Actes de JEP-TALN-RECITAL 2024
Toulouse, France, 8 au 12 juillet 2024

Implémentation ouverte et étude de BEST-RQ pour le traitement de la parole

Ryan Whetten[1] Titouan Parcollet[2] Marco Dinarelli[3] Yannick Estève[1]
(1) Laboratoire Informatique d'Avignon, Avignon Université, France
(2) Samsung AI Center, Cambridge, United Kingdom
(3) Univervisté Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000, Grenoble, France

RÉSUMÉ

L'apprentissage auto-supervisé (SSL) a fait ses preuves pour le traitement automatique de la parole mais est généralement très consommateur de données, de mémoire et de ressources matérielles. L'approche BEST-RQ (BERT-based Speech pre-Training with Random-projection Quantizer) est une approche SSL performante en reconnaissance automatique de la parole (RAP), plus efficiente que wav2vec 2.0. L'article original de Google qui introduit BEST-RQ manque de détails, comme le nombre d'heures de GPU/TPU utilisées pour le pré-entraînement et il n'existe pas d'implémentation open-source facile à utiliser. De plus, BEST-RQ n'a pas été évalué sur d'autres tâches que la RAP et la traduction de la parole. Dans cet article, nous décrivons notre implémentation open-source de BEST-RQ et réalisons une première étude en le comparant à wav2vec 2.0 sur quatre tâches. Nous montrons que BERT-RQ peut atteindre des performances similaires à celles de wav2vec 2.0 tout en réduisant le temps d'apprentissage d'un facteur supérieur à deux. [1]

ABSTRACT

Open Implementation and Study of BEST-RQ for Speech Processing

Self-Supervised Learning (SSL) has proven to be useful in various speech tasks. However, these methods are generally very resource demanding. BERT-based Speech pre-Training with Random-projection Quantizer (BEST-RQ), is an SSL method that has shown great performance on Automatic Speech Recognition (ASR) while being simpler than other SSL methods, such as wav2vec 2.0. Despite BEST-RQ's great performance, details are lacking in the original paper, such as the amount of GPU/TPU hours used in pre-training, and there is no official easy-to-use open-source implementation. Furthermore, BEST-RQ has not been evaluated on other downstream tasks aside from ASR and speech translation. In this work, we describe a re-implementation of BEST-RQ and perform a preliminary study with a comparison to wav2vec 2.0 on four downstream tasks. We discuss the details of our implementation. We show BEST-RQ can achieve similar downstream performance as wav2vec 2.0 while decreasing training time by over a factor of two.

MOTS-CLÉS : Apprentissage auto-supervisé, reconnaissance de la parole, reconnaissance du locuteur, repérage de mots-clés.

KEYWORDS: Self-supervised learning, speech recognition, speaker recognition, keyword spotting.

1. Cet article est une traduction de Whetten *et al.* (2024) publié au workshop ICASSP SASB 2024 sur l'apprentissage auto-supervisé.

1 Introduction

L'apprentissage auto-supervisé (SSL) est une technique d'apprentissage dans lequel les étiquettes à prédire sont extraites des étiquettes des données d'entrée elles-mêmes. Le SSL peut ainsi tirer parti de grandes quantités de données non étiquetées lors d'une phase de pré-apprentissage, et utiliser une quantité réduite de données déjà étiquetées pour obtenir des résultats très impressionnants sur une grande variété de tâches (Mohamed *et al.*, 2022). Dans le domaine du traitement automatique de la parole, le SSL a permis d'obtenir des résultats à l'état de l'art dans des tâches telles que la reconnaissance automatique de la parole (ASR), la reconnaissance automatique des émotions (ER), la vérification automatique du locuteur (ASV) et la compréhension du langage parlé (SLU) (Mohamed *et al.*, 2022; Yang *et al.*, 2021)

Cependant, le pré-entraînement SSL est très coûteux en termes de données, de mémoire et de calcul. Par exemple, les auteurs de wav2vec 2.0 ont déclaré avoir utilisé environ 2 400 heures de GPU V100 et une taille de lot (*batch*) de 1,6 heure uniquement pour le modèle de base (Baevski *et al.*, 2020). Pour des ensembles de données très volumineux comme pour *LeBenchmark*, les auteurs ont déclaré avoir utilisé 54 600 heures de GPU A100 pour leur modèle extra-large (Parcollet *et al.*, 2023a).

Malgré les efforts déployés pour améliorer l'efficacité d'autres modèles SSL largement utilisés pour la parole, tels que HuBERT et data2vec (Chen *et al.*, 2023; Baevski *et al.*, 2023), le processus reste gourmand en ressources.

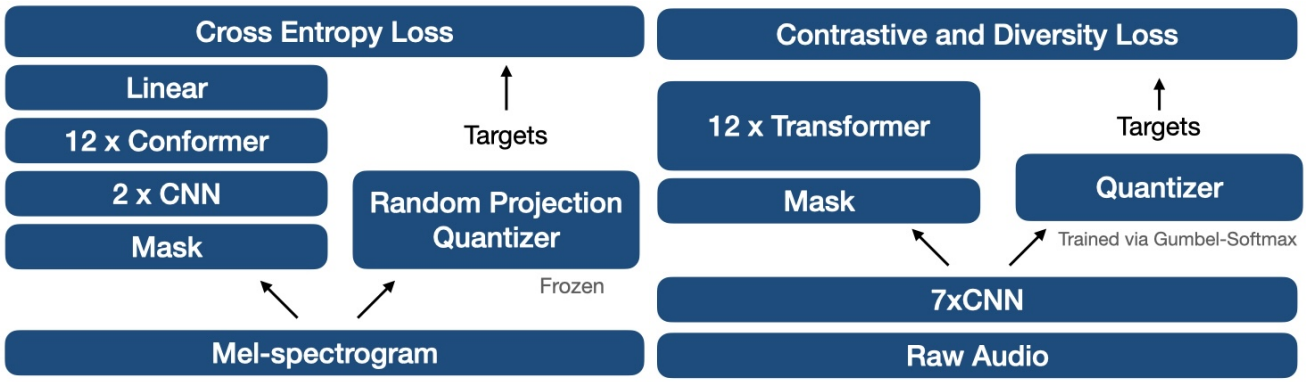
L'une des raisons de ce coût élevé est liée aux extracteurs de caractéristiques acoustiques qui sont généralement mis en œuvre sous la forme d'une série de couches de réseaux neuronaux convolutifs (CNN). Des études récentes ont montré qu'ils peuvent être remplacés par des solutions plus efficaces sans perte de performances (Parcollet *et al.*, 2023b).

Un modèle récent, BEST-RQ (*BERT-based Speech pre-Training with Random-projection Quantizer*) (Chiu *et al.*, 2022), réduit ce coût en réintroduisant l'emploi des banques de filtres Mel à la place de couches convolutives qui doivent être apprises. Grâce à cela et à d'autres simplifications (voir la section 2.1), BEST-RQ semble être l'une des méthodes SSL les plus efficaces proposées jusqu'à présent, tout en conservant des performances très compétitives pour la reconnaissance automatique de la parole.

Actuellement, il n'existe aucune implémentation officielle sous licence libre de BEST-RQ, ce qui limite l'accès à la communauté d'une méthode d'apprentissage SSL très efficace. De plus, les performances de BEST-RQ n'ont été étudiées que pour deux tâches, la reconnaissance automatique de la parole et la traduction vocale (Zhang *et al.*, 2023). Notre objectif est de combler ces lacunes.

Dans cet article, nous présentons notre implémentation *open-source* d'un discrétiseur à projection aléatoire utilisant SpeechBrain (Ravanelli *et al.*, 2021), et le résultat de nos premières expériences en comparant BestRQ à wav2vec 2.0. [2] Nous analysons le temps de calcul des pré-apprentissages ainsi que les performances sur les tâches suivantes : reconnaissance automatique de la parole, vérification automatique du locuteur, classification de l'intention et reconnaissance des émotions. Les résultats montrent qu'un discrétiseur à projection aléatoire peut atteindre des performances similaires à celles de wav2vec 2.0 sur ces différentes tâches, avec l'avantage supplémentaire de réduire de plus de la moitié le temps de pré-apprentissage auto-supervisé. Nous pensons que notre implémentation *open-source* pourra servir de point de départ à des recherches ultérieures, facilitant l'exploration de diverses

2. Le code de notre implémentation de BEST-RQ est disponible à https://github.com/speechbrain/speechbrain/pull/2309



(a) BEST-RQ (b) wav2vec 2.0

FIGURE 1 – Diagrammes de l'architecture BEST-RQ et de wav2vec 2.0. BEST-RQ opère sur des mel-spectrogrammes, utilise un discrétiseur statique et des couches de type *conformer*. De son côté, wav2vec 2.0 fonctionne sur de l'audio brut, apprend la discrétisation et utilise des couches de type *transformer*.

architectures pour l'apprentissage auto-supervisé (SSL) dans le cadre du traitement automatique de la parole.

2 Contexte

Bien qu'il existe d'autres modèles SSL efficaces pour le traitement de la parole, tels que HuBERT et data2vec (Chen *et al.*, 2023; Baevski *et al.*, 2023), wav2vec 2.0 semble plus répandu et utilisé par la communauté de la parole, avec plusieurs outils et modèles disponibles gratuitement, ce qui rend la comparaison plus aisée. Dans cette section, nous présentons une comparaison entre wav2vec 2.0 et BEST-RQ.

2.1 wav2vec 2.0 vs. BEST-RQ

BEST-RQ et wav2vec 2.0 présentent quelques différences majeures, l'une d'entre elles étant que **BEST-RQ n'opère pas directement sur la forme d'onde représentant le signal de parole**. Au lieu de cela, BEST-RQ traite plus classiquement des coefficients de banque de filtres log Mel à 80 dimensions avec deux couches CNN pour l'extracteur de représentations acoustiques. De son côté, wav2vec 2.0 opère sur la forme d'onde du signal audio et utilise sept couches CNN pour extraire des représentations acoustiques.

L'utilisation d'un extracteur de représentations acoustiques conçu à la main, comme les banques de filtres Mel, réduit considérablement la quantité de mémoire et de calculs nécessaires pour entraîner le modèle. Cependant, cela peut limiter les performances du modèle car les banques de filtres sont des vues compressées de la parole basées sur des connaissances psycho-acoustiques (c'est-à-dire basées sur l'expertise humaine) et, par conséquent, pourraient supprimer des indices acoustiques qu'un extracteur non construit à la main pourrait apprendre.

Une deuxième différence majeure est que **BEST-RQ utilise une couche linéaire initialisée de**

manière aléatoire et un livre de codes (*codebook*) pour la quantification et la discrétisation de l'audio. Ces deux composants sont gelés tout au long de l'apprentissage. Les banques de filtres Mel sont projetées via la couche linéaire, et l'indice de l'entrée du livre de codes la plus proche de la projection est utilisé comme cible. L'entrée du livre de codes la plus proche est trouvée en prenant l'*argmin* de la distance normalisée entre chaque entrée du livre de codes et la projection.

Ensuite, un masque est appliqué à une partie des banques de filtres Mel et l'objectif du modèle est de deviner les cibles correctes pour les sections masquées. Cette tâche est traitée comme une tâche de classification utilisant l'entropie croisée pour calculer la perte.

Inversement, dans wav2vec 2.0, il y a deux livres de codes qui sont appris en utilisant une fonction *softmax* de type Gumbel. La fonction de coût pour wav2vec 2.0 est plus complexe et consiste en une combinaison d'une fonction de type *constrative loss* (c'est-à-dire l'identification de l'étiquette correcte parmi un ensemble de fausses étiquettes sélectionnées au hasard) et d'une fonction de type *diversity loss* (pour empêcher le modèle de se réduire à l'utilisation d'une seule entrée dans le livre de codes).

Une dernière différence majeure est que **BEST-RQ utilise des couches de type *conformers* au lieu de couches de type *transformers***. Bien que l'utilisation de couches de type *conformers* dans une architecture wav2vec 2.0 a déjà été étudiée (Zhang *et al.*, 2020), nous ne considérons dans notre étude que l'architecture standard wav2vec 2.0, car notre objectif n'est pas de prouver qu'une méthodologie est meilleure qu'une autre, mais plutôt de montrer que notre implémentation fonctionne comme prévu et se compare favorablement à une autre solution bien connue. La figure 1 illustre schématiquement les principales différences entre ces deux architectures.

Dans (Chiu *et al.*, 2022), les auteurs ont montré qu'il est possible d'obtenir de bonnes performances en transcription automatique sans avoir à apprendre un extracteur de représentations acoustiques ou des livres de codes. En théorie, cela réduit considérablement la complexité du modèle, de la fonction de coût et de la passe arrière de mise à jour des poids. Le temps de pré-apprentissage n'ayant pas été indiqué dans l'article original de BEST-RQ, nous visons à montrer empiriquement comment cela affecte le pré-entraînement et proposons des évaluations sur transcription automatiques et trois autres tâches en aval. Pour des raisons de reproductibilité et pour permettre à d'autres chercheurs d'étudier les effets de l'utilisation d'un discrétiseur à projection aléatoire, notre implémentation est disponible sous licence libre.

3 Expériences

Pour nos expériences, nous pré-entraînons un modèle wav2vec 2.0 et notre modèle BEST-RQ, puis nous les utilisons pour un sous-ensemble des tâches du benchmarck MP3S (Zaiem *et al.*, 2023). Dans les sections suivantes, nous décrivons l'environnement de pré-apprentissage, l'architecture des modèles, et les tâches évaluées.

3.1 Paramètres de pré-apprentissage

Pour le pré-apprentissage, nous utilisons 960 heures de parole issues du corpus LibriSpeech (c'est-à-dire les sous-corpus *train-clean-100*, *train-clean-360* et *train-other-500*) (Panayotov *et al.*, 2015). L'apprentissage auto-supervisé de chacun des modèles est réalisé sur 42 époques ou environ 200k

itérations (*mises à jour des poids*) en utilisant huit cartes GPU Tesla V100 à 32Go de RAM. Nous sauvegardons un point de contrôle à l'époque 21 (environ 100k itérations), pour évaluer l'évolution de la performance au cours de l'apprentissage. Nous notons la différence entre ces versions du modèle en les nommant *100k* et *200k*.

Nous utilisons les mêmes paramètres de mise en lot (*batch*) dynamique pour les modèles wav2vec 2.0 et BEST-RQ, la taille maximale du lot étant de 100 secondes. Comme nous utilisons huit GPU, la taille totale des lots est de 800 secondes, soit environ 13,33 minutes. Nous avons choisi cette taille de lot pour pouvoir réaliser nos expériences sur de petits GPU et dans un temps limité en utilisant seulement quelques centaines d'heures de GPU.

3.2 Hyper-paramètres du modèle

Nous utilisons l'architecture du modèle wav2vec 2.0-base (Baevski *et al.*, 2020) comme référence, en utilisant l'implémentation disponible dans SpeechBrain. Nous avons seulement modifié les paramètres du lot dynamique pour avoir une taille de lot maximale de 100 secondes, comme indiqué précédemment.

Pour notre implémentation de BEST-RQ, nous suivons globalement la description des auteurs, tout en introduisant **quelques changements clés** découverts lors de nos expériences préliminaires, et qui se sont révélés très importants pour les performances du modèle en raison de la taille relativement petite de notre lot (13 minutes contre 18 heures dans l'article original des chercheurs de Google), la petite quantité de données de pré-entraînement (960 heures contre 12 millions dans (Zhang *et al.*, 2023)), et le nombre de GPUs à notre disposition. Nous **avont réduit le nombre de couches de type *conformers* de 24 à 12**. Ce choix a été fait pour correspondre au même nombre de couches que le modèle wav2vec 2.0-base, et a permis au modèle de s'adapter aux GPU utilisés. Nous avons **ajouté un *layer drop* avec une probabilité de 0,05**, nous avons **réduit le taux d'apprentissage initial à 0,0008** et **enfin nous avons augmenté le ratio de masquage à environ 60% de l'audio** (c'est-à-dire que 15% des trames du mel-spectrogramme sont sélectionnées aléatoirement pour être masquées avec les trois trames suivantes). Des résultats marquants de nos expériences préliminaires sont présentés dans la section 4.2.

3.3 Les tâches visées

Pour les tâches finales, nous suivons la méthodologie des benchmarks MP3S (Zaiem *et al.*, 2023) et SUPERB (Yang *et al.*, 2021), c'est-à-dire que le modèle SSL est gelé et que l'entrée du modèle neuronal spécifique à la tâche finale est une somme pondérée des sorties des couches cachées du modèle pré-entraîné.

Pour la tâche de **reconnaissance automatique de la parole**, nous utilisons respectivement le sous-corpus *train-clean-100 et dev-clean* pour l'apprentissage et la validation, puis nous évaluons sur les sous-ensembles *test-clean* et *test-other* de LibriSpeech (Panayotov *et al.*, 2015).

Pour cette tâche, le modèle neuronal spécifique à la tâche est composé de deux couches BiLSTM, suivie d'une couche linéaire, d'une couche softmax, puis d'une fonction de coût CTC (Zaiem *et al.*, 2023). Chaque couche BiLSTM a une taille de 1024 et se voit appliqué un dropout de 0,2 durant l'apprentissage. La métrique que nous utilisons pour évaluer les performances est le taux d'erreur sur

les mots (WER). Nous indiquons le WER avec et sans l'application du modèle de langage officiel [3] *4-gram*.

Pour la tâche de **vérification automatique du locuteur**, nous utilisons Voxceleb1 (Nagrani *et al.*, 2017). Ce jeu de données est composé d'énoncés de plus de 1 000 célébrités recueillis sur YouTube. Il s'agit d'une tâche de classification binaire où, étant donné deux fichiers audio, le modèle doit déterminer si le locuteur est le même ou non dans les deux enregistrements.

Nous utilisons l'architecture ECAPA-TDNN (Desplanques *et al.*, 2020) du benchmark MP3S. Pour mesurer les performances de l'ASV, nous utilisons l'*Equal Error Rate* (EER).

Pour **la classification des intentions**, nous utilisons SLURP (Bastianelli *et al.*, 2020), qui est connu pour être plus difficile que d'autres ensembles de données pour la même tâche, consistant en 177 locuteurs à partir de 72k fichiers audio totalisant 58 heures d'audio. La tâche consiste à classer un énoncé donné dans l'une des 18 catégories ou scénarios suivants : *email*, *calendar* ou *play* (comme dans *play next song*).

Pour l'architecture neuronale, nous utilisons à nouveau celle de MP3S. Cette architecture est composée une couche BiLSTM d'une taille de 1024, suivie d'une couche linéaire, d'une couche de *statistical pooling*, puis d'une couche linéaire pour la classification finale. La métrique utilisée pour cette tâche est la Précision.

Pour la tâche de **reconnaissance des émotions**, nous utilisons l'ensemble de données IEMOCAP (Busso *et al.*, 2008). Cet ensemble de données contient environ 12 heures de données provenant de 10 locuteurs jouant des scénarios avec quatre émotions différentes (neutre, heureux, triste et en colère). Les performances sont mesurées à l'aide d'une validation croisée sur 10 sous-ensembles (*10-fold cross validation*). Comme pour la tâche de vérification du locuteur, nous utilisons une architecture ECAPA-TDNN du benchmark MP3S.

Afin d'explorer les capacités de BEST-RQ au-delà de sa capacité à produire des représentations de parole, nous réalisons une expérience supplémentaire dans laquelle nous mettons à jour les poids du modèle pour la tâche de reconnaissance automatique de la parole sur les données LibriSpeech.

Le modèle est *fine-tuné* sur *train-clean-100* et nous l'évaluons ensuite sur *test-clean* et *test-other*, avec ou sans l'utilisation du modèle *4-gram*.

4 Résultats

Dans cette section, nous présentons des résultats sur le temps de pré-apprentissage, sur les tâches finales, ainsi que des résultats d'expériences préliminaires qu'il nous semble pertinent de partager.

4.1 Benchmark MP3S

Nos résultats sur les tâches présentées dans la section précédente sont rapportés dans le Tableau 1. Pour les tâches de vérification du locuteur, de classication d'intention et de reconnaissance des émotions, BEST-RQ est légèrement plus performant que wav2vec 2.0, alors que son pré-entraînement est environ 2,4 fois plus rapide que celui de wav2vec 2.0.

3. Disponible à l'adresse `openslr.org/11/`

Model / Task	# Par.	GPU hours	LibriSpeech train-100 ASR				VoxCeleb	SLURP	IEM.
Metric			WER ↓				EER ↓	Acc. ↑	Acc. ↑
			Clean	Clean LM	Other	Other LM	ASV	IC	ER
W2V2 100k	90.9M	130	16.26	10.63	40.17	30.83	4.56	72.8	60.9
W2V2 200k	90.9M	262	13.89	9.45	33.55	25.49	3.83	74.5	63.0
BRQ 100k	83.0M	54	16.79	10.79	38.09	28.31	3.84	74.3	61.3
BRQ 200k	83.0M	109	15.11	9.76	34.06	24.74	3.53	74.8	63.8

TABLE 1 – Résultats sur les tâches en aval à 100k et 200k pas. Les performances de BEST-RQ sont similaires à celles de wav2vec, mais avec moins de la moitié du temps d'apprentissage. Le temps d'apprentissage et le nombre de paramètres sont indiqués respectivement sous *GPU hours* et *# Par.*.

Model	Fine-Tune LibriSpeech train-100			
	Clean	Clean LM	Other	Other LM
W2V2 100k	16.42	10.29	36.62	27.25
W2V2 200k	13.47	8.73	29.64	21.92
BRQ 100k	14.59	9.00	31.49	23.04
BRQ 200k	12.21	7.78	26.81	19.67

TABLE 2 – Résultats de la mise au point sur l'ASR avec LibriSpeech *train-100*.

wav2vec 2.0 s'est avéré plus performant que BEST-RQ sans modèle de langage sur la tâche de reconnaissance automatique de la parole. Cependant, avec le modèle de langage, BEST-RQ obtient des résultats très proches de ceux de wav2vec 2.0 sur la tâche *test-clean* et même légèrement meilleurs sur la tâche *test-other*. Enfin, lorsque nous affinons BEST-RQ ou wav2vec 2.0 sur la tâche de reconnaissance automatique de la parole, BEST-RQ obtient de meilleurs résultats que wav2vec 2.0, comme l'illustrent les résultats présentés dans le tableau 2.

En examinant la différence entre les perfomances des modèles à 100k et 200k itérations dans ce tableau, nous remarquons wav2vec 2.0 s'améliore plus fortement durant l'apprentissage. Nous pensons que cela est dû au fait que la transformation du signal de parole en séquence d'*embeddings* dans wav2vec 2.0 doit être apprise – au contraire de BEST-RQ qui utilise des mel-spectrogrammes pré-calculés – ce qui implique une convergence plus lente.

4.2 Impact du taux de masquage

Avant de procéder aux expériences présentées dans la sous-section précédente, nous avons appris tous les modèles sur 18 époques, ou environ 87k itérations, sur 4 GPUs 2080Ti de 11Go, en nous focalisant uniquement sur les performances pour la reconnaissance de la parole sur le sous-corpus *dev-clean* de LibriSpeech. Nous avons fait varier le taux de masquage et la taille du livre de codes en utilisant des taux de masquage compris entre 1 % et 12 %.

Comme le montre le tableau 3, la réduction du livre de codes (CB) de 8192 à 1024 ne semble pas modifier les performances de manière cohérente ou significative. Pour un taux de masquage de 1 %, le livre de codes plus petit est légèrement plus performant, tandis que pour un taux de 10 %, il est légèrement moins performant. Au contraire, l'augmentation du nombre de trames masquées a un impact important, diminuant le WER sur les données *dev-clean* d'environ 15 % (d'un WER d'environ 35 % à un peu plus de 20 %) lorsque le taux de trames masquées passe de 1 à 12 %.

Mask %	Dev-Clean WER	CB
1%	34.08	1024
1%	36.45	8192
5%	25.24	8192
10%	21.10	1024
10%	20.68	8192
12%	20.11	8192

TABLE 3 – Impact de la modification du taux de masquage et de la taille du livre de codes. Le masque % concerne l'indice de la trame de départ choisi pour le masque. Étant donné que les trois trames suivantes sont également masquées, le taux de masquage réel est quatre fois plus important.

5 Discussion et conclusion

Dans ce travail, nous avons décrit notre implémentation *open-source* de BEST-RQ et nous l'avons comparé à wav2vec 2.0 en termes de temps de pré-entraînement et de performance sur diverses tâches en aval. BEST-RQ démontre des performances comparables à celles de wav2vec 2.0 tout en diminuant le temps de pré-apprentissage auto-supervisé de plus de la moitié. Le code de notre implémentation sera bientôt disponible dans la boîte à outils *SpeechBrain* (Ravanelli *et al.*, 2021).

Nous émettons l'hypothèse que BEST-RQ converge beaucoup plus rapidement que wav2vec 2.0 parce qu'il démarre avec des mel-spectrogrammes et qu'il a environ 8 millions de paramètres en moins, ce qui lui permet d'obtenir des performances comparables dans sa version à 200k itérations. Les différences de performance entre les modèles 100k et 200k suggèrent que le modèle wav2vec 2.0 pourrait surpasser BEST-RQ avec plus de temps ou d'autres paramètres d'entraînement (tels qu'une taille de lot plus importante).

Néanmoins, nous pensons que nos résultats sont révélateurs des capacités de ces méthodes avec seulement quelques centaines d'heures de GPU et nous continuerons dans cette voie pour nos travaux futurs.

Références

BAEVSKI A., BABU A., HSU W.-N. & AULI M. (2023). Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International Conference on Machine Learning*, p. 1416–1429 : PMLR.

BAEVSKI A., ZHOU Y., MOHAMED A. & AULI M. (2020). wav2vec 2.0 : A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, **33**, 12449–12460.

BASTIANELLI E., VANZO A., SWIETOJANSKI P. & RIESER V. (2020). SLURP : A Spoken Language Understanding Resource Package. In B. WEBBER, T. COHN, Y. HE & Y. LIU, Éds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 7252–7262, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.emnlp-main.588.

BUSSO C., BULUT M., LEE C.-C., KAZEMZADEH A., MOWER E., KIM S., CHANG J. N., LEE S. & NARAYANAN S. S. (2008). IEMOCAP : Interactive emotional dyadic motion capture database. *Language resources and evaluation*, **42**, 335–359.

CHEN W., CHANG X., PENG Y., NI Z., MAITI S. & WATANABE S. (2023). Reducing Barriers to Self-Supervised Learning : HuBERT Pre-training with Academic Compute. *arXiv preprint arXiv :2306.06672*.

CHIU C.-C., QIN J., ZHANG Y., YU J. & WU Y. (2022). Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, p. 3915–3924 : PMLR.

DESPLANQUES B., THIENPONDT J. & DEMUYNCK K. (2020). Ecapa-tdnn : Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv :2005.07143*.

MOHAMED A., LEE H.-Y., BORGHOLT L., HAVTORN J. D., EDIN J., IGEL C., KIRCHHOFF K., LI S.-W., LIVESCU K., MAALØE L. *et al.* (2022). Self-supervised speech representation learning : A review. *IEEE Journal of Selected Topics in Signal Processing*.

NAGRANI A., CHUNG J. S. & ZISSERMAN A. (2017). Voxceleb : a large-scale speaker identification dataset. *arXiv preprint arXiv :1706.08612*.

PANAYOTOV V., CHEN G., POVEY D. & KHUDANPUR S. (2015). Librispeech : an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, p. 5206–5210 : IEEE.

PARCOLLET T., NGUYEN H., EVAIN S., BOITO M. Z., PUPIER A., MDHAFFAR S., LE H., ALISAMIR S., TOMASHENKO N., DINARELLI M. *et al.* (2023a). LeBenchmark 2.0 : a Standardized, Replicable and Enhanced Framework for Self-supervised Representations of French Speech. *arXiv preprint arXiv :2309.05472*.

PARCOLLET T., ZHANG S., VAN DALEN R., RAMOS A. G. C. & BHATTACHARYA S. (2023b). On the (In) Efficiency of Acoustic Feature Extractors for Self-Supervised Speech Representation Learning. In *Interspeech 2023*.

RAVANELLI M., PARCOLLET T., PLANTINGA P., ROUHE A., CORNELL S., LUGOSCH L., SUBAKAN C., DAWALATABAD N., HEBA A., ZHONG J. *et al.* (2021). SpeechBrain : A general-purpose speech toolkit. *arXiv preprint arXiv :2106.04624*.

WHETTEN R., PARCOLLET T., DINARELLI M. & ESTÈVE Y. (2024). Open Implementation and Study of BEST-RQ for Speech Processing. In *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

YANG S.-W., CHI P.-H., CHUANG Y.-S., LAI C.-I. J., LAKHOTIA K., LIN Y. Y., LIU A. T., SHI J., CHANG X., LIN G.-T. *et al.* (2021). Superb : Speech processing universal performance benchmark. *arXiv preprint arXiv :2105.01051*.

ZAIEM S., KEMICHE Y., PARCOLLET T., ESSID S. & RAVANELLI M. (2023). Speech Self-Supervised Representation Benchmarking : Are We Doing it Right ? *arXiv preprint arXiv :2306.00452*.

ZHANG Y., HAN W., QIN J., WANG Y., BAPNA A., CHEN Z., CHEN N., LI B., AXELROD V., WANG G. *et al.* (2023). Google usm : Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv :2303.01037*.

ZHANG Y., QIN J., PARK D. S., HAN W., CHIU C.-C., PANG R., LE Q. V. & WU Y. (2020). Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv :2010.10504*.