean F0 value per speaker is known in advance. While in the prefix-based normalization method a part of the audio is used to calculate the mean pitch. Those two options provide useful trade-offs. In the first case, we need to have a closed set of speakers but have a better precision while in the second we sacrifice quality but gain flexibility.

Model	Vocab. size	**Bitrate**, bit/s	WER
Topline	-	$512 \cdot 10^3$	2.2
HuBERT	50	125.5	24.2
HuBERT	100	167.4	13.5
HuBERT	200	210.6	7.9

Table 3: Bitrate/ASR WER trade-off. Topline corresponds to the original data encoded with 32-bit PCM.

Vocoder. In the initial release of the library, we provide Tacotron2 as a mel-spectrogram estimation module (i.e., the G function) followed by WaveGlow (Prenger et al., 2019) neural vocoder (i.e., the V function) as used by Lakhotia et al. (2021).[2] These operate on deduplicated pseudo-unit streams with vocabulary sizes of 50, 100, and 200. In a follow-up release, we aim to include HiFi-GAN-based vocoders similarly to Polyak et al. (2021); Kharitonov et al. (2021a). We found those to generate better audio quality with higher computational performance. However, as described in Section 2, the main drawback of dropping G and directly feeding the discrete units to V is the need for a unit duration prediction model. We plan to include such models as well in the next release.

4 Examples

Alongside the core functionality of the library, we provide a set of illustrative examples. The goal of these examples is two-fold: (a) to illustrate the usage of particular components of the library, and (b) to serve as a starter code for a particular type of application. For instance, a probing example (Section 4.1) can be adapted for better studying used representations, while discrete resynthesis (Section 4.2) could provide a starter code for an application operating on units (e.g., language modeling or a high-compression speech codec).

4.1 Speaker Probing

A vibrant area of research studies properties of "universal" pre-trained representations, such as GLoVE (Pennington et al., 2014) and BERT (Devlin et al., 2018). Examples span from probing for linguistic properties (Adi et al., 2017b; Ettinger

[2] WaveGlow is used as a part of `TacotronVocoder`. Both Tacotron2 and WaveGlow were trained on LJ speech (Ito and Johnson, 2017).

HE PASSES ABRUPTLY FROM PERSONS OF ABRUPT ACID FROM WHICH HE PROCEEDS ARIGHT BY ...
HE PASSES ABRUPTLY FROM PERSONS AND CHARCOAL EACH ONE OF THE CHARCOAL ...
HE PASSES ABRUPTLY FROM PERSONS FEET AND TRAY TO A CONTENTION OF ASSOCIATION THAT ...

Table 4: Three continuations of the same prompt (in pink), generated by the speech continuation example under different random seeds. Sampled from a language model trained trained on HuBERT-100 units.

et al., 2016; Adi et al., 2017a; Conneau et al., 2018; Hewitt and Manning, 2019) to discovering biases (Bolukbasi et al., 2016; Caliskan et al., 2017).

In contrast, widely used pre-trained representations produced by HuBERT (Hsu et al., 2021) and wav2vec 2.0 (Baevski et al., 2020) are relatively understudied. Few existing works include (van Niekerk et al., 2021; Higy et al., 2021).

We believe our library can provide a convenient tool for research in this area. Hence, as the first example, we include a probing experiment similar to the one proposed in (van Niekerk et al., 2021; Adi et al., 2019). We study whether the extracted representations contain speaker-specific information. In this example, we experiment with quantized and continuous representations provided by CPC and HuBERT. We randomly split LibriSpeech dev-clean utterances into train/test (90%/10%) sets[3] and train a two-layer Transformer for 5 epochs to predict a speaker's anonymized identifier, based on an utterance they produced. From the results reported in Table 2 we see that the continuous representations allow identifying speaker on hold-out utterances. In contrast, the quantization adds some speaker-invariance, justifying its use.

4.2 Speech Resynthesis

The next example is the discrete speech resynthesis, i.e., the speech audio $\rightarrow$ deduplicated units $\rightarrow$ speech audio pipeline. Fig. 4 illustrates how simple its implementation is with `textless-lib`.

The discrete resynthesis operation can be seen as a lossy compression of the speech. Indeed, if a sequence of n units (from a vocabulary $\mathcal{U}$) encodes a speech segment of length l, we straightforwardly obtain a lossy codec with bitrate $\frac{n}{l}\lceil\log_2|\mathcal{U}|\rceil$ bits per second. Further, the token stream itself can be compressed using entropy encoding and, assuming a unigram token model, the compression rate becomes: $-\frac{n}{l}\cdot\sum_{u\in\mathcal{U}}\mathbb{P}(u)\log_2\mathbb{P}(u)$. In Table 3 we report compression rate/word error rate (WER) trade-off achievable with the HuBERT-derived unit systems, as a function of the vocabulary size. WER is calculated using the wav2vec 2.0-based Automatic Speech Recognition (ASR) w.r.t. and uses the ground-truth transcripts. To calculate the compression rate, the unigram token distribution was fitted on the transcript of LibriLight 6K dataset (Rivière and Dupoux, 2020). From Table 3 we observe that discretized HuBERT representations have a strong potential for extreme speech compression (Polyak et al., 2021).[4] Our provided implementation reports the bitrate.

4.3 Speech Continuation

Finally, we include a `textless-lib` re-implementation of the full GSLM speech continuation pipeline (Lakhotia et al., 2021), as depicted in Figure 1. Table 4 presents ASR transcripts of three different continuations of the same prompt, generated using different random seeds. We use a LARGE wav2vec 2.0 model, trained on LibriSpeech-960h with CTC loss. Its decoder uses the standard KenLM 4-gram language model.

5 Discussion and Future Work

We introduced `textless-lib`, a Pytorch library aimed to advance research in textless modeling of spoken language, by simplifying textless processing and synthesizing spoken language. We described the main building blocks used to preprocess, quantize, and synthesize speech. To demonstrate the usability of the library, we provided three usage examples related to (i) representation probing, (ii) speech compression, and (iii) speech continuation. The proposed library greatly simplifies research in the textless spoken language processing, hence we believe it will be a handful not only for speech researchers but to the entire NLP community.

As a future work for `textless-lib` we envision improving performance of the existing building blocks, adding new example tasks (e.g., translation (Lee et al., 2021b) or dialog (Nguyen et al., 2022)), extending the set of provided pre-trained models, and introducing the possibility of training the different components.

[3]We have to create a new split as the standard one has disjoint sets of speakers, making this experiment impossible.

[4]In contrast to our setup, Polyak et al. (2021) worked with non-deduplicated streams, hence obtained different bitrates.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017a. Analysis of sentence embedding models using prediction tasks in natural language processing. *IBM Journal of Research and Development*, 61(4/5):3–1.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017b. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks.

Yossi Adi, Neil Zeghidour, Ronan Collobert, Nicolas Usunier, Vitaliy Liptchinsky, and Gabriel Synnaeve. 2019. To reverse the gradient or not: An empirical comparison of adversarial and multi-task learning in speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3742–3746. IEEE.

Junyi Ao, Rui Wang, Long Zhou, Shujie Liu, Shuo Ren, Yu Wu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, et al. 2021. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. *arXiv preprint arXiv:2110.07205*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357.

Tom B. Brown et al. 2020. Language models are few-shot learners. In *Proc. of NeurIPS*.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2021. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*.

Jan Chorowski, Grzegorz Ciesielski, Jarosław Dzikowski, Adrian Lancucki, Ricard Marxer, Mateusz Opala, Piotr Pusz, Paweł Rychlikowski, and Michał Stypułkowski. 2021. Aligned contrastive predictive coding. *arXiv preprint arXiv:2104.11946*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. *arXiv preprint arXiv:2108.06209*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin et al. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W. Black, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. 2019. The Zero Resource Speech Challenge 2019: TTS without T. In *Proc. INTERSPEECH*, pages 1088–1092.

Ewan Dunbar, Julien Karadayi, Mathieu Bernard, Xuan-Nga Cao, Robin Algayres, Lucas Ondel, Laurent Besacier, Sakriani Sakti, and Emmanuel Dupoux. 2020. The Zero Resource Speech Challenge 2020: Discovering discrete subword and word units. In *Proc. INTERSPEECH*, pages 4831–4835.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Bertrand Higy, Lieke Gelderloos, Afra Alishahi, and Grzegorz Chrupała. 2021. Discrete representations in neural models of spoken language. *arXiv preprint arXiv:2105.05582*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. Libri-light: A benchmark for ASR with limited or no supervision. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673.

Kavita Kasi and Stephen A Zahorian. 2002. Yet another algorithm for pitch tracking. In *2002 ieee international conference on acoustics, speech, and signal processing*, volume 1, pages I–361. IEEE.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu-Anh Nguyen, Morgane Rivière, Abdelrahman Mohamed, Emmanuel Dupoux, et al. 2021a. Text-free prosody-aware generative spoken language modeling. *arXiv preprint arXiv:2109.03264*.

Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux. 2021b. Data augmenting contrastive learning of speech representations in the time domain. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 215–222. IEEE.

Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. 2018. Crepe: A convolutional representation for pitch estimation. In *ICASSP*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Proc. of NeurIPS*.

Felix Kreuk, Adam Polyak, Jade Copet, Eugene Kharitonov, Tu-Anh Nguyen, Morgane Rivière, Wei-Ning Hsu, Abdelrahman Mohamed, Emmanuel Dupoux, and Yossi Adi. 2021. Textless speech emotion conversion using decomposed and discrete representations. *arXiv preprint arXiv:2111.07402*.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*.

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Adelrahman Mohamed, et al. 2021. Generative spoken language modeling from raw audio. *arXiv preprint arXiv:2102.01192*.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, and Wei-Ning Hsu. 2021a. Direct speech-to-speech translation with discrete units. *arXiv preprint arXiv:2107.05604*.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Juan Pino, Jiatao Gu, and Wei-Ning Hsu. 2021b. Textless speech-to-speech translation on real data. *arXiv preprint arXiv:2112.08352*.

Mike Lewis et al. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*.

Yinhan Liu et al. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Max Morrison, Rithesh Kumar, Kundan Kumar, Prem Seetharaman, Aaron Courville, and Yoshua Bengio. 2021. Chunked autoregressive gan for conditional waveform synthesis. *arXiv preprint arXiv:2110.10139*.

Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Evgeny Kharitonov, Alexei Baevski, Ewan Dunbar, and Emmanuel Dupoux. 2020. The Zero Resource Speech Benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. In *Advances in Neural Information Processing Systems (NeurIPS) – Self-Supervised Learning for Speech and Audio Processing Workshop*.

Tu Anh Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Elkahky, Paden Tomasello, Robin Algayres, Benoit Sagot, Abdelrahman Mohamed, and Dupoux Emmanuel. 2022. Generative spoken dialogue language modeling. *arXiv preprint arXiv:2203.16502*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters et al. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *arXiv preprint arXiv:1905.09263*.

Morgane Rivière and Emmanuel Dupoux. 2020. Towards unsupervised learning of speech features in the wild. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 156–163.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

Jing Shi, Xuankai Chang, Tomoki Hayashi, Yen-Ju Lu, Shinji Watanabe, and Bo Xu. 2021. Discretization and re-synthesis: an alternative method to solve the cocktail party problem. *arXiv preprint arXiv:2112.09382*.

Benjamin van Niekerk, Leanne Nortje, Matthew Baas, and Herman Kamper. 2021. Analyzing speaker information in self-supervised models to improve zero-resource speech processing. *arXiv preprint arXiv:2108.00917*.

Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang. 2021. Unispeech: Unified speech representation learning with labeled and unlabeled data. *arXiv preprint arXiv:2101.07597*.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.