Quantizer robustness training in Generative Spoken Language Modeling

March 28, 2025

Raphaël Bernas
ENSTA-MVA
raphael.bernas@ensta.fr

Maxime Corlay
ENSTA-MVA
maxime.corlay@ensta.fr

Adrien Letellier
ENSAE-MVA
adrien.letellier@ensae.fr

Emilie Zheng
ENS-MVA
emilie.zheng@ens.psl.eu

Abstract

A Speech-to-Speech (S2S) system is often divided into three main components : Speech-to-Unit, Unit Language Model (LM), and Unit-to-Speech. In the paper "Augmentation Invariant Discrete Representation for Generative Spoken Language Modeling", the author primarily focuses on the Speech-to-Unit component. This component consists of an encoder and a quantizer model, which takes a traditional continuous sound signal as input and produces a discrete output. Several methods have been proposed for generating these discrete outputs, but most have shown robustness issues. To address this, the authors propose a processing method for the quantizer to improve the quality of the outputs. Unfortunately, their approach has limitations. In their paper, they discuss the architectural choices for those quantizers, but the validity of the metric they use itself is also questionable, particularly in a Speech-to-Speech context, where measuring word distance accurately is challenging. In this work, we are trying to dive further in order to better understand this.

1. Introduction

Our work can be found in the notebook here : https://github.com/Raphael-Bernas/NLP-project

In "Augmentation Invariant Discrete Representation for Generative Spoken Language Modeling" by Itai Gat, Felix Kreuk et al. [2] The authors focus on the question of robustness for the quantizer function of a S2S model. They mainly focus on improving performance when faced with augmentation of the audio input. Let us dive a bit deeper into the theory behind: we denote by f_θ our S2S model. The model takes as input a signal in $s : \mathbb{R}_+ \mapsto \mathbb{R}$ discretized into a sequence $(s(t_0), s(t_0 + dt), \cdots, s(t_0 + n \times dt), \cdots)$ with dt called the "sampling rate". State-of-the-art S2S models often have the following autoencoder architecture : $f_\theta = \cdots \circ \Phi_{dec} \circ \cdots \circ E \circ \Phi_{enc}$ where E is called a **quantizer**. A quantizer maps an embedding into a discrete form (used to tokenize the signal here). Models often rely on k-means, an unsupervised method to cluster points for the quantizer. In [2], the idea is to replace this quantizer by a trained multilayer perceptron (MLP). Thus, they introduce a training process to improve robustness. We denote by g an augmentation function taking a signal s and returning a perturbated signal $g(s)$. For a given augmentation function g (may be a random selection of augmentation), a proper quantizer E_0 (for their first iteration of the method, they use k-means), a signals dataset $\mathcal{D}_S$ and a new quantizer E_1^θ (with θ, its parameters), they solve :

$$\min_\theta -\frac{1}{|\mathcal{D}_S|} \sum_{s \in \mathcal{D}_S} log\left(p(E_0 \circ \Phi_{enc}(s) | E_1^\theta \circ \Phi_{enc} \circ g(s))\right)$$

To evaluate their method's results the authors introduce a metric to "measure" robustness. We begin our work with a discussion about this metric.

2. Discussion about Unit Edit Distance

Limitations The authors propose a robustness metric based on the Levenshtein distance between the outputs of normal and perturbed inputs. This metric allows them to compare sequences of different lengths. However, a weakness of the Levenshtein distance is that it is a quantized-level metric that does not take into account the similarity between speech tokens. The quantized units that are fed into the unit language model are not supposed to be orthogonal but rather they are in an embedding space. Hence, not all token differences are equally significant. For example, if the tokens mimic phonemes, the Levenshtein-based metric would have the same cost to substitute tokens that are very

different, such as /k/ and /θ/, and ones that are closer, such as /o/ and /$ɔ$/.

Clustering of k-means might exacerbate this problem. As the clustering is not density-based, close speech units may be assigned to different tokens, making the Levenshtein distance too rigid. Finally, the fact that the metric increases monotonically with the number of units (Figure 2 of the paper) reveals another problematic aspect of the UED, namely that it artificially increases with the size of the vocabulary.

Extensions of the Levenshtein distance were developed to address these limitations, such as pointwise mutual information (PMI) methods to assign different costs depending on the tokens that we want to substitute ([4], [5]). However, such approaches are less straightforward to implement since first one is required to learn the costs for every substitution from the data.

Dynamic Time Warping Unlike the Levenshtein approaches, Dynamic Time Warping (DTW) can compare sequences with continuous and multidimensional values. In a setting where the tokens are in an embedding space, DTW on the hidden representations of the encoder was shown to be well correlated with human evaluation to measure speech similarity [1].

In our setting, we could apply DTW using different representations, namely the embeddings from the unit language model or the embeddings from the unit-to-speech model. Here, since we focus on the speech-to-unit, an alternative is to use the centroids of the k-means clustering.

Model	ABX Score
HuBERT, E_1, vocab size of 50	0.12
HuBERT, E_2, vocab size of 50	0.12

Table 1. ABX error rate, DTW. The scores are lower than for the UED which gave scores of 0.37 for both models. However, we are still concluding that the training of a second quantizer does not really improve performance.

While DTW has theoretical advantages, practical implementation remains a challenge. The main problem is the computational cost given the high dimensionality of the hidden representations. Further experiments could be to try DTW after a dimensionality reduction technique. The choice of DTW or Levenshtein distance also depends on the clustering approach. For k-means clustering with poorly separated clusters, DTW might be more appropriate, but if we had a well-separated density-based clustering the UED could be enough.

3. Practical Experiments

3.1. General setup

We followed the guidelines of the paper [2], and trained a quantizer E_1 on top of E_0 (k-means trained on top of HuBERT) using the CTC loss. Unless stated otherwise, the model of the quantizer E_1 is a single layer MLP (simplified compared to the original paper, because the dataset is way smaller). In section 2.2, we investigate the effect of the iterative method, that is training another quantizer E_2 on top of E_1 using the same pipeline used for training E_1 with E_0. In the appendix, we provide results of the same experiments conducted on different encoders (wav2vec2 and WavLM).

3.2. Robustness to different augmentations

We propose to study the robustness of the quantizer E_1 to the following augmentations that alter the sound, but do not change the content of the audio.

We kept these transformations that were already proposed by the authors.

- Pitch shifting : We performed upsampling (resp. downsampling) on the original signal, effectively decreasing of (resp. increasing) the frequencies of the signal by a given number of semi tones.
- Noise adding : We add a gaussian noise to the original signal, with fixed mean and variance.

We also implemented these new augmentations.

- Background noise : We added some bips to the audio, simulating some backgroung noise.
- Echo : We simulate a closed space by adding some echo to the audio.
- Missing samples : We randomly dropped out some measurements in the audio, setting them to 0, simulating some imperfect sound captor.
- Volume change : We changed the volume of the audio, since the content should not depend on it.
- Signal clipping : We clip the signal to keep it within some given range.
- Frequency clipping : We applied some frequency filterings, taking out all the frequencies that are not in a given range.

We trained the quantizer E_1 using all the transformations listed above, that were uniformly applied on the training dataset (more details on the training in the appendix).

We used two metrics to evaluate the robustness of $E1$: the deduplicated Unit Edit Distance (UED) and the ABX metric.

In figure 2, we show our results, comparing E_0 with E_1 using both the UED score on test samples that were not in the training set. We also show the ABX scores on the quantizers $E1$ and $E2$.

We get significantly worse results than the original paper, but this could be caused by the quantity of data : we trained

our quantizer on a dataset much smaller than the one used in the paper.

3.3. Iterative training

In this section, we consider another quantizer $E2$ trained on top of $E1$ just as $E1$ was trained on top of $E0$ (with the same hyperparameters and number of epochs).

We give our results of the scores on different augmentations in figure 2. We remark that most of the times, $E2$ outperforms $E1$ based on the UED scores.

There is some improvement compared to $E1$ when considering the UED distance. However, since $E1$ is not better than $E0$, we can not expect $E2$ to be better than $E0$.

This metric also shows slight improvement of $E2$ over $E1$.

We also computed the ABX metric of the 2 quantizers $E1$ and $E2$, and we do not get significant improvement.

3.4. Model of the quantizer

	UED E0	UED E1	UED E2
Missing samples	1.57	2.29	2.23
Volume change	0.42	0.40	0.29
Noise adding	4.02	5.26	5.12
Pitch shifting	1.85	3.67	2.70
ABX Evaluation			
E1		E2	
0.39		0.38	

Table 2. UED scores for the quantizers E0 (kmeans), E1, and E2 trained on Wav2vec2, along with ABX evaluation results.

3.5. Experiments : Discussion & Conclusion

While the numerical results show improvement of the robustness of the quantizer $E1$ (or $E2$) over $E0$, it remains unclear whether this robustness is relevant after going through a unit-to-speech model.

4. Testing new model

We used a larger encoder (with 1024 outputs instead of 768) to see the impact of using a different model. We used hubert-large-ls960-ft instead of using hubert-base-ls960. The iterative method is the same as in the previous section. We give our results of the scores on different augmentations in table 3.

We observed that the results change a lot from one run to another. On figures 1 and 2, we show the training loss.

We obtained a decreasing loss. For the training E1-E2, we obtained a negative ctc loss, which shows the great instability of the method. The great instability is may be due to the little number of inputs of the dataset we use. 50 samples is very few to learn.

	UED E0	UED E1	UED E2
Noise adding	1.72	2.59	4.35
Time stretch	2.48	3.10	4.55
Pitch shifting	2.36	3.34	2.1
Clipping	5.28	6.06	4.05
ABX Evaluation			
E1		E2	
0.32		0.33	

Table 3. UED scores for the quantizers E0 (kmeans), E1 and E2, and ABX scores for E1 and E2 trained on HuBERT Large.

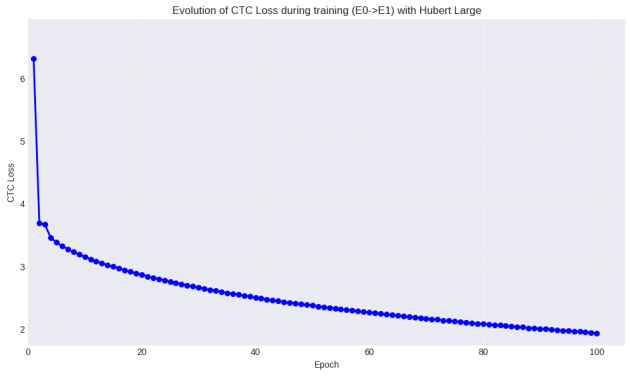

Figure 1. CTC Loss E0-E1

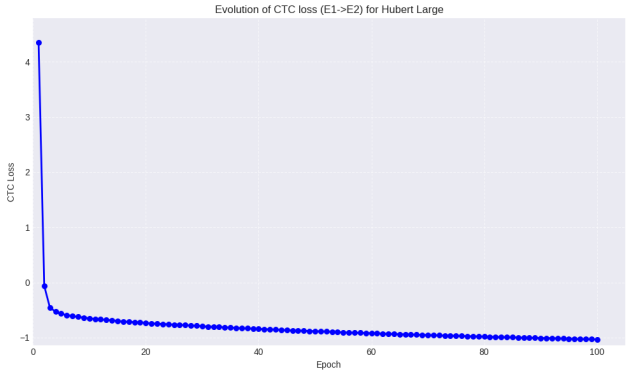

Figure 2. CTC Loss E1-E2

5. Improving quantizer

In [2], the authors fixed a MLP structure for their quantizer. This choice allows them to iterate their training thanks to a low computation cost. But we wonder whether such a choice is optimal. Indeed, in the paper, it does not appear clearly that iterative training yields far better results than the basic training. One could even remark that for some task, the $E1$ quantizer outperformed its iterative "children". Thus, we propose here a study which compares a MLP against an attention based quantizer.

Why use attention ? There are several reasons for this

choice. A first intuition is that we expect that past and future parts of the signal to impact the decision made by the model for tokenization. It is important to remark that when one tries to identify a phoneme from another speech, they will try to predict the phoneme based on the precedent one but also on the next (that is why sometimes you could comprehend a word without hearing well all phonemes). The second reason is more down-to-earth, indeed, Transformers [3] represent a highly developed class of pytorch. But in the interest of honesty, we will also train an MLP with as many parameters as our Transformer quantizer. Let us quickly recall the attention formula from Transformers [3]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

where Q is the query matrix, K the key matrix and V the value matrix. The main idea from the softmax used is that it will weight each value in order for the model to takes a decision based on all data it sees and learn how to discriminate which part of the signal is important. We hope that this method will allow the model to makes proper guess of how to operate tokenization. In Table 4 we detail the results obtained. We can conclude from those results that their choice of MLP architecture has shown better results in most cases than Transformers. Unfortunately, we cannot tell whether it is due to the low epochs training we have performed or due to the underperformance of Transformers here.

Augmentation	**E0**	**E1B**	**E1T**
gaussian_noise	3.03896	3.36652	5.24032
time_stretch	2.00196	4.45919	4.60026
pitch_shift	1.84524	3.98918	4.23216
clipping	6.24134	3.70988	6.28224
lowpass	6.07134	4.04882	5.63811
bandpass	7.37028	4.23944	5.90323
highpass	2.22209	3.02597	4.72911
little_bips	5.21451	3.99738	6.18291
big_bips	0.76819	2.41804	2.75212
echo	3.10572	3.33085	4.97011
corruption	0.92529	2.39520	2.72410
volume_change	0.00000	2.21813	0.00000

ABX Evaluation	
E1B	**E1T**
0.37	0.33

Table 4. UED Scores for Boosted MLP Quantizer (E1B) and Transformer Quantizer (E1T) across different augmentations with ABX scores too.

6. Conclusion

The paper presents an approach to improving robustness in Speech-to-Unit models by introducing an Augmentation-Invariant Discrete Representation. The results demonstrate that their proposed quantizer E2, when trained iteratively on top of E1, shows improvement in robustness against various audio augmentations compared to the baseline k-means quantizer (E0). However, while the numerical results show improvement in the robustness of quantizer E1 (or E2) over E0, it remains unclear whether this robustness is relevant after going through a unit-to-speech model.

The variations in UED scores across different augmentations suggest that the approach may be more effective for certain types of audio distortions than others. Moreover, the results obtained depend a lot on the run.

Changing the encoder model size significantly impacts results across different augmentations. Additionally, the limited dataset size (50 samples) may affect the reliability of the results.

References

[1] Martijn Bartelds, Wietse de Vries, Faraz Sanal, Caitlin Richter, Mark Liberman, and Martijn Wieling. Neural representations for modeling variation in speech. *Journal of Phonetics*, 92:101137, 2022. 2

[2] Itai Gat, Felix Kreuk, Tu Anh Nguyen, Ann Lee, Jade Copet, Gabriel Synnaeve, Emmanuel Dupoux, and Yossi Adi. Augmentation invariant discrete representation for generative spoken language modeling, 2023. 1, 2, 3

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 4

[4] Martijn Wieling, Jelena Prokić, and John Nerbonne. Evaluating the pairwise string alignment of pronunciations. In *Proceedings of the EACL 2009 Workshop on Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education (LaTeCH–SHELT&R 2009)*, pages 26–34, 2009. 2

[5] Martijn Wieling, Jelke Bloem, Kaitlin Mignella, Mona Timmermeister, and John Nerbonne. Measuring foreign accent strength in english: Validating levenshtein distance as a measure. *Language Dynamics and Change*, 4(2):253–269, 2014. 2

7. Appendix A : Some more results

	UED E0	UED E1	UED E2
Missing samples	1.57	2.16	2.29
Volume change	0.42	0.40	0.29
Noise adding	4.02	5.26	5.12
Pitch shifting	1.85	3.67	2.70

ABX Evaluation	
E1	E2
0.37	0.37

Table 5. UED scores for the quantizers E0 (kmeans), E1 and E2, and ABX scores for E1 and E2 trained on HuBERT.