Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

Lingua Libre à l'ère de l'automatisation: l'I.A. au service du crowdsourcing d'un corpus oral

Camille Lavigne[1] Florian Cuny[1]
(1) Université de Lorraine, 54000 Nancy, France
lavignecamille37@gmail.com, recherche@floriancuny.fr

RÉSUMÉ

Lingua Libre, projet participatif collectant de la production orale, a amassé plus de 380 heures d'enregistrements, 1 350 000 fichiers audio, dans près de 300 langues différentes. Le potentiel d'un tel jeu de données pour tester des hypothèses linguistiques ou pour des tâches d'ASR est prometteur, mais diminué par le manque d'harmonisation et de nettoyage systématique des données. Ce travail est un pas supplémentaire vers un jeu de données issu de Lingua Libre de grande qualité et standardisé. Ce travail révèle des disparités récurrentes entre les enregistrements et la transcription qui en est fournie. Ces erreurs, bien que rares, sont régulières, et potentiellement évitables. En effet, le modèle d'ASR Wav2Vec 2.0-Base après affinage est capable de détecter une large part de ces erreurs. Il pourrait être un puissant outil à la disposition du contributorat, pour les assister à la tâche de patrouille.

ABSTRACT

Lingua Libre in the Age of Automation : A.I. serving crowdsourcing of a speech corpora

Lingua Libre, a project for collecting speech, has amassed over 380 hours of recordings and 1,350,000 audio files in almost 300 different languages. The potential of such a dataset for testing linguistic hypotheses or for Automatic Speech Recognition (ASR) tasks is promising, but diminished by the lack of harmonization and systematic data cleaning. This work is an additional step toward a high-quality, standardized dataset from Lingua Libre. It reveals recurring disparities between the audio tracks and their provided transcriptions. These errors, although rare, are consistent and potentially avoidable. Indeed, the Wav2Vec 2.0-Base ASR model, after fine-tuning, is capable of detecting a large portion of these errors. It could become a powerful tool available to contributors to assist them with the patrol task.

MOTS-CLÉS : Wav2Vec, affinage, transcription phonétique, crowdsourcing, automatisation, reconnaissance automatique de la parole.

KEYWORDS: Wav2Vec, fine-tuning, phonetic transcription, crowdsourcing, automation, ASR.

ARTICLE : **Accepté à** CORIA-TALN.

1 Introduction

Avec l'avènement du *big data*, la collecte de données fiables est devenue un enjeu majeur. Cette qualité est d'autant plus recherchée que la quantité n'a plus grand-chose à offrir. Le web a déjà été entièrement récupéré par de grandes entreprises telles que OpenAI ou Google (Villalobos *et al.*, 2024),

Actes de l'atelier Science Participative pour les Données et Corpus Linguistiques 2025 (ParCol), pages 11–24.

soulevant des interrogations quant à la propriété intellectuelle et pour des gains en performance qui s'amenuisent. Dans ce contexte, les projets participatifs présentent de multiples avantages. Les corpus qu'ils construisent sont en général libres de droit, permettant un accès et une utilisation possible au plus grand nombre. Ils sont portés par une communauté et sont en conséquence peu coûteux. Lorsqu'ils sont bien encadrés, ils peuvent aboutir à des jeux de données de grande qualité.

Un exemple récent de réussite du *crowdsourcing* dans le domaine de la reconnaissance automatique de la parole (ASR) est le projet Common Voice (Ardila *et al.*, 2020). Sa genèse débute en 2019, avec le constat que les données de production de parole présentent à la fois un coût prohibitif et tendent à oublier les langues qui ne sont pas dominantes. L'initiative a depuis recueilli plusieurs dizaines de milliers d'heures d'enregistrement dans plus de 150 langues différentes. Il s'est aujourd'hui imposé comme un jeu de données incontournable pour la tâche de transcription automatique "*speech-to-text*".

Cette inflation de la taille des jeux de données s'est accompagnée d'une explosion en taille des modèles de *deep learning* qui utilisent ces jeux de données pour leur entraînement. Pour les tâches d'ASR, c'est Wav2Vec 2.0 (Baevski *et al.*, 2020) qui, en 2020, ouvre la voie. Il marque la première tentative fructueuse d'adapter l'apprentissage auto-supervisé au traitement automatique de la parole. Ce franc succès peut être attribué à son architecture révolutionnaire à l'époque de son introduction. L'approche privilégiée du moment était de combiner une couche convolutive pour réduire la dimension des données d'entrées en frame. Cette représentation quantisée pouvait ensuite être traitée par une couche récurrente, RNN (Hannun *et al.*, 2014), ou LSTM, frame par frame. Néanmoins, l'architecture même de ces réseaux récurrents les rend difficiles à paralléliser par nature. Cette difficulté à la parallélisation les rend difficiles à implémenter à grande échelle sur de larges jeux de données et avec un grand nombre de paramètres. Une des innovations de Wav2Vec 2.0 est de remplacer cette couche récurrente par une architecture transformer-encoder avec mécanisme d'attention. Cette innovation est elle-même directement inspirée des progrès dans le traitement automatique des données textuelles par le modèle Bert (Devlin *et al.*, 2019). Elle permet dès 2020 d'atteindre des architectures de tailles encore jamais vues : 300 000 000 de paramètres en l'occurrence pour Wav2Vec 2.0 en version Large.

Des années après la publication initiale, ses successeurs Wav2Vec XLSR (Conneau *et al.*, 2020) et WavLM (Chen *et al.*, 2022) comptent toujours parmi les modèles les plus performants pour les tâches d'ASR allant de la transcription orthographique ou phonétique (Xu *et al.*, 2021) automatique à la vérification de locuteur (Fan *et al.*, 2021). Dans le même temps, d'autres architectures ont fait parler d'elles, comme son homologue Whisper (Radford *et al.*, 2022). On déploie alors des modèles dont les paramètres se comptent en milliards, entraînés sur des corpus de centaines de milliers d'heures.

La tendance commence à s'inverser avec le dernier né de la famille Wav2Vec 2.0, MMS (Pratap *et al.*, 2023) dont l'une des prouesses citées dans l'abstract est de battre Whisper au jeu de la transcription avec un dataset d'entraînement dix fois plus petit. Il est notable que pour améliorer la qualité du jeu de données d'entraînement de MMS, une version antérieure de Wav2Vec 2.0 a été utilisée pour la segmentation des enregistrements audio. Il semble donc qu'avec de bons modèles, il est possible de construire de bons jeux de données qui à leur tour permettent d'entrainer de meilleurs modèles avec moins de données. Un cercle vertueux semble possible, en tous cas jusqu'à un certain point.

C'est dans ce contexte qu'évolue le projet Lingua Libre. Débuté en 2018 dans un format proche de celui de Common Voice, il accumule aujourd'hui plusieurs centaines d'heures d'enregistrement dans un grand nombre de langues. Avec une base de contributeurs et contributrices bénévoles active mais limitée par le temps et l'envergure du projet, se pose la question de l'automatisation partielle. L'ambition serait de construire un meilleur corpus avec un bon modèle pour entraîner un meilleur modèle et obtenir un meilleur corpus. Ce processus permettrait entre autres d'économiser la précieuse

énergie du contributorat.

2 Méthodologie

2.1 Lingua Libre en quelques chiffres

Lingua Libre est une initiative de *crowdsourcing* mise en place par Wikimédia France. Son fonctionnement est similaire à celui de Common Voice. Après avoir créé un profil sur le site et s'être vu attribué un identifiant unique, les contributeurs et contributrices sont invités à lire des mots et expressions de leur choix dans une langue qui peut être leur langue maternelle ou une langue apprise plus tard dans leur vie (ils peuvent alors préciser leur niveau de maîtrise, entre débutant jusqu'à locuteur natif). Leur voix est enregistrée durant la lecture du texte. Suite à la phase de lecture, le contributeur peut ré-écouter l'enregistrement de sa voix pour en vérifier la conformité avec le texte supposément lu. Enfin, il ne reste plus à l'utilisateur ou l'utilisatrice qu'à soumettre sa contribution à la base de données Lingua Libre. Il faut noter que dans l'état actuel de l'initiative, aucune vérification systématique par un pair ou une machine n'est effectuée après soumission d'une contribution, les éventuels défauts détectés a posteriori pouvant cependant être signalés au contributorat.

Depuis le lancement de la version alpha en août 2018, Lingua Libre a accumulé 1 394 748 enregistrements émanant de 2 581 locuteurs-locutrices uniques, ce dans 297 langues différentes. La quantité totale d'enregistrements est estimée à environ 400 heures, ce qui est en fait un large jeu de données selon les standards actuels. Il n'approche pas cependant l'échelle des corpus les plus larges du moment qui contiennent jusqu'à plusieurs dizaines de milliers d'heures d'enregistrements et de locuteurs différents (Pratap *et al.*, 2020; Ardila *et al.*, 2020). Une autre information importante concernant Lingua Libre est le déséquilibre qu'il existe entre le nombre de contributeurs hommes et femmes, avec un ratio d'environ deux tiers-un tiers.

Il est important de souligner la dimension multilingue de Lingua Libre, le français ne représentant qu'un tiers du nombre total d'enregistrements, et plus remarquable encore, moins de 6% pour l'anglais. Bien que dans la suite de ce travail, seul le français soit traité, le but serait évidemment d'étendre notre méthode à toutes les langues enregistrées sur Lingua Libre.

2.2 Types d'erreurs

Afin de proposer un outil d'assistance à la vérification des transcriptions, il est nécessaire de connaître la nature des erreurs présentes dans le jeu de données. Les solutions diffèrent en fonction de la nature de celles-ci. On a tout d'abord les problèmes dans la forme : audio défectueux, transcription vide ou contenant des caractères invalides... Ces erreurs pourraient être détectées en amont de l'ajout en base de données. Elles sont en général détectables avec un algorithme "*rule-based*" assurant l'intégrité de celles-ci. Nous verrons dans la partie résultats que ces erreurs sont rares, mais présentes dans le corpus Lingua Libre, et elles ralentissent le pré-traitement des données.

Viennent ensuite les erreurs de fond, qui se concrétisent en général par une piste audio inaudible ou une transcription qui ne reflète pas le contenu de celle-ci. Un cas courant de non-concordance entre la transcription et l'audio est un enregistrement qui commence après le début de la production de parole. On peut aussi trouver des cas où la transcription et la production de parole disent des choses

complètement différentes. Ces erreurs de fond sont plus difficiles à détecter, car il est en général impossible de calibrer un algorithme "*rule-based*" à cet effet. Ce sont ces erreurs pour lesquelles trouver un algorithme efficace est un enjeu.

Pour détecter ces erreurs, une méthode déjà proposée par Marjou (2021) pour ce même jeu de données consiste à entraîner un modèle de *deep learning* sur une tâche de transcription phonétique. Il est ensuite possible de comparer la transcription phonétique générée automatiquement par le modèle à l'inférence et avec une traduction en API de la transcription qui a été faite par l'utilisateur. Un décalage trop grand entre prédiction et transcription du contributeur peut indiquer une erreur.

Il serait également possible d'appliquer la même méthodologie avec les transcriptions orthographiques. Nous préférons cependant suivre la méthode qui utilise la transcription phonétique car elle nous semble plus robuste, le français ayant une orthographe non phonétique.

2.3 Conception du jeu de données

Concernant le français spécifiquement, Lingua Libre recense, au 16 avril 2025 (date de collecte) : 417 913 pistes audio venant de 666 locuteurs-locutrices uniques. Si Lingua Libre rend disponible au téléchargement les enregistrements sous la forme de jeux de données par langue [1], celui pour le français n'a pas été mis à jour depuis le 2 août 2023. Pour bénéficier des enregistrements les plus récents, nous avons donc mis en place notre propre pipeline de téléchargement [2].

Cependant, Lingua Libre ne propose pas de transcription phonétique, mais uniquement la transcription orthographique, correspondant au mot ou à la locution que le locuteur a lu lors de l'enregistrement.

Pour récupérer la transcription de l'audio sous une forme phonétique, nous utilisons le Wiktionnaire [3], dans une extraction de ses données datant du 1er avril 2025 [4]. Pour chaque transcription orthographique unique dans notre liste d'enregistrements (247 275 transcriptions), nous récupérons les prononciations en alphabet phonétique international de la section "Français" de la page correspondante sur le Wiktionnaire. Si la page n'existe pas, si la section en français n'existe pas, si aucune prononciation n'est renseignée (52 666 transcriptions concernées) ou si plusieurs prononciations sont renseignées (homographes non homophones, variantes régionales…) (8 038 transcriptions concernées), nous retirons de notre jeu de données les enregistrements audio associés. Ainsi, nous conservons uniquement les enregistrements pour lesquels une unique prononciation phonétique est renseignée sur le Wiktionnaire, soit un total de 328 059 enregistrements partagés entre 529 locuteurs-locutrices. Par conséquent, nous avons écarté environ 21,5 % des enregistrements en français disponibles sur Lingua Libre.

En revanche, les transcriptions longues ainsi que les mots ayant deux voyelles qui se suivent sont conservés, en opposition avec Marjou (2021).

1. https://lingualibre.org/datasets/
2. Le code de cette pipeline ainsi que notre modèle sont disponibles à l'adresse : https://codeberg.org/Poslovitch/wav2vec-lili
3. https://fr.wiktionary.org
4. https://dumps.wikimedia.org/frwiktionary/20250401/

2.4 Nettoyage et standardisation du jeu de données

Si l'on s'intéresse aux caractéristiques de ces enregistrements présentées dans la Table 1, on s'aperçoit sans surprise que les enregistrements sont courts, aussi bien en durée d'enregistrement qu'en longueur de transcription. Cette observation est à attendre puisque pour Lingua Libre, l'unité n'est pas la phrase comme c'est le cas pour le corpus Common Voice : l'unité de base est le mot, et plus rarement des expressions figées plus longues.

	min	**moy.**	**max**	**total**
Durée (seconde)	0.32	1.23	6.51	402 990
Nombre de phones	1	10.97	105	3 598 275
Nombre de mots	1	1.39	24	454 579

TABLE 1 – Caractéristique des enregistrements du corpus Lingua Libre pour le français. Un "mot" désigne ici une suite de lettres de l'alphabet séparées par un espace.

Concernant le nettoyage et la standardisation d'un jeu de données pour la parole, les points qui suivent sont directement inspirés de ceux mis en avant par Junczyk (2024). Ils avancent que tout jeu de données a un but et doit être approprié à une tâche précise. Lingua Libre a d'ores et déjà été utilisé dans la recherche (Hutin & Allassonnière-Tang, 2022a,b, 2024) pour tester des hypothèses linguistiques. Les conclusions sont positives sur le potentiel du projet pour l'étude des langues. Il s'agit ici d'un premier cas d'utilisation à des fins de recherche en linguistique. Lingua Libre attribuant un identifiant unique pour chaque utilisateur, celui-ci pourrait également être utilisé sur des tâches de *speaker verification*. Enfin, à chaque enregistrement audio est associée sa transcription orthographique : il est donc envisageable de l'exploiter aussi à des fins de transcription automatique. C'est sur un analogue de cette tâche de transcription automatique que l'accent sera mis ici.

Toujours d'après Junczyk (2024), un jeu de données doit être accessible et découvrable. Sur ces points Lingua Libre est exemplaire, avec des audios ayant soit une licence Creative Commons 4.0 Attribution-ShareAlike [5], soit Creative Commons 4.0 Attribution [6], soit Creative Commons Zéro [7] permissive. Le jeu de données (bien qu'obsolète) est téléchargeable gratuitement depuis le site officiel [8].

Un bon jeu de données est également diversifié. Il est vrai que ce corpus est constitué essentiellement de mots et expressions figées. Les productions langagières sont courtes, 1.23 secondes en moyenne, donc faciles à apprendre et généraliser en théorie pour un modèle d'ASR State-of-The-Art. Néanmoins il présente l'avantage d'inventorier la diversité de la productivité de la langue française au travers de nombreux néologismes et d'une certaine diversité géographique pas toujours aisés à transcrire pour les ASR actuels. De plus, l'inconvénient des productions courtes peut être tourné en avantage par le biais de la *data-augmentation*.

Un jeu de données doit être annoté. C'est une des faiblesses soulignée de manière récurrente par la communauté de chercheurs et chercheuses en linguistique l'ayant utilisé. Il ne s'agit de pas jeter la pierre au contributorat, la question des informations à caractère personnel n'étant pas un sujet à

5. https://creativecommons.org/licenses/by-sa/4.0/
6. https://creativecommons.org/licenses/by/4.0/
7. https://creativecommons.org/publicdomain/zero/1.0/
8. https://lingualibre.org/wiki/LinguaLibre:Main_Page

prendre à la légère. Néanmoins, il est certain que le manque de détails sur le profil locuteur (l'âge, notamment) limite son utilisation.

On s'attend également à ce que le jeu de données soit nettoyé. Il n'existe pas à ce jour de processus systématique de vérification par les pairs tel qu'implémenté par le projet Common Voice. Bien que la qualité des transcriptions semble globalement bonne car suffisante pour tester des hypothèses linguistiques, un certain nombre d'erreurs récurrentes pourrait être évité si un processus de vérification systématique était mis en place. Pour assurer une meilleure qualité des transcriptions sans ajouter une charge de travail supplémentaire aux contributeurs-contributrices bénévoles, l'intelligence artificielle pourrait se révéler un allié de poids.

2.5 Partition et pré-traitement du jeu de données

La durée maximum des pistes audio étant inférieure à la taille maximale de la fenêtre de Wav2Vec 2.0 (environ 10 secondes), tous les échantillons ont pu être conservés sans troncation.

Aucune partition pour l'entrainement, la validation et le test n'a été proposée à ce jour à notre connaissance. La partition proposée pour cette expérience consiste à retenir les exemples dont l'index au modulo 7 est égal à 0 pour le jeu de validation et les exemples dont l'index au modulo 20 est égal à 0 pour le jeu de test[9]. Cette partition n'est pas utilisable pour de la vérification de locuteur car les locuteurs et locutrices du jeu de test sont également présents dans les jeux d'entrainement et de validation. Cette partition est détaillée dans la Table 2.

	nombre d'enregistrements	**durée (h)**	**part (%)**
Entrainement	266 608	91	81.4
Validation	46 761	16	14.3
Test	14 027	6	4.3

TABLE 2 – Partition finale du jeu de données Lingua Libre

2.6 Modèle et hyper-paramètres

Dans le but d'assister le contributeur dans la tâche de correction des erreurs de transcriptions, deux familles de modèles sont envisageables. Whisper, modèle *open source* de OpenAI pour la transcription automatique, pourrait théoriquement être affiné pour de la transcription phonétique. Il présente cependant l'inconvénient d'être gourmand en ressource de calculs, à cause de sa nature auto-régressive. À l'inverse, Wav2Vec 2.0 est plus économe, celui-ci ne nécessitant qu'un unique "*forward pass*" pour obtenir une prédiction sur toute la transcription. Parce que nos ressources en calculs sont très limitées, nous avons choisi ce dernier dans sa version Base de 94 000 000 de paramètres.

L'implémentation est faite en Pytorch pour permettre plus de liberté quand à la *data-augmentation* appliquée durant l'entraînement. Dans sa version originale, les auteurs de Wav2Vec 2.0 utilisent une stratégie de *masking* similaire au framework specAugment (Park *et al.*, 2019). Aucune *data-augmentation* de ce type n'est appliquée pour cet affinage car elle allonge significativement le temps

9. Un document csv contenant la liste des fichiers audios et leur partition est disponible à côté du code source : https://codeberg.org/Poslovitch/wav2vec-lili

de convergence du modèle. Il est probable que de meilleurs résultats pourraient être obtenus avec un modèle de taille plus large (300 millions à 1 milliard de paramètres) et une *data-augmentation* plus agressive.

Pour compenser le manque de transcriptions longues, chaque échantillon passé en entrée du modèle est une concaténation de 1 à 5 enregistrements audio et de leurs transcriptions. Ceux-ci sont sélectionnés et concaténés aléatoirement durant l'affinage, introduisant une plus grande diversité dans les exemples étudiés par le modèle. De plus, cette technique d'augmentation pourrait favoriser l'émergence d'un modèle sémantiquement agnostique, avec une moindre capacité de modélisation du langage, les transcriptions concaténées n'ayant aucun sens. Cela permet une meilleure modélisation du contenu de la piste audio. Cette hypothèse reste néanmoins à tester et sera laissée en suspend dans le présent travail.

Seule une carte graphique NVIDIA GeForce RTX 3050 Laptop 4GB VRAM était disponible pour l'affinage. L'entraînement a donc a eu lieu sur plusieurs jours. Il a été nécessaire de recourir à la méthode d'accumulation de gradient pour compenser la petite taille du batch, fixé à 4. Avec une accumulation de 16 *backward pass*, la taille effective du batch après accumulation est de 64. Le modèle est optimisé sur la base du *Connectionist Temporal Classification loss*. Cette fonction de coût peut s'avérer difficile à faire converger. Pour stabiliser et accélerer la convergence, un *gradient clipping* d'une valeur maximum de 0.3 est appliqué durant l'affinage. L'optimiseur Adam (Kingma & Ba, 2017) est utilisé avec un *learning rate* à $4e^{-6}$. Le *dropout* est fixé à 0.1 dans toutes les couches du modèle. Pour accélérer les temps de calcul, le modèle est entraîné en *mixed precision* FP16 (Micikevicius *et al.*, 2018).

2.7 Métriques

Trois évaluations distinctes sont proposées. Une première partie sera consacrée aux erreurs de forme. Bien qu'elle soit succinte, cette analyse présente un intérêt car elle est indicative de la qualité générale des données sur laquelle repose le reste des analyses. Elle pourrait également inviter les gestionnaires de la base de données Lingua Libre à prendre des mesures pour éviter l'ajout d'entrées erronnées. Pour les transcriptions, un test d'intégrité simple consiste à vérifier que les unicodes des caractères présents dans les transcriptions font partie de l'ensemble des unicodes de l'Alphabet Phonétique International (API) standard [10]. Il faut aussi s'assurer que la transcription phonétique n'est pas vide. Enfin, pour les pistes audios, il s'assurer que leur format soit conforme à ce qui est attendu, c'est-à-dire un tableau d'*integer* de dimension 1 *channel* (mono) et n *time step*.

Dans un deuxième temps, il convient d'évaluer la performance de notre Wav2Vec Lingua Libre (LiLi) sur le jeu de données de test après affinage. La métrique utilisée pour cette tâche est la Phoneme Error Rate (PER). Le modèle pré-entrainé Wav2Vec Phoneme (Xu *et al.*, 2021) servira de baseline. La comparaison n'est pas parfaite dans la mesure où Wav2Vec Phoneme est entrainé à produire des transcriptions avec un plus grand nombre de caractères de l'API pour permettre ses capacités multi-langue. Il sera donc évalué avec les transcriptions phonologiques générées par la librairie eSpeak à partir des transcriptions orthographiques de Lingua Libre comme ce fut le cas pour son entraînement. Cette baseline, bien qu'imparfaite, donne un ordre de grandeur de ce qu'il est possible d'obtenir en terme de performance sur des tâches de transcriptions phonétiques sur notre jeu de test.

10. Une liste des unicodes correspondants est disponible à : https://en.wikipedia.org/wiki/Phonetic_symbols_in_Unicode

Le modèle est utilisé tel que proposé par la librairie transformers de Hugging Face [11] sans affinage préalable. Notons que Wav2Vec Phoneme est elle-même une version affinée de Wav2Vec 2.0 Large qui comporte 3 fois plus de paramètres que Wav2Vec 2.0 Base. Notre Wav2Vec LiLi sera également évalué face au modèle GIPFA, entraîné sur une version antérieure du corpus. Une fois de plus, la comparaison n'est pas parfaite, dans la mesure où la partition du corpus n'est pas la même, ainsi que le pré-traitement.

Dans un troisième temps, la capacité de notre modèle affiné à détecter des erreurs de transcription sera testée. Pour détecter des erreurs de concordance entre transcription et contenu de la piste audio, nous comparons la distance de Levenshtein entre la transcription produite par notre Wav2Vec 2.0 affiné et le gold. Cette distance est ensuite normalisée par le nombre de phonèmes dans la transcription cible, car nous avons trouvé empiriquement que cette métrique donne de meilleures résultats après normalisation. Passé un certain seuil, une distance de Levenshtein élevée pourrait indiquer une transcription totalement erronée. Une vérification manuelle sera effectuée sur les 500 audios avec la plus haute distance de Levenshtein normalisée. Nous reportons ensuite le taux d'erreur pour différents seuils de distance de Levenshtein normalisée. Cette méthode de détection est directement reprise de Marjou (2021), à la différence que ce dernier ne normalise pas sa distance de Levenshtein. La question du seuil avait été laissée en suspend, d'où l'intérêt de cette analyse.

Tous les résultats, à l'exception des erreurs de forme évaluées sur l'ensemble du jeu de données, sont obtenus sur le jeu de test, à la différence de Marjou (2021) qui réalise toutes ses expériences sur l'ensemble de son jeu de données.

3 Résultats

3.1 Erreurs de forme

Pour ce qui est de la forme des transcriptions phonétiques, sur la totalité des 186 571 transcriptions uniques retenues, seules 26 d'entre elles ne sont pas valides : 23 contiennent des caractères qui n'existent pas dans l'API [12], 3 sont des transcriptions vides. Au total c'est moins de 0.001% des transcriptions phonétiques qui sont erronées dans la forme, ce qui semble attester de la qualité des prononciations dans les entrées du Wiktionnaire. Une correction systématique de ces quelques erreurs serait cependant bénéfique aux utilisateurs d'une version ultérieure du jeu de données.

Pour ce qui est des enregistrements audio, parmi les 328 059 enregistrements retenus, 12 ne respectent pas le format attendu. Ceux-ci semblent avoir été enregistrés dans un mode autre que mono, car lorsque qu'ils sont ouverts avec torchaudio, un tableau de dimension 2 pour les channels est retourné. Bien que cela représente une part infime du jeu de données, ce problème de forme est gênant car il fera planter le code de la plupart des informaticiens souhaitant utiliser le jeu de données. Une correction serait bienvenue.

Enfin, le sampling rate n'est pas standardisé : la plupart des pistes audio sont enregistrées en 44kHz ou 48kHz. Une standardisation en 44kHz ou 16kHz pour les tâches d'ASR pourrait aider à la prise en main du jeu de données.

11. Le modèle en question est librement accessible à l'URL suivante : https://huggingface.co/facebook/Wav2Vec2-lv-60-espeak-cv-ft

12. Les phonèmes du français considérés comme valides sont les suivants : i, e, ɛ, a, ɑ, ɔ, o, u, y, ø, œ, ə, ɛ̃, œ̃, ɑ̃, ɔ̃, ã, p, b, t, d, k, ɡ, f, v, s, z, ʃ, ʒ, m, n, ɲ, ŋ, l, ʁ, j, w, ɥ.

3.2 Performance de Wav2Vec 2.0 après affinage

L'affinage permet à notre modèle Wav2Vec LiLi de surpasser les performances de Wav2Vec Phoneme aussi bien en terme que PER que de transcriptions correctes (Table 3). Il n'est pas surprenant que notre modèle affiné performe mieux qu'un modèle généraliste multi-langue. La capacité de notre modèle à généraliser sur le jeu de test atteste de la qualité générale des données de Lingua Libre. Les performances de Wav2Vec Phoneme sont similaires à celles reportées par les auteurs sur d'autres langues.

	PER	**transcriptions correctes (%)**
Wav2Vec Phoneme (Xu *et al.*, 2021)	0.27	21
GIPFA (Marjou, 2021)	-	75
Wav2Vec LiLi	0.131	43.8

TABLE 3 – Comparaison des performances de Wav2Vec Phoneme, GIPFA et de Wav2Vec LiLi affiné sur le jeu de données Lingua Libre.

GIPFA ne rapporte pas la PER sur le jeu de test. Seul le pourcentage de transcriptions correctes est cité. L'écart de performance entre Wav2Vec LiLi et GIPFA est significatif, GIPFA produisant 50% plus de transcriptions sans erreurs. Cette différence peut s'expliquer par le manque d'entrainement de notre modèle. Celui-ci a encore une marge de progression conséquente. Plus encore, notre jeu de test contient probablement plus d'exemples difficiles en raison de notre pré-traitement moins restrictif. Malgré tout, avec un modèle plus large et un entraînement plus long, les 75% de transcriptions sans erreurs ne semblent pas inatteignables, même avec notre jeu de test plus difficile.

En plus de la performance brute du modèle, les auteurs de GIPFA s'intéressent à la performance du modèle en fonction de la longueur de la transcription. Cette question est pertinente dans la mesure où Lingua Libre contient peu de transcriptions longues. On devrait donc s'attendre à une diminution significative des perfomances sur les transcriptions longues. Les résultats obtenus par Wav2Vec LiLi sont comparables à ce qui a été obtenu par les auteurs de GIPFA pour les transcriptions correctes, comme le montre la Table 4. Pour les transcriptions incorrectes, la longueur moyenne est plus élevée pour Wav2Vec LiLi. Cette différence pourrait venir du fait que notre jeu de test contient plus de transcriptions longues que le jeu de GIPFA. Il est néanmoins impossible de vérifier cette hypothèse sans connaître la partition exacte du jeu de données pour GIPFA.

	longueur moy. correct	**longueur moy. incorrect**
GIPFA (Marjou, 2021)	7.51	8.65
Wav2Vec LiLi	7.45	9.20

TABLE 4 – Comparaison des longueurs moyennes (phonèmes) des transcriptions correctes et incorrectes de GIPFA et de Wav2Vec LiLi.

Cette comparaison des longueurs moyennes et du nombre de transcriptions correctes n'est néanmoins pas très informative. Que le modèle soit plus susceptible de faire une erreur quand la transcription est longue que lorsqu'elle est courte est un résultat attendu. Pour aller plus loin, il est possible de calculer la PER moyenne par tranche de longueur.

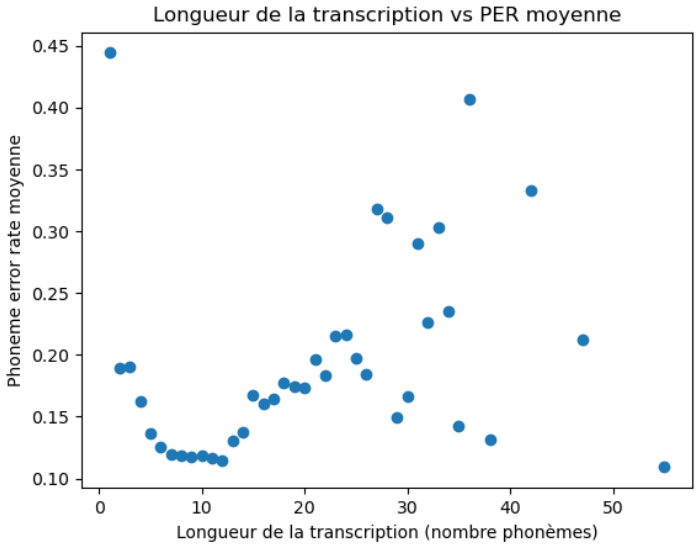

FIGURE 1 – PER moyenne en fonction de la longueur de la transcription

Dans la Figure 1, on remarque que les transcriptions très courtes ne sont pas mieux transcrites que les transcriptions longues. Dans les faits, les transcriptions ne contenant qu'un seul phonème ont la PER la plus haute, environ 0.45. Les transcriptions d'une longueur comprise entre 7 et 12 ont la PER la plus faible, autour de 0.12. Cette tranche optimale de 7 à 12 pourrait s'expliquer par le fait que la longueur moyenne des transcription du jeu de données est de 10.97 (Table 1). Il y a donc un grand nombre de transcriptions dont la longueur est comprise entre 7 et 12 phonèmes (hors *data-augmentation*). Les transcriptions longues, dont la longueur est supérieure à la moyenne, sont transcrites aussi bien que les transcriptions très courtes (< 6 phonèmes). Au delà de 25 phonèmes, le nombre d'exemples devient limité, la dispersion des points augmente et il devient difficile du conclure. Il serait intéressant de faire la même analyse sur un modèle entrainé sans *data-augmentation* pour constater l'efficacité de cette dernière.

3.3 Correction d'erreurs assistée

Une fois le modèle entrainé à transcrire la piste audio, il est possible de l'utiliser pour détecter les enregistrements défectueux. Sans surprise, plus le seuil de distance de Levenshtein choisi est élevé, plus le taux d'enregistrements considérés comme contenant un défaut est élevé.

Dans la Figure 2, on observe néanmoins que le taux d'audio défectueux n'est jamais de 100% : la meilleure performance obtenue pour un seuil de distance de Levenshtein normalisée est de 0.77 entraînant un taux de pistes audio défectueuses atteignant 38%. À noter que la corrélation entre la valeur du seuil et le taux d'audios défectueux est très forte, avec un coefficient de corrélation $r^2 = 0.952$. Noter également qu'avec un seuil supérieur à 1, le nombre d'échantillons devient faible (<50) et la corrélation diminue. Nous considérons ces valeurs comme des valeurs abhérentes dues à ce faible nombre d'échantillons et ne calculons pas les taux d'erreurs pour les valeurs de seuil supérieures à 1 pour garder un modèle linéaire stable.

Au vu de la corrélation qui existe entre distance de Levenshtein normalisée et taux d'audios défectueux, cet algorithme semble tout à fait viable à utiliser pour la détection d'erreurs. Il ne pourra néanmoins

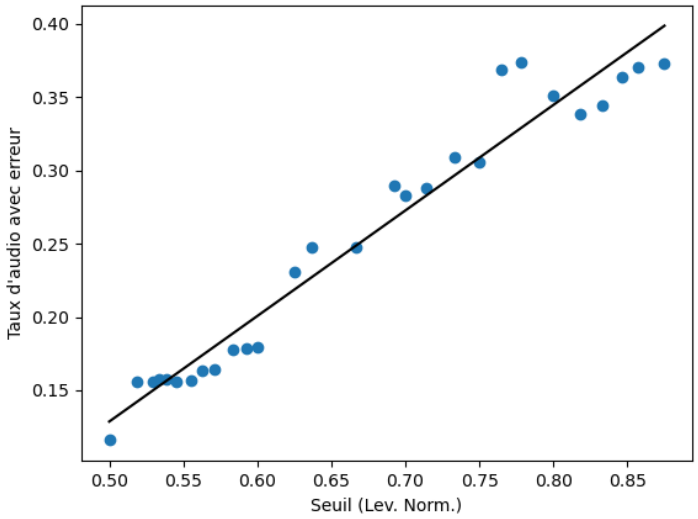

FIGURE 2 – Taux d'audios contenant une erreur en fonction du seuil retenu. La ligne noire correspond à la droite de régression appliquée au nuage de point.

se substituer à une intervention humaine. La question sous-jacente est donc de savoir si cet algorithme peut faire économiser du temps aux contributeurs-contributrices. On peut alors prendre le nombre d'audios défectueux corrigés par temps de correction cumulé (t) des contributeurs-contributrices $t = \frac{n \times c}{60}$.

Où n correspond au nombre total d'enregistrements à écouter, c une constante correspondant au temps passé par écoute, le tout divisé par 60 pour rapporter le temps en minutes. Il faut ensuite appliquer un ajustement logarithmique sur les valeurs temps car la relation temps-nombre d'erreurs corrigées est log-linéaire. On obtient alors le graphique en Figure 3.

La corrélation entre temps et nombre d'erreurs corrigées est de $r^2 = 0.978$ une fois l'ajustement logarithmique sur le temps appliqué. La corrélation est donc une fois de plus très forte. À cause de la nature logarithmique de cette relation, il est évident que plus le nombre d'audios corrigés est élevé, moins il devient rentable d'en corriger d'autres pour trouver des erreurs. Il faut bien garder à l'esprit que le nombre d'audios à corriger dépend directement du seuil choisi. Si le seuil retenu avait été de 0.5 et que les 500 audios au-dessus de ce seuil avaient été écoutés et corrigés tel qu'effectué par les auteurs, le temps passé aurait été d'environ 80 minutes et 58 erreurs corrigées.

Avec ce second modèle log-linéaire, il est possible d'estimer le nombre total d'erreurs contenues dans le jeu de test, l'ordonnée à l'origine de la droite de régression étant de 18.5 et sa pente de -20 :

$$NbErrorsTestSet = 18.5 * \log(\frac{14000 \times 10}{60}) - 20 = 123.46$$

Notre jeu de test contient environ 123 erreurs, si la relation découverte précédemment reste vraie pour un plus grand nombre d'enregistrements. Le temps nécessaire pour corriger les 14 000 enregistrements du jeu de test aurait été d'environ 2 333 minutes (environ 39 heures). Pour en corriger 62, soit la moitié des erreurs du jeu du test, seules 82 minutes sont nécessaires, soit 28 fois moins de temps de correction. L'algorithme pourrait donc permettre un gain de temps significatif dans la correction des erreurs de la base de données de Lingua Libre.

Cette algorithme peut être utilisé en production à la fois pour corriger au fur et à mesure les nouvelles

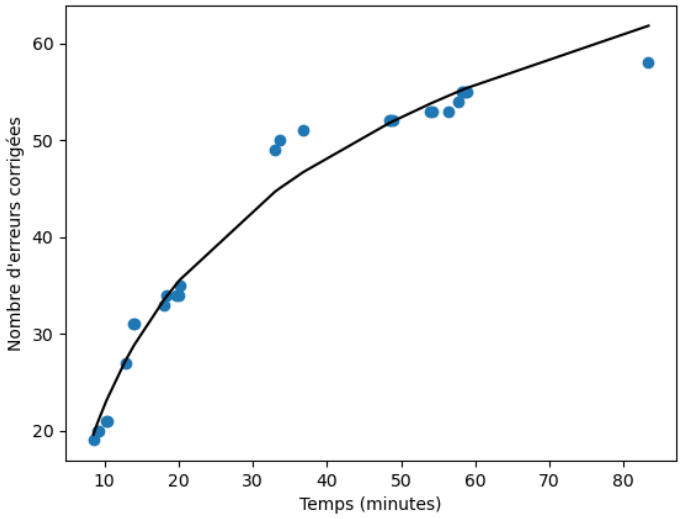

FIGURE 3 – Nombre d'audios défectueux corrigés par temps passé à l'écoute et la correction. La ligne noire correspond à la droite de régression appliquée au nuage de point. L'ajustement logarithme a été retiré pour faciliter la lecture.

entrées des locuteurs et locutrices. Mais il pourrait également servir à corriger les erreurs contenues dans la base de données existante. Au lieu de fixer un seuil particulier, une liste des enregistrements classés par distance de Levenshtein normalisée pourrait être utilisée. La base de données serait alors vérifiée et corrigée par ordre de priorité, petit à petit, selon le temps disponible du contributorat. Noter que si le nombre de 123 erreurs se révélait grossièrement juste, le taux d'erreurs contenues dans le jeu de test serait de $\frac{13}{14000} = 0.008$, moins de 1%.

4 Conclusion

Le jeu de données Lingua Libre a été présenté dans le détail. Ses principales forces sont le libre accès de ces données et le grand volume de données disponible. Ses principales faiblesses sont le manque de métadonnées à propos des locuteurs et l'absence d'un processus systématique de vérification des entrées en base de données. La qualité du jeu de données semble néanmoins bonne, avec un taux de transcriptions erronées estimé à moins de 1%.

Nous proposons ici une méthode simple pour partitionner le jeu de données pour de la transcription automatique. Après affinage sur ce jeu de données, le modèle Wav2Vec LiLi montre de bonnes performances pour la tâche de transcription phonétique, surpassant Wav2Vec Phoneme sur le jeu de test.

À partir de ce modèle affiné, nous présentons une méthode de détection des erreurs de transcription des pistes audio. Cette méthode vise à trier les entrées de la base par distance de Levenshtein normalisée décroissante. Les enregistrements ayant une distance de Levenshtein normalisée élevée contiennent probablement une erreur de transcription ou un fort bruit de fond, signalant ainsi une inadéquation entre ce qui a été prononcé et ce qui devrait être prononcé dans l'enregistrement. Cette méthode pourrait permettre un gain de temps considérable dans la correction des erreurs en base ainsi qu'une

vérification continue de la qualité des nouvelles entrées.

À l'avenir, il serait intéressant de comparer les performances sur de longues transcriptions avec et sans *data-augmentation*. Il faudrait également entrainer le modèle plus longtemps pour obtenir la meilleure performance possible de sa part. Reproduire l'expérience de GIPFA pour une comparaison plus juste serait tout autant intéressant. Peut-être également qu'un modèle aussi large que Wav2Vec 2.0 est inutile et qu'un modèle de dimension plus modeste serait suffisant pour cette tâche.

Les transcriptions phonétiques sont ici utilisées mais celles-ci présentent des limites handicapantes. Notamment, certains mots ne disposent pas de prononciation indiquée dans le Wiktionaire et ne peuvent donc être traités. Il serait par conséquent instructif de reproduire l'expérience en exploitant les transcriptions orthographiques et de comparer la différence de performance. Un système hybride pourrait être envisagé selon les résultats obtenus.

Remerciements

Nos remerciements au contributorat du Wiktionaire francophone et à toutes celles et ceux qui ont enregistré des mots sur Lingua Libre.

Références

ARDILA R., BRANSON M., DAVIS K., HENRETTY M., KOHLER M., MEYER J., MORAIS R., SAUNDERS L., TYERS F. M. & WEBER G. (2020). Common voice : A massively-multilingual speech corpus.

BAEVSKI A., ZHOU H., MOHAMED A. & AULI M. (2020). wav2vec 2.0 : A framework for self-supervised learning of speech representations.

CHEN S., WANG C., CHEN Z., WU Y., LIU S., CHEN Z., LI J., KANDA N., YOSHIOKA T., XIAO X., WU J., ZHOU L., REN S., QIAN Y., QIAN Y., WU J., ZENG M., YU X. & WEI F. (2022). Wavlm : Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, **16**(6), 1505–1518. DOI : 10.1109/jstsp.2022.3188113.

CONNEAU A., BAEVSKI A., COLLOBERT R., MOHAMED A. & AULI M. (2020). Unsupervised cross-lingual representation learning for speech recognition.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). Bert : Pre-training of deep bidirectional transformers for language understanding.

FAN Z., LI M., ZHOU S. & XU B. (2021). Exploring wav2vec 2.0 on speaker verification and language identification.

HANNUN A., CASE C., CASPER J., CATANZARO B., DIAMOS G., ELSEN E., PRENGER R., SATHEESH S., SENGUPTA S., COATES A. & NG A. Y. (2014). Deep speech : Scaling up end-to-end speech recognition.

HUTIN M. & ALLASSONNIÈRE-TANG M. (2022a). Investigating phonological theories with crowd-sourced data : The Inventory Size Hypothesis in the light of Lingua Libre. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, p. 23–28, Seattle, United States : Association for Computational Linguistics. DOI : 10.18653/v1/2022.sigmorphon-1.3, HAL : hal-03725715.

HUTIN M. & ALLASSONNIÈRE-TANG M. (2022b). Operation lili : Using crowd-sourced data and automatic alignment to investigate the phonetics and phonology of less-resourced languages. *Languages*, **7**(3). DOI : 10.3390/languages7030234.

HUTIN M. & ALLASSONNIÈRE-TANG M. (2024). L'apport des données participatives pour l'étude linguistique des français du monde : le cas de l'opposition /a/. *Journal of French Language Studies*, **34**(2), 249–272. DOI : 10.1017/S0959269523000200.

JUNCZYK M. (2024). Framework for curating speech datasets and evaluating asr systems : A case study for polish.

KINGMA D. P. & BA J. (2017). Adam : A method for stochastic optimization.

MARJOU X. (2021). Gipfa : Generating ipa pronunciation from audio.

MICIKEVICIUS P., NARANG S., ALBEN J., DIAMOS G., ELSEN E., GARCIA D., GINSBURG B., HOUSTON M., KUCHAIEV O., VENKATESH G. & WU H. (2018). Mixed precision training.

PARK D. S., CHAN W., ZHANG Y., CHIU C.-C., ZOPH B., CUBUK E. D. & LE Q. V. (2019). Specaugment : A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*, interspeech$_2$019 : $ISCA$. DOI : 10.21437/$interspeech$.2019 − 2680.

PRATAP V., TJANDRA A., SHI B., TOMASELLO P., BABU A., KUNDU S., ELKAHKY A., NI Z., VYAS A., FAZEL-ZARANDI M., BAEVSKI A., ADI Y., ZHANG X., HSU W.-N., CONNEAU A. & AULI M. (2023). Scaling speech technology to 1,000+ languages.

PRATAP V., XU Q., SRIRAM A., SYNNAEVE G. & COLLOBERT R. (2020). Mls : A large-scale multilingual dataset for speech research. In *Interspeech 2020*, interspeech$_2$020 : $ISCA$. DOI : 10.21437/$interspeech$.2020 − 2826.

RADFORD A., KIM J. W., XU T., BROCKMAN G., MCLEAVEY C. & SUTSKEVER I. (2022). Robust speech recognition via large-scale weak supervision.

VILLALOBOS P., HO A., SEVILLA J., BESIROGLU T., HEIM L. & HOBBHAHN M. (2024). Will we run out of data ? limits of llm scaling based on human-generated data.

XU Q., BAEVSKI A. & AULI M. (2021). Simple and effective zero-shot cross-lingual phoneme recognition.