Apprentissage de modèles frugaux pour les langues peu dotées à partir de larges modèles d'ASR

Mohammad Mohammadamini, Marie Tahon, Aghilas Sini, Antoine Laurent

Apprentissage de modèles frugaux pour les langues peu dotées à partir de larges modèles d'ASR

Mohammad Mohammadamini[1] Marie Tahon[1] Aghilas Sini[1] Antoine Laurent[1]
(1) LIUM, Le Mans Université, France
first.last@univ-lemans.fr

RÉSUMÉ

Les langues à faibles ressources ne souffrent pas uniquement d'un mangue de données langagières, mais sont également face à des ressources computationnelles contraintes. Les récents modèles d'ASR multilingues permettent de réduire la quantité de données nécessaires à l'entraînement de systèmes utilisables pour les langues à faibles ressources. Cependant, ils imposent une contraintes supplémentaires : la quantité de ressources de calcul requises pour l'entraînement et l'inférence. Nous démontrons qu'un avantage de ces modèles larges est de générer des données de bonne qualité pour entraîner ensuite des modèles frugaux dédiés à une langue peu dotée dont les performances sont proches de celles des modèles larges. Nos expériences sur le Kurde Central, une langue peu dotée, montrent qu'un modèle large (ici seamless) obtient un WER de 8.18 sur Asosoft (resp. 20.01 sur Fleurs), alors que notre modèle frugal atteint 7.57 (resp. 22.46) avec un modèle 75 fois plus petit.

ABSTRACT

Learning frugal models for low-resource langauges leveraging large ASR models

Low-resource languages often suffer not only from a lack of language resources but also from limited computational resources. Recent multilingual ASR models reduce the amount of data required to train a practical system for low-resource languages ; however, they impose another constraint : the need for substantial computational resources for training and inference. We demonstrate how these large models can be leveraged to generate high-quality speech recognition data for low-resource languages, enabling the training of lightweight models that achieve results close to those of the large models. Our experiments are conducted on Central Kurdish, which is a low-resource language. The obtained WERs of 8.18 and 20.01 on the Asosoft and Fleurs protocols for the Central Kurdish language using the Seamless large model, and attaining WERs of 7.57 and 22.46 on the same protocols with a 75x smaller model, demonstrates the efficiency of our proposed approach.

MOTS-CLÉS : pseudo-labels, langues peu dotées ASR, contraintes computationnelles ASR, langue Kurde.

KEYWORDS: pseudo-labeling, low-resource ASR, computation constraint ASR, Kurdish language.

1 Introduction

Les avancées récentes en reconnaissance automatique de la parole (*Automatic Speech Recognition* - ASR) à base de modèles auto-supervisés multilingues, ont permis de rendre cette technologie plus accessible pour les langues peu dotées (Baevski *et al.*, 2020; Barrault *et al.*, 2025; Radford *et al.*, 2022). L'intérêt majeur de ces modèles larges est de pouvoir être adapté avec peu de données tout en

maintenant des performances intéressantes pour les langues à faible resources. Malheureusement, leur utilisation nécessite une puissance de calcul conséquente inadaptée (en particulier lors de l'inférence) aux ressources disponibles dans les communautés linguistiques faiblement dotées. Cet article étudie le potentiel de ces modèles larges d'ASR pour générer des pseudo-labels de bonne qualité dans une langue peu dotée. Ces données peuvent ensuite être utilisées pour apprendre des modèles plus légers qui atteignent des performances prometteuses tout en réduisant significativement les coûts de calcul.

Une des difficultés dans l'utilisation de pseudo-labels est d'assurer leur qualité. (Hwang *et al.*, 2022) montre que les pseudo-labels générés par un modèle *maître* permettent d'entraîner des modèles *élèves* aussi performants que si les données avaient été annotées manuellement. Dans ces travaux, les échantillons labellisés automatiquement sont ensuite filtrés suivant une estimation de la confiance du modèle. De façon similaire, (Khurana *et al.*, 2021) propose une méthode pour générer des pseudo-labels à partir de plusieurs prédictions issues de différents dropouts. Les échantillons avec le meilleur accord sont sélectionnées pour l'adaptation au domaine. Dans (Lugosch *et al.*, 2022), un premier modèle multilingue est entraîné, puis *fine-tuné* à chaque langue. Ensuite des pseudo-labels sont générés par ces modèles fine-tunés, ce qui permet de re-entraîner le modèle multilingue sur ces pseudo-labels. Dans (Higuchi *et al.*, 2022) une paire de modèles online et offline interagissent et apprennent l'un de l'autre. L'interaction, basée sur une approche de type momentum, améliore graduellement la qualité des peuso-labels générés. Trouver le moyen le plus efficace pour apprendre un système d'ASR sur des pseudo-labels souvent bruités, reste un défi important.

La grande majorité des travaux sur les pseudo-labels ont réalisés sur des langues fortement dotées (Barrault *et al.*, 2025; Hwang *et al.*, 2022; Khurana *et al.*, 2021; Higuchi *et al.*, 2022), cependant on peut noter quelques initiatives sur des langues peu dotées (Nandi *et al.*, 2023; Getman *et al.*, 2024; Bhogale *et al.*, 2024). En effet, la motivation première est d'augmenter la quantité de données d'entraînement. Nous explorons l'utilisation de pseudo-labels dans le cas des langues peu dotées pour pallier à la fois à la rareté des données et aux contraintes de ressources, notamment en inférence.

Nous focalisons notre étude sur le Kurde Central, une langue faiblement dotée (Bamfo Odoom *et al.*, 2024; Veisi *et al.*, 2022; Mohammadamini *et al.*, 2025). Malgré de récents avancées en ASR (Veisi *et al.*, 2022), cette langue n'est toujours pas incluse dans les systèmes commerciaux. Le Kurde avec ses cinq dialectes, est parlé par plus de 35 millions de personnes, principalement au Moyen Orient. Nos travaux traitent du Kurde Central, parlé par 8 millions de locuteurs natifs (Sheyholislami, 2021) et écrit dans une version modifiée du script Arabe.

La section 2 décrit l'approche proposée et ses composants. Les données annotées manuellement et les audio bruts sont décrits dans la section 3. La section 4 présente le processus de génération et de sélection des pseudo-labels, et la dernière section 5 détaille les performances obtenues.

2 Approche proposée

L'approche développée pour l'apprentissage d'un modèle d'ASR furgal repose sur trois composants : la segmentation du signal de parole, un modèle d'ASR multilingue, et un modèle d'ASR frugal pour le Kurde Central. Nous utilisons une quantité limité de données annotées, mais une grande base d'enregistrements bruts (Figure 1). Tout d'abord le modèle multilingue Seamless v2 est fine-tuné à l'aide des données annotées. Ce modèle sert ensuite à générer des pseudo-labels à partir des enregistrements bruts préalablement segmentés. Enfin, un modèle frugal est appris avec les données

pseudo-labels sélectionnés après plusieurs filtrages. Tout au long de la chaîne de traitement, la même normalisation est appliquée (correction Unicode, ponctuation, nombre) (Mahmudi *et al.*, 2019).

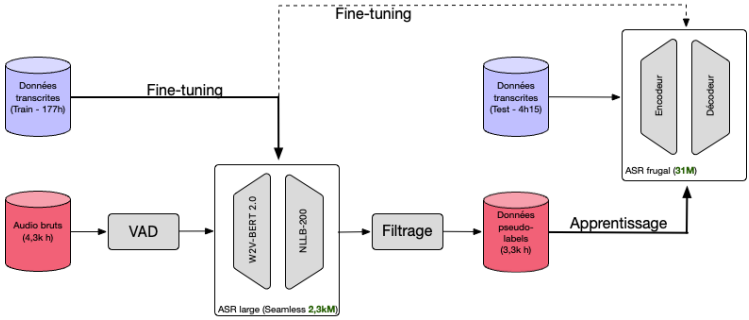

FIGURE 1 – *Approche proposée pour l'apprentissage d'un modèle d'ASR frugal.*

2.1 Segmentation

Plusieurs systèmes de segmentation existent : détection d'activité vocale (VAD) ou modèles neuronaux (DNN). Même si les modèles DNN atteignent de meilleures performances que les modèles VAD, ils restent très dépendant de la langue (Tsiamas *et al.*, 2022). Dans notre cas, nous avons observé que pour des enregistrements de bonne qualité audio, les approches VAD étaient plus robustes. Ainsi, nous avons implémenté un VAD basé sur une détection d'énergie. Pour un signal de parole donné, si plus de 30 trames (durée 10 ms) consécutives sont silencieuses, cela marque la fin d'un segment. Pour assurer des silences consistants au début et à la fin de chaque segment, nous conservons 10 trames de silences en début et fin des segments.

2.2 Modèle d'ASR large

Le deuxième composant est un modèle multilingue large. Il est fine-tuné sur les données annotées pour fournir ensuite des pseudo-labels de bonne qualité qui serviront à entraîner le modèle frugal. Seamless (Barrault *et al.*, 2025) est une collection de plusieurs modèles conçus pour la traduction textuelle à partir du texte (T2TT), de la parole (S2TT), ou la traduction orale à partir de la parole (S2ST) et de l'ASR. Dans nos travaux, seul le modèle d'ASR est utilisé. Il combine un encodeur de parole Wav2Vec-BERT 2.0 (Chung *et al.*, 2021) et un décodeur NLLB-200 (Costa-Jussà *et al.*, 2024). L'encodeur consiste en 24 couches de conformer entraînées sur 4,5 millions d'heures de parole provenant de 143 langues. Le signal audio est représentés par 80 bancs de filtres de Mel. La quantité de données par langue n'est pas connue. NLLB, appris initialement pour la traduction, sert ici de décodeur pré-entraîné. L'encodeur et le décodeur sont fine-tunés conjointement pour chacune des tâches traitant de la parole (S2TT, S2ST et ASR). La version finale de Seamless inclut 2,3 milliards de paramètres.

Une première évaluation de Seamless sur le Kurde Central montre qu'il doit être améliorer avant de générer des pseudo-labels de bonne qualité. Ainsi, l'adaptation est réalisée en le fine-tunant sur un corpus dédié au Kurde, annoté manuellement. Le modèle fine-tuné est alors utilisé pour générer les pseudo-labels.

2.3 Modèle d'ASR frugal

Le modèle d'ASR que nous avons développé consiste en une simple architecture transformer avec 12 couches dans l'encodeur, et 6 couches dans le décodeur, soit 31 millions de paramètres à apprendre – 75 fois moins que le modèle large. Les signaux sont représentés par des bancs de filtre log-Mel. Dans toutes nos expériences, nous apprenons soit un *tokenizer* unigramme SentencePiece spécifique à la langue, soit tokenizer au niveau caractères. Le tokenizer est entraîné sur les transcriptions générées par le modèle d'ASR large.

3 Ressources disponibles

3.1 Enregistrements de parole transcrits manuellement

Les données annotées manuellement proviennent de deux sources différentes. La première est Common Voice 18 [1] avec 117k échantillons validés. La second est la partition *train* du corpus Asosoft v1 qui contient 42.5k enregistrements de 700 phrases couvrant la distribution des diphones Kurdes (Veisi *et al.*, 2022). Le corpus total comptabilise un grand nombre d'enregistrements, cependant la quantité de phrases uniques (19.1k) reste un facteur limitant en terme de variabilité linguistique (Table 1, partie Train).

	Corpus	# audio	# Phrases uniques	Durée
Train	Common Voice 18	117k	18.4k	134h
	Asosoft v1 - *train* (Veisi *et al.*, 2022)	42.5k	700	43h
	Total	163.5k	19.1k	177h
Eval.	Asosoft v1 - *test* (Veisi *et al.*, 2022)	800	100	75min
	Fleurs test (Conneau *et al.*, 2022)	921	351	3h
	Total	1.721k	0.451k	4h15min

TABLE 1 – Données transcrites manuellement (train et évaluation)

3.2 Enregistrements bruts

Les données non transcrites proviennent d'enregistrements de livre audio disponibles publiquement. Nous avons collecté 1026 livres audio couvrant plusieurs thématiques, soit 4,3k heures d'enregistrements de parole de bonne qualité. Le nombre de livres audio par thématique est listé Table 2.

3.3 Données d'évaluation

Les modèles large et frugaux sont évalués sur deux corpus (Table 1, partie Eval.). Tout d'abord la partition de test de Asosoft v1, qui inclut 100 phrases uniques enregistrées dans un environnement contrôlé par 8 locuteurs natifs de Kurde Central. Cette partition couvre 10 thématiques différentes. La

1. https ://commonvoice.mozilla.org/en/datasets

Thématique	Quantité	Thématique	Quantité
Nouvelle	90	Religion	81
Roman	204	Histoire	85
Langage et théorie critique	45	Poésie	74
Politique	75	Auto/biographie	89
Contes traditionnels	68	Divers	56
Littérature jeunesse	20	Féminisme	25
Philosophie, Psychologie, Sociologie	114		
Total			1026

TABLE 2 – Thématiques couvertes par les enregistrements bruts

partition de test de Fleurs (Conneau *et al.*, 2022) contient des données parallèles dans 102 langues. Le sous ensemble en Kurde Central inclut 351 phrases uniques et 921 enregistrements.

4 Filtrage des pseudo-labels

Plusieurs facteurs peuvent contribuer à dégrader la qualité des pseudo-labels comme le bruit du signal audio fourni en entrée du système d'ASR large, la présence de musique en fond sonore, distortion du signal et autres artefacts. Pour limiter ces phénomènes, nous appliquons des filtres en séries pour retirer les échantillons problématiques générés à partir des enregistrements bruts décrits section 3.2.

i) Transcription partielle Pour identifier les phrases partiellement transcrites, nous limitons le nombre de mots par minute (WPM) à l'intervalle $[90; 200]$. Ce choix est motivé par le fait que le nombre moyen de mots prononcés à voix haute par minute est entre 117 et 239 (Brysbaert, 2019). Une marge confortable permet de conserver un grand nombre d'échantillons.
ii) Phrases longues ou courtes Les données dont la durée est inférieure 1 s (respectivement supérieure à 20 s) ou dont le nombre de tokens est inférieur à 3 (resp. supérieur à 50) sont écartées.
iii) Faible confiance Pour chaque pseudo-label, on calcule une estimation de la confiance à partir des scores de sortie du modèle d'ASR (*logits*) à l'aide de l'expression : $\frac{1}{N}\sum_{i=1}^{N} \max_j \text{Softmax}(\text{logits}_{i,j})$, avec N le nombre de token de la séquence, $\text{Softmax}(\text{logits})_{i,j}$ la pseudo-probabilité que le token j soit à la position i. Les échantillons dont la confiance est inférieure à 0.9 sont écartés.
iv) Présence de répétitions Dans certains cas, le modèle d'ASR génère une sortie répétitive sans aucun sens. Ainsi, si le pseudo-label contient n-grams ($n = 1, 2, 3$) avec plus de deux répétitions consécutives, il est supprimé. Ce critère permet de filtrer une proportion significative de signaux de mauvaise qualité, y compris des phrases avec du code-switching, ou de la parole dialectale.

Après avoir appliqué cette série de filtres, le nombre de pseudo-labels est réduit de 2.3 millions à 1.77 million, soit une durée totale de 3,3k heures d'enregistrement, ou encore 22,3 millions de tokens.

5 Résultats

5.1 Performance du modèle d'ASR large

Dans les expériences suivantes, les résultats présentés sont obtenus sur les deux corpus d'évaluation Asosoft et Fleurs décrits section 3.3. La première série d'expériences a été réalisée en utilisant le modèle Seamless large adapté pour le Kurde. La Table 3 présente les performances en taux d'erreur mots (WER) obtenues sur notre meilleur modèle. La première ligne est la baseline Seamless avant fine-tuning. La baseline obtient un WER de 24.04 sur Asosoft et 38.33 sur Fleurs. Nous rappelons que des données en Kurde ont été utilisées pour entraîner le modèle Seamless, mais nous ne savons pas en quelle proportion. Après adaptation à l'aide des données annotées manuellement, nous améliorons significativement le WER sur Asosoft à 8.18 et sur Fleurs à 20.31. Nous souhaitons mentionner ici que parmi les articles évaluant l'ASR sur Asosoft (Veisi *et al.*, 2022; Abdullah *et al.*, 2024), notre modèle atteint de meilleures performances (voir les deux dernières lignes).

Model	**Asosoft**	**Fleurs**
Seamless baseline	24.04	38.33
Seamless fine-tuned (ours)	**8.18**	**20.31**
XLS-R-2b + LM (Abdullah *et al.*, 2024)	11.8	-
SGMM + Lexicon + LM (Veisi *et al.*, 2022)	13.9	-

TABLE 3 – Résultats du grand modèle (WER)

5.2 Performances du modèle d'ASR frugal

Pour les modèles frugaux, le tokenizer unigram SentencePiece exploitant le texte généré par le modèle d'ASR large, a été entraîné avec plusieurs tailles de vocabulaire allant de 2k à 10k pour étudier l'influence du vocabulaire sur la capacité de généralisation du modèle d'ASR. En effet, une de nos hypothèses est que la qualité du tokenizer peut avoir un impact important sur les performances du modèle. Les résultats en taux d'erreur en caractère (CER) et mot (WER) sont présentés Table 4.

Pour la première expérience, le petit modèle transformer est entraîné à partir de zéro uniquement avec les données **annotées manuellement (HA)**. Le taux d'erreur élevé peut s'expliquer par le fait que les données annotées sont linguistiquement limitées et empêche le modèle de généraliser suffisamment sur les jeux d'évaluation. Pour contourner ce problème, nous entraînons un tokenizer au niveau caractère (dernière colonne). Le modèle généralise mieux mais présente toujours des performances dégradées sur les deux corpus, en comparaison avec le modèle baseline fine-tuné.

Dans une seconde expérience, le modèle léger est cette fois entraîné sur les **pseudo-labels (PL)**. Le modèle atteint des performances proches de celles obtenues par le modèle baseline. Ces résultats démontrent sans ambiguité l'efficacité de la méthode proposée. Sur chaque corpus d'évaluation, nous observons une dégradation d'environ 2 points de WER par rapport à la baseline. Pour limiter cette dégradation, nous entraînons le modèle avec l'ensemble des données disponibles, annotées manuellement et pseudo-label (HA+PL). Les résultats montrent que cette approche est légèrement moins performante que d'entraîner sur PL uniquement. Nous pensons que ce comportement est dû au

Vocab	2k	5k	10k	char
		Fleurs test set		
HA	66.87/87.30	72.18/91.70	72.56/91.87	26.22/61.79
PL	5.92/22.67	6.06/22.83	6.04/22.83	7.051/24.51
HA+PL	5.97/22.55	6.11/23.00	6.28/23.42	7.63/25.91
PL→HA	**5.90/22.46**	**5.94/22.59**	**6.00/22.71**	**6.97/24.02**
HA+PL→HA	6.05/22.59	6.96/24.32	7.21/24.06	7.14/24.19
		Asosoft test set		
HA	31.23/50.84	60.34/75.01	68.74/90.91	4.80/25.39
PL	1.26/9.02	1.27/8.81	1.18/8.54	1.33/9.44
HA+PL	2.02/10.11	1.80/9.05	1.80/10.07	1.48/9.57
PL→HA	**1.09/7.67**	**1.12/8.52**	**1.15/8.42**	**1.21/8.76**
HA+PL→HA	2.19/10.47	2.47/10.09	2.67/12.43	1.56/9.21

TABLE 4 – Résultats ASR modèles légers (CER/WER) utilisant des données pseudo-étiquetées.

fait que les données HA contiennent beaucoup de répétitions de phrases identiques, ce qui pourrait déséquilibrer la distribution des mots dans les données d'entraînement.

Dans le cas PL→HA, le modèle est d'abord appris sur PL puis fine-tuné pour 5 époques sur HA. Pour différentes tailles de vocabulaire, on observe une légère amélioration du WER. Cette méthode pourrait permettre aux modèles de rattraper la propagation des erreurs existantes dans les données PL. Enfin pour les dernières lignes HA+PL→HA, les modèles sont entraînés sur un mix de données, puis fine-tunés sur les données HA pour 5 époques. Comme le montrent les résultats, cette approche dégrade les performances. La dernière colonne présente les résultats obtenus avec un tokenizer au niveau caractère. Comme évoqué précédemment, cette condition permet d'améliorer grandement les résultats pour les modèles entraînés sur HA uniquement, notamment parce qu'une partie des données ont été conçues pour couvrir l'espace phonétique de la langue Kurde. Ainsi, nous constatons que les données PL semblent avoir un impact très positif sur le modèle de tokenizer également.

# Paramètres (M)	Asosoft	Fleurs
15	2.38/13.20	9.49/31.70
31	1.27/8.81	6.06/22.83
76	1.09/7.93	5.60/21.81
268	**1.05/7.51**	**5.44/21.20**

TABLE 5 – Impact de la taille du modèle frugal sur les performances (CER/WER)

5.3 Compromis entre taille du modèle et performance

Pour aller plus loin dans l'analyse, nous avons testé plusieurs architectures des modèles frugaux. Les résultats présentés Table 5.2 sont obtenus avec des modèles tous entraînés uniquement sur les pseudo-labels et un tokenizer avec une taille de vocabulaire de 5k. Le plus petit modèle (15M paramètres)

est composé de 8 couches de transformers (6 pour l'encodeur, 2 pour le décodeur) avec une couche dense (FC) de taille 1024, et des tailles de projection pour les vecteurs K, V et Q de 256. Ce modèle obtient un WER de 13.20 sur Asosoft et 31.70 sur Fleurs, ce qui reste assez loin des résultats obtenus avec le modèle Seamless large. Dans le second modèle (31M), le nombre de couches de l'encodeur est augmenté à 12, et du décodeur à 6. La couche dense (FC) est augmenté également à 2048, et les tailles des projections de K, V et Q sont maintenues à 256. Cette architecture est celle qui a été évaluée dans la section précédente.

D'autres ajustements élargissent les couches denses (FC) à (512, 1024) et (1024,512), soit 76M de paramètres et à (1024,4096) et (4096,1024), soit 268M de paramètres. Nous observons que les trois dernières architectures atteignent des performances très proches voire meilleures que celles obtenues par le modèle Seamless large. Par exemple, avec 76M et 268M, le WER sur le corpus Asosoft dépasse celui obtenu par le modèle large.

5.4 Ressources de calcul pour l'inférence

Afin de mieux caractériser encore le caractère frugal de notre modèle en comparaison avec le modèle Seamless large, nous avons réalisé les inférences sur les deux corpus d'évaluation sur des machines CPUs 6-core Xeon E5-2696 CPUs et des GPUs 2-core RTX 8000. La taille des batchs est fixée à 6. Le modèle frugal (31M) a terminé l'inférence sur CPU en 36 s (resp. 136 s) sur Asosoft (resp. Fleurs). L'inférence du modèle Seamless (2,3kM) n'est pas réalisable en temps réel sur CPU, par contre sur GPU, le modèle frugal est 18 fois plus rapide (voir Table 6).

Modèle	**Params**	**Asosoft (75min)**			**Fleurs (179min)**		
		WER	**GPU**	**CPU**	**WER**	**GPU**	**CPU**
Modèle frugal	31M	7.67	**4s**	**36s**	22.46	**14s**	**2m18s**
Seamless Large v2	2.3B	8.18	75s	-	20.31	4m21s	-

TABLE 6 – Comparaison entre modèles large et frugal (WER, GPU, CPU)

6 Conclusion

Les récents modèles d'ASR multilingues, tels que Whisper et Seamless, réduisent la quantité de données nécessaires à l'adaptation de modèles pour des langues peu dotées. Cependant, ce gain n'est possible que si des ressources computationnelles conséquentes sont disponibles, ce qui n'est pas le cas dans la plupart des communautés linguistiques peu dotées. Nous proposons une approche qui exploite le potentiel de ces modèles pour la génération de pseudo-labels de qualité, et permet ainsi l'apprentissage de modèles frugaux dont les performances sont très prometteuses. Les modèles d'ASR développés dans notre étude sont 75 fois plus petits que Seamless large mais obtiennent des résultats similaires voir meilleurs. Une généralisation des expériences sur d'autres langues peu dotées permettra de confirmer la méthodologie proposée. De plus, l'analyse fine de la propagation des erreurs à travers les pseudo-labels apportera de nouvelles pistes d'exploration futures.

Remerciements

Ce travail a été financé par le projet DGA RAPID COMMUTE. Il a été réalisé grâce aux ressources HPC de GENCI-IDRIS (Grant 2022-AD01101256) ainsi que les ressources de calcul propres du LIUM.

Références

ABDULLAH A. A., VEISI H. & RASHID T. (2024). Breaking walls : Pioneering automatic speech recognition for central kurdish : End-to-end transformer paradigm.

BAEVSKI A., ZHOU H., MOHAMED A. & AULI M. (2020). wav2vec 2.0 : a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA : Curran Associates Inc.

BAMFO ODOOM B., PAOLA GARCIA PERERA L., HANSANTI P., BARRAULT L., ROPERS C., WIESNER M., MURRAY K., MOURACHKO A. & KOEHN P. (2024). Speech data from radio broadcasts for low resource languages. In E. SALESKY, M. FEDERICO & M. CARPUAT, Éds., *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, p. 134–139, Bangkok, Thailand (in-person and online) : Association for Computational Linguistics. DOI : 10.18653/v1/2024.iwslt-1.18.

BARRAULT L., CHUNG Y.-A., MEGLIOLI M. C. & OTHER (2025). Joint speech and text machine translation for up to 100 languages. *Nature*, **637**(8046), 587–593. DOI : 10.1038/s41586-024-08359-z.

BHOGALE K. S., MEHENDALE D., PARASA N., G S. K. R., JAVED T., KUMAR P. & KHAPRA M. M. (2024). Empowering low-resource language asr via large-scale pseudo labeling. In *Interspeech 2024*, p. 2519–2523. DOI : 10.21437/Interspeech.2024-2396.

BRYSBAERT M. (2019). How many words do we read per minute ? a review and meta-analysis of reading rate. *Journal of Memory and Language*, **109**, 104047. DOI : https ://doi.org/10.1016/j.jml.2019.104047.

CHUNG Y.-A., ZHANG Y., HAN W., CHIU C.-C., QIN J., PANG R. & WU Y. (2021). w2v-bert : Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, p. 244–250.

CONNEAU A., MA M., KHANUJA S., ZHANG Y., AXELROD V., DALMIA S., RIESA J., RIVERA C. & BAPNA A. (2022). Fleurs : Few-shot learning evaluation of universal representations of speech. *arXiv preprint arXiv :2205.12446*.

COSTA-JUSSÀ M. R., CROSS J., ÇELEBI O. *et al.* (2024). Scaling neural machine translation to 200 languages. *Nature*, **630**(8018), 841–846.

GETMAN Y., GROSZ T., HIOVAIN-ASIKAINEN K. & KURIMO M. (2024). Exploring adaptation techniques of large speech foundation models for low-resource asr : a case study on northern sámi. In *Interspeech 2024*, p. 2539–2543. DOI : 10.21437/Interspeech.2024-479.

HIGUCHI Y., MORITZ N., LE ROUX J. & HORI T. (2022). Momentum pseudo-labeling : Semi-supervised asr with continuously improving pseudo-labels. *IEEE Journal of Selected Topics in Signal Processing*, **16**(6), 1424–1438. DOI : 10.1109/JSTSP.2022.3195367.

HWANG D., SIM K. C., HUO Z. & STROHMAN T. (2022). Pseudo label is better than human label. In *Interspeech 2022*, p. 1421–1425. DOI : 10.21437/Interspeech.2022-11034.

KHURANA S., MORITZ N., HORI T. & ROUX J. L. (2021). Unsupervised domain adaptation for speech recognition via uncertainty driven self-training. p. 6553–6557. DOI : 10.1109/ICASSP39728.2021.9414299.

LUGOSCH L., LIKHOMANENKO T., SYNNAEVE G. & COLLOBERT R. (2022). Pseudo-labeling for massively multilingual speech recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 7687–7691. DOI : 10.1109/ICASSP43922.2022.9746832.

MAHMUDI A., VEISI H., MOHAMMADAMINI M. & HOSSEINI H. (2019). Automated kurdish text normalization.

MOHAMMADAMINI M., SINI A., TAHON M. & LAURENT A. (2025). Scaling pseudo-labeling data for end-to-end low-resource speech translation (the case of Kurdish language). In *Interspeech 2025*, p. 898–902. DOI : 10.21437/Interspeech.2025-887.

NANDI R. N., MENON M., MUNTASIR T., SARKER S., MUHTASEEM Q. S., ISLAM M. T., CHOWDHURY S. & ALAM F. (2023). Pseudo-labeling for domain-agnostic Bangla automatic speech recognition. p. 152–162. DOI : 10.18653/v1/2023.banglalp-1.16.

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

SHEYHOLISLAMI J. (2021). *The Cambridge History of the Kurds*, chapitre The History and Development of Literary Central Kurdish. Cambridge University Press.

TSIAMAS I., GÁLLEGO G. I., FONOLLOSA J. A. R. & COSTA-JUSSÀ M. R. (2022). Shas : Approaching optimal segmentation for end-to-end speech translation. In *Interspeech 2022*, p. 106–110. DOI : 10.21437/Interspeech.2022-59.

VEISI H., HOSSEINI H., MOHAMMADAMINI M., FATHY W. & MAHMUDI A. (2022). Jira : a central kurdish speech recognition system, designing and building speech corpus and pronunciation lexicon. *Language Resources and Evaluation*, **56**(3), 917–941. DOI : 10.1007/s10579-022-09594-4.