On Generative Spoken Language Modeling from Raw Audio

Kushal Lakhotia∗, Eugene Kharitonov∗, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte§, Tu-Anh Nguyen†, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, Emmanuel Dupoux‡

Facebook AI Research
textlessNLP@fb.com

Abstract

We introduce Generative Spoken Language Modeling, the task of learning the acoustic and linguistic characteristics of a language from raw audio (no text, no labels), and a set of metrics to automatically evaluate the learned representations at acoustic and linguistic levels for both encoding and generation. We set up baseline systems consisting of a discrete speech encoder (returning pseudo-text units), a generative language model (trained on pseudo-text), and a speech decoder (generating a waveform from pseudo-text) all trained without supervision and validate the proposed metrics with human evaluation. Across 3 speech encoders (CPC, wav2vec 2.0, HuBERT), we find that the number of discrete units (50, 100, or 200) matters in a task-dependent and encoder-dependent way, and that some combinations approach text-based systems.1

1 Introduction

An open question for AI research is creating systems that learn from natural interactions as infants learn their first language(s): from raw uncurated data, and without access to text or expert labels (Dupoux, 2018). Natural Language Processing (NLP) systems are currently far from this requirement. Even though great progress has been made in reducing or eliminating the need for expert labels through self-supervised training objectives (Brown et al., 2020; Peters et al., 2018; Radford et al., 2019; Devlin et al., 2019; Liu et al., 2019b; Dong et al., 2019; Lewis et al., 2020), the basic units on which these systems are trained are still textual. Young children learn to speak several years before they can read and write, providing a proof of principle that language can be learned without any text. Being able to achieve ‘textless NLP’ would be good news for the majority of the world’s languages, which do not have large textual resources or even a widely used standardized orthography (Swiss German, dialectal Arabic, Igbo, etc.), and which, despite being used by millions of users, have little chance of being served by current text-based technology. It would also be good for ‘high-resource’ languages, where the oral and written forms often mismatch in terms of lexicon and syntax, and where some linguistically relevant signals carried by prosody and intonation are basically absent from text. While text is still the dominant form of language present on the web, a growing amount of audio resources like podcasts, local radios, social audio apps, online video games provide the necessary input data to push NLP to an audio-based future and thereby expand the inclusiveness and expressivity of AI systems.

Is it possible to build an entire dialogue system from audio inputs only? This is a difficult challenge, but breakthroughs in unsupervised representation learning may address part of it. Unsupervised learning techniques applied to speech were shown to learn continuous or discrete representations that capture speaker invariant phonetic content (Versteegh et al., 2016; Dunbar et al., 2020), despite themselves not being phonemic (Schatz et al., 2021). Recent developments in self-supervised learning have shown impressive results as a pretraining technique (van den Oord et al., 2017; Chung et al., 2019; Hsu et al., 2021), to the extent that Automatic Speech Recognition (ASR) on par with the state of the art from two years back can be built with 5000 times less labelled speech (Baevski et al., 2020b), or even no with no labelled speech at all (Baevski et al., 2021). Of course, ASR still assumes access to text to learn a language model (LM) and the mapping...
Figure 1: Setup of the baseline model architecture, tasks, and metrics.

The high level idea (see Figure 1) is that automatically discovered discrete units can be used to encode speech into "pseudo-text" (speech-to-unit, S2u), which is used in turn to train a generative language model (unit-based language model, uLM) and to train a speech synthesizer (unit-to-speech, u2S). This enables learning an LM from scratch without text, and use it to generate speech conditionally or unconditionally, essentially replicating what toddlers achieve before learning to read. Early studies using discrete codes learned from an autoencoder show the feasibility of such an approach, but remain at a level of a demo (van den Oord et al., 2017).

In this paper, we address one major conceptual stumbling block which has, thus far, prevented such early studies from having the transformative impact they could have in language technology: model evaluation. We contend that it will be impossible to make progress in this area beyond demos unless proper evaluation methods enabling system comparison are established.

Evaluation for speech generation is difficult due to the continuous, variable and multi-level nature of the speech waveform, and the necessity both to capture fine-grained acoustic details to generate intelligible audio and to abstract away from them to learn higher level language concepts. Text-based models do not have this problem, since the input is already expressed in terms of mid-level discrete units (characters or words), and are typically evaluated with unsupervised metrics close to the learning objectives like perplexity or log likelihood. Here, such an approach is not directly applicable even if we rely on discrete pseudo-text units, since such metrics would depend in an unknown fashion on their granularity (number, duration, and distribution), making the comparison of models that use different units infeasible.

Conceptually, generative spoken language models can be evaluated at two levels, the acoustic and the language levels, and through two modes of operation, encoding and generation, resulting in $2 \times 2$ tasks (see Table 1 and Figure 1). Acoustic Unit Discovery (encoding at the acoustic level) consists of representing speech in terms of discrete units discarding non-linguistic factors like speaker and noise. Spoken Language Modeling (encoding at the language level) consists of learning the probabilities of language patterns. Speech Resynthesis (generation for acoustic modeling) consists of generating audio from given acoustic units. This boils down to repeating in a voice of choice an input linguistic content encoded with speech units. Speech Generation (generation for language modeling) consists of generating novel and natural speech (conditioned on some prompt or not). Compared to standard text generation, a critical and novel component of the audio variant is clearly the discovery of units since it conditions all the other components. This is why we devote our analyses of model architectures to the unit-to-speech component specifically, and leave it for further work to evaluate how the downstream components can also be optimized for spoken language generation.

The major contributions of this paper are as follows: (1) We introduce two novel evaluation metrics for the generation mode of spoken language modeling at the acoustic and language levels respectively. Our key insight is to use a generic pretrained ASR system to establish model-independent assessments of the intelligibility (acoustic level) and meaningfulness (language level) of the produced outputs. The ASR system converts the generated waveform back to text, enabling us to adapt standard text-based metrics for these two levels. (2) We validate these metrics through comparison with human evaluation. We show a high degree of concordance between human and machine evaluations of intelligibility and meaningfulness of generated audio. (3) We show that these metrics can be predicted by simpler ones geared to evaluate the encoding mode of the spoken LM. Zero-shot metrics borrowed from
Table 1: Tasks and metrics proposed to evaluate encoding/generation quality of models at the acoustic or language levels. Bold fonts highlights the main metric used for each category (Section 3 for details).

<table>
<thead>
<tr>
<th>Level</th>
<th>Encoding</th>
<th>Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Spoken</td>
<td>Speech</td>
</tr>
<tr>
<td>LM</td>
<td>Spot-the-word, Syntax-Acc</td>
<td>AUC-of-VERT/PPX, cont-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLEU, PPX@o-VERT</td>
</tr>
<tr>
<td>Acoustic</td>
<td>Acoustic</td>
<td>Resynthesis</td>
</tr>
<tr>
<td>Unit Disc.</td>
<td>ABX-across, ABX-within</td>
<td>PER-from-ASR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CER,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CER-from-ASR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOS</td>
</tr>
</tbody>
</table>

previous studies in the Zero Resource Speech Challenges (Versteegh et al., 2016; Nguyen et al., 2020) correlate well with their generative counterpart, offering an easier proxy to rapidly iterate on model selection. (4) We systematically study the effect of the type of encoding units by factorially crossing three recent speech-to-unit encoders—CPC, Wave2vec 2.0, and HuBERT—with three codebook sizes for the discrete units: 50, 100, 200. We keep constant the rest of the system built from out-of-the-box components (standard Transformer for the uLM, Tacotron 2 for u2S). We show that both the encoder type and the number of units matter, and that they matter differently depending on the evaluation task. (5) We open source our evaluation tools and models to help reproducibility and comparability with future work.

In Section 3, we introduce the ASR, zero-shot and human evaluation metrics, in Section 4 we present the models, in Section 5, we analyze the results and discuss them in Section 6.

2 Related Work

Unsupervised Speech Representation Learning aims to distill features useful for downstream tasks, such as phone discrimination (Kharitonov et al., 2021; Schneider et al., 2019) and semantic prediction (Lai et al., 2021; Wu et al., 2020), by constructing pretext tasks that can exploit large quantities of unlabeled speech. Pretext tasks in the literature can be roughly divided into two categories: reconstruction and prediction. Reconstruction is often implemented in the form of auto-encoding (Hsu et al., 2017a), where speech is first encoded into a low-dimensional space, and then decoded back to speech. Various constraints can be imposed on the encoded space, such as temporal smoothness (Ebbers et al., 2017; Glarner et al., 2018; Khurana et al., 2019, 2020), discreteness (Ondel et al., 2016; van den Oord et al., 2017), and presence of hierarchy (Lee and Glass, 2012; Hsu et al., 2017b).

Prediction-based approaches which task a model to predict information of unseen speech based on its context, have gained increasing interest recently. Examples of information include spectrograms (Chung et al., 2019; Wang et al., 2020; Chi et al., 2021; Liu et al., 2020; Chung and Glass, 2020; Liu et al., 2020; Ling et al., 2020; Ling and Liu, 2020), cluster indices (Baevski et al., 2019; Hsu et al., 2021), derived signal processing features (Pascual et al., 2019; Ravanelli et al., 2020), and binary labels of whether a candidate is the target unseen spectrogram (van den Oord et al., 2018; Schneider et al., 2019; Baevski et al., 2020a; Kharitonov et al., 2021; Baevski et al., 2020b).

Speech Resynthesis. Recent advancements in neural vocoders enabled generating natural sounding speech and music (Oord et al., 2016; Kumar et al., 2019; Kong et al., 2020). These are often conditioned on the log mel-spectrogram for the generation process. Learning low bit rate speech representations in an unsupervised manner has attracted attention from both the machine learning and the speech communities (Baker et al., 2019; Baldi et al., 2019; Tjandra et al., 2020; Morita and Koda, 2020; Vapnik and Tong, 2020). These representations can later be used for generation without text, which is particularly important for low-resource languages (Dunbar et al., 2019, 2020). van den Oord et al. (2017) proposed a Vector-Quantized Variational Auto-Encoder (VQ-VAE) model to learn discrete speech units, which will be later used for speech synthesis using a WaveNet model. Eloff et al. (2019) suggested a VQ-VAE model followed by a FFTNet vocoder model (Jin et al., 2018). Tjandra et al. (2020) suggested using transformer...
(Vaswani et al., 2017) together with a VQ-VAE model for unsupervised unit discovery, and van Niekerk et al. (2020) combines vector quantization together with contrastive predictive coding for acoustic unit discovery. Another line of work uses representations from an ASR acoustic model that are combined with identity and prosodic information for voice conversion (Polyak et al., 2020b,a, 2021b). In terms of evaluation, the Zero-Resource challenge (Dunbar et al., 2019, 2020; Nguyen et al., 2020) used bitrate together with human evaluation. In this paper we additionally introduce an ASR based evaluation metric.

3 Evaluation Methods

We present two sets of automatic evaluation metrics; the first one assesses the output of generative speech models (ASR metrics, Section 3.1); the second one, the encoded representations (zero-shot probe metrics, Section 3.2). Finally, we present the human evaluations (Section 3.3).

3.1 Generation: ASR Metrics

We present our new evaluation metrics for generation tasks. The first task, speech resynthesis, involves S2u, which encodes input speech into units and u2S, which decodes it back to speech. In this task, we wish to evaluate intelligibility of the resulting speech. The second task, speech generation, involves the full S2u→uLM→u2S pipeline, and we wish to evaluate meaningfulness of the generated speech. Our overall idea is to use ASR to convert the generated speech back to text and then use text-based metrics.

Speech Resynthesis Intelligibility: ASR-PER.

The ideal metric for intelligibility would be to use humans to transcribe the resynthesized speech and compare the text to the original input. An automatic proxy can be obtained by using a state-of-the-art ASR system pretrained on a large corpus of real speech.\(^2\) Our main metric is Phone Error Rate (PER), which only uses an acoustic-model ASR, without fusing with an additional language model (Chorowski and Jaitly, 2016). In preliminary experiments we also experimented with a full ASR with an LM and computed Word Error Rate (WER) and Character Error Rate (CER) to give partial credit. The latter is probably closer to human intelligibility metrics, as humans cannot turn off their lexicon or language model. We also computed such metrics by training a fitted ASR model for each resynthesis model on a specific training corpus. The logic of this last test is that it provides a more direct measure of the information lost in the S2u→u2S pipeline, because it could adapt to systematic errors introduced by the u2S model. Since the scores between these different approaches correlated highly, we only report here the results on the PER for a pretrained ASR model that is the simplest to deploy.

Speech Generation Quality and Diversity: AUC on Perplexity and VERT.

Text generation evaluation typically involves two axes: the quality of the generated text (with automatic metrics like mean perplexity or negative log likelihood computed on a reference large language model) and the diversity (with metrics like self-BLEU;\(^3\) Zhu et al., 2018). Typically, there is a trade-off between these two dimensions based on the temperature hyperparameter used for sampling from the language model, whereby at low temperature, the system outputs good sentences but not varied, and at high temperatures, it outputs varied sentences, but not very good. This results in model comparison being either based on 2D plots with lines representing the trade-off between quality and diversity, or on aggregate metrics like the area under the curve. Preliminary explorations (see Appendix Section 7.2) with our models revealed two problems preventing a straightforward application of such a scoring strategy.

First, we found that for some models, at a low enough temperature, self-BLEU score stopped increasing, but the systems started to repeat more and more words within a sentence (e.g., “the property the property the property”). We therefore introduce a new metric, auto-BLEU, that measures within-sentence diversity. For a single utterance \(u\), auto-BLEU is calculated as the ratio of \(k\)-grams \(s \in NG_k(u)\) that are repeated at least once:

\[
\text{auto-BLEU}(u, k) = \frac{\sum_i \mathbb{1}[s \in (NG_k(u)\setminus s)]}{|NG_k(n)|}
\]

As with BLEU score, to obtain \(n\)-gram auto-BLEU we calculate the geometric mean of \(\text{auto-BLEU}(u, k)\) obtained for \(k \in [1, n]\) and average over the set of generated utterances. By

\(^2\)We use a base wav2vec 2.0 phoneme detection model trained on LibriSpeech-960h with CTC loss from scratch.

\(^3\)Higher self-BLEU scores indicate lower diversity of the produced text.
Figure 2: Comparison of diversity and perplexity of the generated speech. We plot VERT vs. Median perplexity. The blue diamond corresponds to the oracle reference point. It defines two cut-offs on the curve: VERT @oracle-PPX and PPX @oracle-VERT. The green area corresponds to the AUC metric.

We entirely draw on evaluations from the Zero Resource challenge series (Versteegh et al., 2016; Dunbar et al., 2019; Nguyen et al., 2020) for comparability with published work and refer to these challenges for details. These metrics are “zero-shot” because they do not require training any classifier, and are either based on distances over embeddings, or on computing probabilities over entire utterances. When they have hyperparameters, these are selected using a validation set.

For acoustic-level evaluation, we use the between-speaker ABX score to quantify how well-separated phonetic categories are. Briefly, it consists of estimating the probability that two tokens of the same category $A$ (x and a) are closer to one another than a token of $A$ (x) and of $B$ (b). The categories are triphones that only differ in the middle phoneme (like bit and bet) and the score is averaged over all possible such pairs. For the across-speaker ABX, a and b are spoken by the same speaker and x by a different one, requiring feature invariance over a speaker change. We also include the bitrate, which has been used in the TTS-without-T challenges (Dunbar et al., 2019) to quantify the efficiency of the discrete units used to resynthesize speech. It is simply the entropy of the sequence of units divided by the total duration.

For language-level evaluation, we use spot-the-word accuracy from the Zero Resource 2021 Benchmark (Nguyen et al., 2020). It consists of detecting the real word from a pair of short utterances like ‘brick’ vs. ‘blick’, matched for unigram and bigram phoneme frequency to ensure that low-level cues do not make the task trivial. This task can be done by computing the probability (or pseudo-probability) of the utterances from the uLM. The test set (sWUGGY) consists of 5,000 word-pseudoword pairs generated by the Google TTS API, filtered for the word being present in the LibriSpeech 960h training set (Panayotov et al., 2015). The ZR21 benchmark also uses higher level metrics, notably, syntactic (based on the sBLIMP dataset), which we did not use because the baselines were too close to chance.

3.3 Human Evaluation Metrics

As above, we asked humans to evaluate two aspects of speech generation: intelligibility and meaningfulness. Intelligibility was assessed using two metrics: i) Mean Opinion Scores (MOS) in...
which raters were asked to evaluate subjectively how intelligible a given audio sample is; and ii) Character Error Rate (CER) computed from written transcriptions providing an objective intelligibility test. As for meaningfulness, we set up a meaningfulness-MOS (MMOS) in which raters were asked to evaluate how natural (considering both grammar and meaning) a given sample is. For both subjective tests, raters evaluate the samples on a scale of 1–5 with an increment of 1.

For the MMOS, we had to select a temperature to sample from. Preliminary experiments showed that humans preferred lower temperatures (yielding also less diverse outputs). Here, we settled on selecting the temperature on a model-by-model basis by constructing a continuation task: We take the 1,000 shortest utterances from LibriSpeech test-clean that are at least 6 seconds long, and use the first 3 seconds as prompts for the uLM (after transcribing them into pseudo-texts). For each prompt, we generated 10 candidate continuations of the same length (in seconds) as the utterance which we took the prompt from. We varied temperature \(0.3, 0.4, \ldots, 1.4, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 3.0\), and selected the one yielding the maximal BLEU-2 score with the reference sentence (after ASR). These temperatures were typically between the two boundary temperatures described above.

We evaluated 100 samples from each of the evaluated methods while we enforced at least 15 raters for each sample. The CrowdMOS package (Ribeiro et al., 2011) was used for all subjective experiments using the recommended recipes for detecting and discarding inaccurate scores. The recordings for the naturalness test were generated by the LM unconditionally and conditionally from a 3 seconds prompt. Participants were recruited using a crowd-sourcing platform.

4 Proposed Systems

Here, we present our S2u (Section 4.1), uLM (Section 4.2), and u2S (Section 4.3) components.

4.1 Speech-to-Unit Models

We selected 3 recent state-of-the-art unsupervised Encoders, which we used ‘out of the box’: we did not retrain them nor change their hyperparameters. We also included a log Mel filter-bank baseline (80 filters, computed every 10 ms). We then discretized the embeddings using \(k\)-means. We only give a high level description of these models, and refer to the original publications for details.

**CPC.** Contrastive Predictive Coding (van den Oord et al., 2017) as applied to speech consists of two components: an encoder and a predictor. The encoder produces an embedding \(z\) from speech input. The predictor predicts the future states of the encoder based on the past, and the system is trained with a contrastive loss. We use the CPC model from Rivière and Dupoux (2020), which was trained on a “clean” 6k hour sub-sample of the LibriLight dataset (Kahn et al., 2020; Rivière and Dupoux, 2020). We extract a representation from an intermediate layer of the predictor, which provides a 256-dimensional embedding (one per 10ms), as in the original paper.

**wav2vec 2.0.** Similar to CPC, this model uses an encoder and a predictor, which is trained contrastively to distinguish positive and negative samples from discretized and masked segments of the encoder’s output. We use the large variant of pretrained wav2vec 2.0 (Baevski et al., 2020b) trained on 60k hours of LibriLight dataset (Kahn et al., 2020). This model encodes raw audio into frames of 1024-dimensional vectors (one per 20ms). To choose the best layer, we extracted frozen representations of the 10-hour LibriLight subset from every layer of the model and trained a linear classifier with the CTC loss to predict the phonetic version of the text labels. Layer 14 obtained the lowest PER on LS dev-other (a similar approach was done in Baevski et al. [2021], which in this case selected Layer 15).

**HuBERT.** Unlike CPC and wav2vec 2.0 that use a contrastive loss, HuBERT is trained with a masked prediction task similar to BERT (Devlin et al., 2019) but with masked continuous audio signals as inputs. The targets are obtained through unsupervised clustering of raw speech features or learned features from earlier iterations, motivated by DeepCluster (Caron et al., 2018). We use the base 12 transformer-layer model trained for two iterations (Hsu et al., 2021) on 960 hours of LibriSpeech (Panayotov et al., 2015). This model encodes raw audio into frames of 768-dimensional vectors (one per 20 ms) at each layer and we extract those from the \(6^{th}\) layer as in the original paper.

**LogMel.** As a baseline, we consider a Log Mel Filterbank encoder using 80 frequency bands.
Quantization. We use k-means to convert continuous frame representations into discrete representation by training on LibriSpeech clean-100h (Panayotov et al., 2015). We experiment with codebooks that have 50, 100, and 200 units.

4.2 Unit-Language Model
We use the Transformer model as implemented in fairseq (Ott et al., 2019). We use the transformer_lm_big architecture: It has 12 layers, 16 attention heads, embedding size of 1024, FFN size of 4096, and dropout probability of 0.1, and we train it as a causal LM on sequences of pseudo-text units. Each sample contains up to 3,072 units. We use sampling with temperature for generation.

All language models are trained on a ‘‘clean’’ 6k hours sub-sample of LibriLight used in (Rivi`ere and Dupoux, 2020), transcribed with corresponding discrete units. In preliminary experiments, we found that removing sequential repetitions of units improves performance, hence we apply it universally.7 We hypothesize that this simple modification allows us to use Transformer’s limited attention span more efficiently, as in Hsu et al., 2020.

4.3 Unit-To-Speech Model
We adapt the Tacotron-2 model (Shen et al., 2018) such that it takes pseudo-text units as input and outputs a log Mel spectrogram. To enable the model to synthesize arbitrary unit sequences, including those representing incomplete sentences, we introduce two modifications. First, we append a special ‘‘end-of-input’’ (EOI) token to the input sequence, hinting the decoder to predict the ‘‘end-of-output’’ token when attending to this new token. However, this modification alone may not be sufficient, as the decoder could still learn to ignore the EOI token and correlate end-of-output prediction with the learned discrete token that represents silence as most of the speech contains trailing silence. To address this, we train the model using random chunks of aligned unit sequence and spectrogram, and append the EOI token to unit sequence chunks, such that the audio does not always end with silence. We implement chunking in the curriculum learning fashion, where the chunk size gradually grows (starting with 50 frames with an increment of 5 per epoch) to increase the difficulty of the task. For waveform generation, we use the pre-trained flow-based neural vocoder WaveGlow (Prenger et al., 2019). This model outputs the time-domain signal given the log Mel spectrogram as input. All u2S models were trained on LJ Speech (LJ) (Ito and Johnson, 2017).

5 Results
In Figure 3, we report the overall results of our models and our LogMel baseline as a function of the number of quantized units on our main automated and human metrics. More detailed results follow in the following sections, including two character-based toplines: one uses the oracle transcripts for training the LM, the other uses transcripts produced by the pre-trained ASR model.

5.1 Results on the Resynthesis Task
Overall resynthesis results are shown in the bottom middle and right cells of Figure 3 for our main automatic (PER) and human scores (MOS), respectively, averaged across the LS and LJ evaluation sets. We observe that across all models, increasing the number of units uniformly leads to better scores suggesting that the u2S component can take benefit from extra details of the input to produce a more realistic output. HuBERT and CPC seem to be giving the best results, for both humans and models better capturing phonetic information than other models at equivalent bitrates.

More detailed results are in Table 2, separating the scores for the LJ and LS resynthesis, and adding extra automatic metrics (CER) and human metrics (human CER). On PER, we found a domain effect: Resynthesizing input from LJ Speech yields lower PER than from LibriSpeech on all unsupervised models. From the viewpoint of the encoder, LJ Speech is out-of-domain; therefore, one would expect that the units are making more errors than for the trained LibriSpeech. On the other hand, the u2S component has learned from LJ Speech encoded with these units, and might have learned to compensate for these lower quality units. When LibriSpeech is offered as input, the u2S component cannot adapt to this nominally better input and ends up yielding lower quality

7For example, a pseudo-text 10 11 11 11 21 32 32 21 becomes 10 11 21 32 21.
outputs. This observation is worth further exploration, as other metrics like CER (using an LM) and human evaluations only replicated this for the models with the lowest score (like LogMel and wav2vec). The automatic PER and CER scores and the human MOS and CER scores all correlate well with one another across the $4 \times 3$ models and baselines. Within the LJ or LS domain, the Pearson $r$ ranged from .95 to .99; across domains it was less good (from .79 to .96), illustrating again the existence of a domain effect. Not shown here, we reached similar conclusions with our fitted-ASR metrics, but with less good score and correlations. Table 2 also shows the results of the two toplines (original text+TTS and ASR+TTS). Interestingly, our best models come within 3% absolute in PER or CER compared to these toplines, are quite close to them in terms of MOS and even beat them in terms of human CER.

5.2 Results on the Generation Task

The upper mid and right cells of Figure 3 show generation results averaging across the unconditional and conditional conditions, on automatic and human evaluations, respectively. The main result is that there is both an effect of number of units and of system. As for resynthesis, 50 units is always worst, but contrary to resynthesis, 200 units is not always better. Overall, the results on generation are congruent with the idea that speech generation both requires good scores on language modeling and on speech synthesis. The best results for a particular model are then a compromise between the number of units that give both scores to either of these tasks. In terms of systems, the best one here is HuBERT. Regarding human evaluations, they show similar patterns with a clear dispreference for 50 units, and either 100 or 200 being better.

Detailed results are shown in Table 3 with separate statistics for conditional and unconditional generation and additional results with PPX@o-VERT and VERT@o-PPX. As expected, the perplexity metric improved with prompts, but not the diversity score. The human results are congruent with the automatic scores, although they tend to prefer more units, perhaps showing that they cannot fully dissociate their judgment...
Table 2: Results on the resynthesis task for 3 unsupervised models plus one LogMel baseline and 3 unit sizes. Bitrates are in bit/sec, PER are for a pretrained phone recognition model without lexicon and LM, CER are derived from a full ASR model (lower is better). Human MOS (upper is better) and CER (computed from transcription, lower is better) are provided (the 95% confidence interval was on average .32 for MOS and 1.8 for human CER).

of meaning from their judgment of intelligibility. The three metrics correlate well with one another (r between .86 and .99) and correlate with their counterpart across task (prompted vs. unprompted: r between .82 and .99). Human evaluations correlated well with the automatic metrics (AUC: r = .97; PPX: r = .92; VERT: r = .75).

5.3 Results for Zero-shot Probe Metrics

In Table 4, we show the results for zero-shot metrics across the different models and baselines. Overall, the performance depends on the linguistic levels while remaining above chance. While performance is excellent at the acoustic level (6.5% error for the best model on ABX-across), it is intermediate at the lexical level (31.3% error for the best model on spot-the-word). Not shown, the syntactic test is close to chance (42% error for the best model on the sBLIMP test). These values are worse than the ASR-topline (3.1% and 29%, for lexicon and syntax, resp.), showing room for improvement.

The metrics correlate well: The ABX score predicts the lexical score (r = 0.85) and the syntax score (r = 0.71). Across the different models, CPC gets the best units (ABX score) and HuBERT gets the best LM scores. In addition, we see a clear effect of number of units (Figure 3). For wav2vec, the performances on all metrics increase with more units, whereas, for CPC and HuBERT a U-shaped pattern emerges on most metrics, with best scores for units of intermediate sizes. It is interesting that the models with the highest bitrate do not always have the best results. This means that encoding too much acoustic information can be detrimental to linguistic encoding in the uLM. See Appendix Section 7.1 showing that ABX has good correlations with automatic and human metrics (r > .88).

6 Discussion and Conclusion

We introduced Generative Spoken Language Modeling as a new unsupervised task bridging the gap between speech and natural language processing.
and related it conceptually to previously studied unsupervised tasks: Acoustic Unit Discovery, Spoken Language Modeling, Discrete Speech Resynthesis, and Text Generation. We introduced a suite of metrics, baselines, and first results on Librilight that sets the playing field for future work. For comparability, we open source our evaluation stack and the best of our baseline models.

Our main contributions are as follows. (1) We established a set of easy to use automatic ASR-based metrics for model comparison at two critical levels for this task: intelligibility of the speech output and meaningfulness in terms of higher linguistic content. We assessed the first through ASR-based PER and CER metrics; and the second using text-generation-based metrics (AUC for PPX/VERT). (2) We found that these two sets of metrics correlated well with human judgment and (3) that they can be approximated with their inference-mode counterparts, which are faster to compute using zero-shot probe tasks. (4) Applying these metrics to pipeline models based on current speech representation learning models and out-of-the-box LM and TTS components, we found that our basic premise is fulfilled: It is possible to train a language model from quantized units derived from audio and using it to generate new speech. The generated speech is English-sounding, with recognizable phonemes and words and locally acceptable syntax (see transcribed examples in the Appendix and audio snippets here: https://speechbot.github.io/gslm). Our automatic metrics confirm the quality of the representations

<table>
<thead>
<tr>
<th>Systems</th>
<th>Encoder architect. units</th>
<th>Generation based metrics</th>
<th>Human Opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>unconditional prompt</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PPX ↓ VERT ↓ AUC ↓</td>
<td>PPX ↓ VERT ↓ AUC ↓</td>
</tr>
<tr>
<td>Controls</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>oracle text</td>
<td></td>
<td>154.5 19.43 –</td>
<td>154.5 19.43 –</td>
</tr>
<tr>
<td>ASR + LM</td>
<td></td>
<td>178.4 21.31 0.18</td>
<td>162.8 20.49 0.04</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LogMel 50</td>
<td></td>
<td>1588.97 – 1083.76</td>
<td>–</td>
</tr>
<tr>
<td>LogMel 100</td>
<td></td>
<td>1500.11 95.50 510.26</td>
<td>–</td>
</tr>
<tr>
<td>LogMel 200</td>
<td></td>
<td>1539.00 – 584.16</td>
<td>–</td>
</tr>
<tr>
<td>Unsupervised</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPC 50</td>
<td></td>
<td>374.26 46.26 19.68</td>
<td>323.9 39.92 18.44</td>
</tr>
<tr>
<td>CPC 100</td>
<td></td>
<td>349.56 41.79 15.74</td>
<td>294.7 42.93 14.06</td>
</tr>
<tr>
<td>CPC 200</td>
<td></td>
<td>362.84 40.28 16.46</td>
<td>303.5 43.42 26.67</td>
</tr>
<tr>
<td>HuBERT-L6 50</td>
<td></td>
<td>376.33 43.06 19.27</td>
<td>339.8 45.85 21.03</td>
</tr>
<tr>
<td>HuBERT-L6 100</td>
<td></td>
<td><strong>273.86</strong> <strong>31.36</strong> <strong>5.54</strong></td>
<td><strong>251.2</strong> <strong>33.67</strong> <strong>5.88</strong></td>
</tr>
<tr>
<td>HuBERT-L6 200</td>
<td></td>
<td>289.36 33.04 7.49</td>
<td>262.4 34.30 6.13</td>
</tr>
<tr>
<td>wav2vec-L14 50</td>
<td></td>
<td>936.97 – 307.91</td>
<td>1106.3 – 330.8</td>
</tr>
<tr>
<td>wav2vec-L14 100</td>
<td></td>
<td>948.96 79.51 208.38</td>
<td>775.1 – 205.7</td>
</tr>
<tr>
<td>wav2vec-L14 200</td>
<td></td>
<td>538.56 61.06 61.48</td>
<td>585.8 – 91.07</td>
</tr>
</tbody>
</table>

Table 3: Results on the generation task for three unsupervised models plus the LogMel baseline and 3 unit sizes. PPX@-o-VERT and VERT@o-PPX are reported as PPX and VERT. ‘-’ : missing or non calculable results. Human MMOS are also provided (the 95% confidence interval was on average .29 for uncond. and .61 for cond.).

<table>
<thead>
<tr>
<th>System</th>
<th>Metrics</th>
<th>S2u</th>
<th>uLM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Nb</td>
<td>ABX</td>
</tr>
<tr>
<td>Toplines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASR+LM</td>
<td>–</td>
<td>–</td>
<td>3.12</td>
</tr>
<tr>
<td>Baselines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LogMel 50</td>
<td>23.95</td>
<td>35.86</td>
<td>48.52</td>
</tr>
<tr>
<td>LogMel 100</td>
<td>24.33</td>
<td>37.86</td>
<td>48.12</td>
</tr>
<tr>
<td>LogMel 200</td>
<td>25.71</td>
<td>39.65</td>
<td>49.62</td>
</tr>
<tr>
<td>Unsupervised</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPC 50</td>
<td>5.50</td>
<td>7.20</td>
<td>32.18</td>
</tr>
<tr>
<td>CPC 100</td>
<td>5.09</td>
<td>6.55</td>
<td>32.18</td>
</tr>
<tr>
<td>CPC 200</td>
<td>5.18</td>
<td>6.83</td>
<td>37.40</td>
</tr>
<tr>
<td>HuBERT-L6 50</td>
<td>7.37</td>
<td>8.61</td>
<td>32.88</td>
</tr>
<tr>
<td>HuBERT-L6 100</td>
<td>6.00</td>
<td>7.41</td>
<td>31.30</td>
</tr>
<tr>
<td>HuBERT-L6 200</td>
<td>5.99</td>
<td>7.31</td>
<td>36.52</td>
</tr>
<tr>
<td>wav2vec-L14 50</td>
<td>22.30</td>
<td>24.56</td>
<td>51.92</td>
</tr>
<tr>
<td>wav2vec-L14 100</td>
<td>18.16</td>
<td>20.44</td>
<td>50.24</td>
</tr>
<tr>
<td>wav2vec-L14 200</td>
<td>16.59</td>
<td>18.69</td>
<td>44.68</td>
</tr>
</tbody>
</table>

Table 4: Results for zero-shot probe metrics for 3 unsupervised models plus one LogMel baseline and 3 unit sizes. ABX within and across speakers, spot-the-word, and acceptability judgments are error rates (lower is better); chance is 50%.
and outputs at the acoustic/phonetic level, but show that improvements are needed at the language level. It is to be expected that performance will increase with larger training sets beyond our 6k hours, as has been noted in the case of text.

(5) We also uncovered specific issues regarding the number of quantized units. For speech resynthesis, the optimum number of units was always 200 by a large margin, reflecting the well known bitrate/intelligibility trade-off (Dunbar et al., 2019). However, for language modeling, this was not necessarily the case, as the more detailed acoustic information may introduce too numerous phonetic details that have no impact at the level of lexical and syntactic representations.

(6) Finally, we found that the choice of units also affected the temperature parameter which is used to control the trade-off between quality and diversity in text-based language model. To address this effect, we proposed a method to normalize the temperature by using an oracle text to build perplexity and diversity anchor points.

Obviously, this is only a first step towards building textless NLP applications that could be applied to any language, even low resource ones. To reach this long term goal, three important challenges need to be addressed.

First, even though we did compare three different encoders and obtained different results, we cannot conclude that one encoder is definitely superior to the others. Our point here was merely to use previously published pretrained encoders, and study systematically the effect of number of units on these encoders. A fuller study including a wider set of encoders and a proper hyperparameter search (including the selection of the embedding layer and the clustering algorithm) would be needed in order to determine which of them is most appropriate for speech generation.

Second, it is to be expected that to further improve generation results, more needs to be done than applying this pipeline to larger training sets. Contrary to text, speech unfolds through time and varies continuously in phonetic space. Speech also contains multilayered representations (phonetic, prosodic, speaker identity, emotions, background noise, etc.). However, both our TTS and our LM were out-of-the-box systems typically used for text applications. More work is needed to adapt these architectures to the richness and variability of the speech signal (see Polyak et al., 2021a, for first steps towards integrating prosody into discrete units). The metrics and baselines we introduced here provide landmarks against which we will measure future progress.

Third, the automatic metrics that we defined here depend on textual resources to build the evaluation ASR and LM models, and on linguistic resources to build the zero-shot metrics. How could this ever be applied to low-resource languages? Note that the linguistic resources we require are used only for model selection, not model training. Our metrics allow for fast iterations in architecture and hyperparameter search, but the overall algorithm is totally unsupervised. Therefore, an important next step is to extend this work to other languages, in order to find a common architecture/hyperparameter set that gives good results in held-out languages (high or low resource). The hope is that once good learning models are tuned using a diverse sample of high resource languages, the same models could be deployed in languages where no such resources are available, and work in a purely unsupervised fashion.

Acknowledgments

We thank Michael Auli and Alexis Conneau for their useful input on wav2vec, and Lior Wolf, Pierre Emmanuel Mazaré, and Gargi Gosh for their support for this project. We would also like to thank the reviewers and editors for their thorough review, and constructive feedback.

References


Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. wav2vec


7 Appendix

7.1 Zero-shot Metrics Correlation Results

In Figure A1, we present the Pearson correlations between the zero-shot metrics and the human and automatic metrics on downstream tasks. The fact that the ABX metric correlates well with these downstream metrics makes it a useful proxy metric for preliminary model and unit size selection, as it is much less costly than generating TTS output and running human or ASR evaluations.

7.2 Effect of Temperature on Outputs

In this section, we describe preliminary experiments we conducted to test the effects of temperature on the generated outputs. As shown in Table A1, the temperature defined qualitatively 4 operating zones. With the lowest temperature, we get repetitive outputs, where the system keeps repeating the same few words. At a slightly higher temperature, the system outputs complete sentences, but they are sampled from a narrow set of topics. At the highest temperature, the system utters an unstructured bag of words. In the mid-temperature range, we observe relatively coherent and varied outputs. This is the range we want to select for our systems. As described in Figure 2, the lowest bound was set by using the oracle PPX (temperature range between 0.2 and 0.65, across unsupervised models) and the highest bound by using the oracle VERT (temperature range between 1.1 to 1.4). In Figure A2 we present human opinion results for samples from these two temperatures, plus an extra mean temperature falling in between. Humans typically preferred the lower temperature.

In Figure A3, we illustrate the continuation method for selecting a single temperature for human meaningfulness judgments in a model-neutral way, as explained in Section 3.3. It consists of generating possible continuations of each prompt and computing the BLEU-2 score with oracle continuation. The temp@cont temperature is defined as the temperature maximizing this score. Computing these estimates with 10 continuations gave continuation temperatures varying between 0.5 and 0.9 across models and unit sizes. These are the temperatures we used for the MMOS results reported in the main paper.

---

8We used NLTK to compute BLEU (Bird et al., 2009).
Figure A1: Patterns of correlations between the zero-shot metrics and the automatic and human metrics. Color scale indicates strength of the Pearson correlation coefficient (we used negative MOS and MMOS to enforce less is better for all metrics).

### Table A1: Unconditional uLM (trained on CPC-100 units) samples, transcribed by ASR, at different temperatures.

<table>
<thead>
<tr>
<th>Temp</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very low temperature samples (stuttering zone)</strong></td>
<td>0.3 the property by James Resell red for liberata or by Jason Downy the property by Jason Downy the property the property the property the property and to take in another path and to take in another path and to take in another path and to take in another path and to take in another path and take in a</td>
</tr>
<tr>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td><strong>Low temperature samples (obsessive zone)</strong></td>
<td>0.7 chapter nineteen of the life of the upper part of the ocean this is all bravos recording only bravos recordings are in the public domain i for more information or to volunteer</td>
</tr>
<tr>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td><strong>Mid temperature samples</strong></td>
<td>1.0 but it is attendant from the people to defend himself from this information pride of the potential in criminal activity a curiosity and impetuosity of the world a war soon acquired</td>
</tr>
<tr>
<td>1.0</td>
<td>finally we ought to have a strong plan a without positively the best type of the public with which we ascend it or extend it our business and as we are a persons of the most strong designs and other affairs of the case we</td>
</tr>
<tr>
<td><strong>High temperature samples (babble zone)</strong></td>
<td>1.5 action of pure blue he said at once a licking streamy at her warm spot of half performed note was a raging oath let it as bir of amole in mood strolling er crass</td>
</tr>
<tr>
<td>1.5</td>
<td>at the swing here as to motions out of the events not time and abe he was any stump headed and flow any he’s the kiln are tama why do ye take the floor</td>
</tr>
</tbody>
</table>
Figure A2: MMOS for unconditional (no prompt) and conditional generated speech sampled at the three reference temperatures (oracle VERT, oracle PPX, and average temperature) (preliminary experiments).

Figure A3: Method for selecting the continuation temperature for MMOS judgments.