MonoASR: un modèle de reconnaissance vocale multilingue frugal et unifié

Ilyes Oukid[*,***], Bilal Faye[*], Hanane Azzag[*], Mustapha Lebbah[**], Said Yacine Boulahia[***]

[*]LIPN, UMR CNRS 7030, Université Sorbonne Paris Nord, 93430 Villetaneuse, France
nom@lipn.univ-paris13.fr
[**]Laboratoire DAVID, Université Paris Saclay, 78035 Versailles, France
mustapha.lebbah@uvsq.fr
[***]École Militaire Polytechnique, BP 17, Bordj El Bahri 16046, Alger, Algérie
boulahia.yacinesaid@gmail.com

Résumé. La reconnaissance automatique de la parole (RAP) convertit la langue parlée en texte et constitue un enjeu majeur. Les modèles récents, tels que *Massively Multilingual Speech* (MMS), couvrent des centaines de langues mais nécessitent l'ajout d'adaptateurs pour chaque langue, ce qui augmente le coût en paramètres et freine l'extensibilité, notamment pour les langues faiblement annotées. Nous introduisons MonoASR, un système multilingue frugal et unifié qui évite ces adaptateurs grâce à une Projection Linguistique Universelle (ULP). Celle-ci associe un token de langue appris aux représentations acoustiques, permettant d'utiliser le même modèle et les mêmes paramètres pour différentes langues. Testé sur le français (langue fortement annotée), l'arabe et le kabyle [1] (langues sous-représentées et complexes), MonoASR obtient des taux d'erreur (WER) inférieurs à MMS, confirmant sa robustesse, sa généralisation et son intérêt pour une transcription multilingue à faible coût. Le code est disponible à : `https://github.com/ilyesqlm/MonoASR`

1 Introduction

La reconnaissance automatique de la parole (RAP) multilingue vise à transcrire la parole dans plusieurs langues à l'aide d'un modèle unifié. Les récents progrès reposent sur l'apprentissage auto-supervisé et des architectures à base d'attention, capables de capturer des régularités phonétiques et syntaxiques tout en favorisant la généralisation interlinguistique (Rekesh et al., 2023; Lin et al., 2024; Sudo et al., 2024; Xue et al., 2024, 2025; Yan et al., 2025).
Parmi les approches récentes, **MMS** (Pratap et al., 2024) illustre l'efficacité des représentations auto-supervisées pour de nombreuses langues. MMS apprend des représentations acoustiques robustes à partir de larges corpus non annotés et utilise des adaptateurs spécifiques pour chaque langue afin d'atteindre de bonnes performances. Cependant, ce mécanisme augmente fortement le nombre de paramètres et complique l'extension du modèle à de nouvelles langues, en particulier lorsqu'elles disposent de peu de données annotées. Ces limitations motivent le besoin

1. Langue amazighe, appartenant à la famille des langues berbères d'Afrique du Nord.

d'une approche frugale et unifiée, capable de traiter plusieurs langues avec un seul ensemble de paramètres.

Pour répondre à ces enjeux, nous présentons **MonoASR**, un modèle multilingue frugal et extensible. MonoASR partage un ensemble unique de paramètres entre toutes les langues et intègre un token de langue appris, fusionné avec les caractéristiques acoustiques au sein d'un module de **Projection Linguistique Universelle (ULP)**. Cette architecture permet de capturer les spécificités linguistiques tout en maintenant un modèle compact et facilement extensible. Nous détaillons MonoASR dans la Section 3 et l'évaluons sur le français, l'arabe et le kabyle, montrant des gains significatifs de WER par rapport à MMS, en particulier pour les langues faiblement annotées.

2 État de l'art

La reconnaissance automatique de la parole (RAP) consiste à apprendre une fonction f_θ qui transforme une séquence acoustique $X = (x_1, \ldots, x_T)$ en une séquence de tokens $Y = (y_1, \ldots, y_N)$ correspondant à la transcription :

$$\hat{Y} = \arg\max_{Y} P(Y|X;\theta), \tag{1}$$

où θ représente les paramètres du modèle. La probabilité de la séquence peut être factorisée pour prédire chaque token en tenant compte des précédents :

$$P(Y|X;\theta) = \prod_{t=1}^{N} P(y_t|y_1, \ldots, y_{t-1}, X;\theta). \tag{2}$$

Les premières méthodes multilingues partageaient un modèle de base entre plusieurs langues, parfois en combinant des embeddings phonétiques ou linguistiques spécifiques. Ces modèles permettaient d'apprendre des représentations acoustiques communes, mais peinent à capturer les différences phonologiques et morphologiques entre langues éloignées.

Pour faciliter le passage à l'échelle, des architectures modulaires avec adaptateurs ont été proposées, comme Master-ASR (Yu et al., 2023). L'apprentissage pour une langue l se formalise ainsi :

$$\theta^*, \phi_l^* = \arg\min_{\theta,\phi_l} \mathcal{L}(f_{\theta,\phi_l}(X_l), Y_l), \tag{3}$$

où X_l et Y_l sont les séquences acoustiques et les transcriptions, θ les paramètres partagés et ϕ_l l'adaptateur spécifique à la langue l. Bien que cette approche permette un transfert partiel entre langues, l'adaptateur doit être appris séparément, ce qui augmente le coût en paramètres et limite l'extensibilité.

Les modèles généralistes, comme Whisper (Radford et al., 2023), s'entraînent sur de vastes corpus multilingues et peuvent générer des transcriptions pour des langues non vues à l'entraînement. Cependant, leur performance sur les langues faiblement annotées est limitée par le déséquilibre des données et la présence de bruit dans les annotations.

Les modèles auto-supervisés, tels que Wav2Vec 2.0 (Baevski et al., 2020) et MMS (Pratap et al., 2024), apprennent des représentations acoustiques robustes à partir de larges corpus non annotés. MMS introduit un adaptateur pour chaque langue. Cette approche améliore les performances, mais chaque adaptateur doit être entraîné séparément, ce qui ajoute plus de deux

millions de paramètres par langue et rend difficile l'ajout de nouvelles langues faiblement annotées. Le risque de sur-apprentissage ou de performances instables est élevé pour ces langues.

En résumé, la prise en charge multilingue repose souvent sur des adaptateurs coûteux ou sur des données massives. Il est difficile de concilier **frugalité**, **extensibilité** et **performance** sur les langues peu annotées. Pour pallier ces limites, nous proposons **MonoASR**, un modèle frugal et unifié qui partage tous ses paramètres entre langues et utilise un token de langue appris fusionné avec les représentations acoustiques, comme détaillé dans la Section 3.

3 Méthodologie

Nous décrivons ici **MonoASR**, une architecture multilingue unifiée pour la reconnaissance automatique de la parole. Le modèle repose sur un partage complet des paramètres entre langues et un conditionnement explicite par des tokens linguistiques appris.

L'architecture de MonoASR est composée des cinq modules suivants :

1. **Extracteur de Caractéristiques (FE)** : transforme le signal audio brut en représentations acoustiques de bas niveau.
2. **Projection de Caractéristiques (FP)** : ajuste la dimension et normalise les représentations pour les rendre compatibles avec les modules suivants.
3. **Projection Linguistique Universelle (ULP)** : encode les représentations acoustiques en tenant compte de la langue cible, grâce à un mécanisme de tokens linguistiques.
4. **Encodeur (En)** : capture les dépendances temporelles et contextuelles au sein des séquences.
5. **Tête de Langage (LM Head)** : projette les représentations encodées dans l'espace du vocabulaire cible pour produire des distributions de probabilité sur les transcriptions.

La Figure 1 illustre le pipeline complet de traitement dans MonoASR.

3.1 Architecture MonoASR

Soit un lot audio $\mathbf{x} \in \mathbb{R}^{N \times T \times d_{\text{in}}}$, où N est la taille du lot, T le nombre de pas temporels, et d_{in} la dimension brute des entrées (par ex. échantillons audio). L'objectif est de produire une séquence de distributions $\mathbf{z} \in \mathbb{R}^{N \times L \times M}$, où L est la longueur obtenue après le sous-échantillonnage réalisé par le FE et M la taille du vocabulaire cible.

Extracteur et Projection de Caractéristiques. Les deux composants sont dérivés d'un encodeur Wav2Vec 2.0 préentraîné (Baevski et al., 2020) et sont conservés **gelés** durant l'entraînement afin de réduire le coût de calcul et de faciliter l'apprentissage par transfert. L'FE extrait des représentations acoustiques intermédiaires capturant les traits phonétiques et prosodiques, puis FP les projette dans un espace latent de dimension d.

$$\mathbf{h} = \text{FP}(\text{FE}(\mathbf{x})) \in \mathbb{R}^{N \times L \times d}. \tag{4}$$

Cette étape permet de réduire la variabilité acoustique et de standardiser les représentations pour toutes les langues.

Projection Linguistique Universelle (ULP). Le module ULP est la pièce centrale de MonoASR. Son rôle est de permettre à un seul ensemble de paramètres partagés θ de représenter des langues différentes, sans recourir à des adaptateurs spécifiques comme dans MMS (Pratap et al., 2024).

Dans MMS, chaque langue l est associée à un adaptateur ϕ_l, et l'apprentissage s'écrit :

$$\hat{Y}_l = \arg\max_Y P(Y|X_l; \theta, \phi_l), \tag{5}$$

ce qui implique un coût paramétrique croissant avec le nombre de langues, chaque adaptateur ajoutant plus de deux millions de paramètres supplémentaires.

Dans MonoASR, nous remplaçons ces adaptateurs par un **token de langue appris** $\mathbf{t}_l \in \mathbb{R}^{1\times d}$, beaucoup plus léger. Ce token agit comme un *routeur*, en injectant l'information linguistique directement dans les représentations acoustiques, tout en maintenant **les mêmes paramètres partagés** pour toutes les langues.

Concrètement, pour une langue l, on concatène le token $\mathbf{t}_l$ aux représentations projetées $\mathbf{h}$:

$$\mathbf{h}' = \text{concat}(\mathbf{t}_l, \mathbf{h}) \in \mathbb{R}^{N\times(L+1)\times d}. \tag{6}$$

Cette séquence est ensuite encodée par $n = 4$ blocs Transformer (Vaswani et al., 2017) partagés, intégrant une normalisation RMSNorm (Zhang et Sennrich, 2019) :

$$\mathbf{u} = \text{ULP}(\mathbf{h}'; \theta) \in \mathbb{R}^{N\times(L+1)\times d}. \tag{7}$$

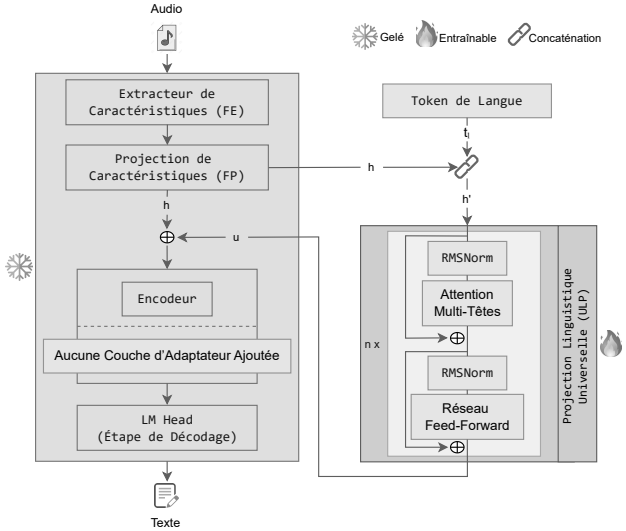

FIG. 1 – *Vue d'ensemble de l'architecture MonoASR et de ses cinq composants : Extracteur de Caractéristiques (FE), Projection (FP), Projection Linguistique Universelle (ULP), Encodeur et LM Head.*

En supprimant ensuite le token de sortie, on obtient :

$$\mathbf{u} = \mathbf{u}[:, 1 : L, :]. \tag{8}$$

Ainsi, contrairement à MMS où l'extension à $|\mathcal{L}|$ langues nécessite $|\mathcal{L}|$ adaptateurs $\{\phi_l\}$ distincts, MonoASR conserve un seul ensemble de paramètres θ et ne dépend que des tokens $\{\mathbf{t}_l\}$, dont la taille est négligeable.

Ce mécanisme fait du token de langue un **vecteur-routeur** : il oriente l'encodage des représentations acoustiques vers la langue cible, permettant une adaptation fine sans gonfler le nombre de paramètres ni multiplier les modules. C'est précisément cette frugalité qui rend MonoASR extensible et efficace sur les langues faiblement annotées.

Encodeur. Les sorties $\mathbf{u}$ sont combinées avec les représentations initiales $\mathbf{h}$ par un mécanisme résiduel :

$$\mathbf{e}_{\text{in}} = \mathbf{h} + \mathbf{u}. \tag{9}$$

L'encodeur traite ensuite $\mathbf{e}_{\text{in}}$ pour capturer les dépendances temporelles et contextuelles longues :

$$\mathbf{e} = \text{Encoder}(\mathbf{e}_{\text{in}}) \in \mathbb{R}^{N \times L \times d}. \tag{10}$$

Tête de Langage (LM Head). Une couche linéaire $W \in \mathbb{R}^{d \times M}$ et un biais $b \in \mathbb{R}^M$ projettent les représentations encodées dans l'espace du vocabulaire. Un softmax produit les distributions de probabilité :

$$\mathbf{z} = \text{Softmax}(W\mathbf{e} + b) \in \mathbb{R}^{N \times L \times M}. \tag{11}$$

Chaque pas temporel est ainsi associé à une distribution sur les unités de sortie (caractères, sous-mots, ou phonèmes).

3.2 Procédure d'entraînement

L'apprentissage de MonoASR repose sur deux stratégies complémentaires, suivies d'une fonction de perte adaptée à la reconnaissance de la parole.

1. **Entraînement simultané** : toutes les langues sont vues en parallèle dans des mini-lots mélangés.
2. **Entraînement progressif** : les langues sont introduites séquentiellement, en adaptant les paramètres partagés et les tokens linguistiques.
3. **Fonction de perte** : la perte CTC assure l'alignement implicite entre séquences acoustiques et transcriptions.

Entraînement simultané. On définit le corpus multilingue comme :

$$\mathcal{D} = \bigcup_{l \in \mathcal{L}} \{(\mathbf{x}_l^i, \mathbf{y}_l^i)\}_{i=1}^{N_l}, \tag{12}$$

où N_l est le nombre d'échantillons pour la langue l. À chaque itération, un mini-lot est tiré aléatoirement de $\mathcal{D}$. L'optimisation globale consiste à résoudre :

$$\theta^* = \arg\min_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \mathcal{L}(f_\theta(\mathbf{x}), \mathbf{y}), \tag{13}$$

où θ désigne tous les paramètres du modèle. Cette approche favorise la généralisation interlinguistique mais peut induire des conflits de gradients entre langues.

Entraînement progressif. Dans ce schéma, les langues sont introduites une par une. Supposons que le modèle ait été entraîné sur une langue source l_0. Lors de l'ajout d'une langue l_1, seuls les paramètres partagés θ et le token $\mathbf{t}_{l_1}$ sont mis à jour :

$$\theta^* = \arg\min_{\theta} \sum_{l\in\{l_0,\ldots,l_k\}} \mathcal{L}(f_\theta(\mathbf{x}_l), \mathbf{y}_l). \tag{14}$$

Cette approche limite l'interférence entre langues et stabilise l'apprentissage, mais elle peut introduire un risque d'oubli catastrophique pour les langues déjà vues.

Fonction de perte (CTC). Nous utilisons la *Connectionist Temporal Classification* (CTC) (Graves et al., 2006), qui permet d'apprendre un alignement implicite entre les sorties $\mathbf{z}$ et les transcriptions cibles $\mathbf{y}$. Soit $\pi = (\pi_1, \ldots, \pi_L)$ une séquence d'alignement de longueur L incluant le token spécial `blank`. La probabilité de $\mathbf{y}$ donnée $\mathbf{x}$ s'écrit :

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{y})} \prod_{t=1}^{L} z_{t,\pi_t}, \tag{15}$$

où $\mathcal{B}$ est la fonction de réduction supprimant les répétitions et les blancs.

La perte CTC correspond à la log-vraisemblance négative :

$$\mathcal{L}_{\text{CTC}} = -\log P(\mathbf{y}|\mathbf{x}). \tag{16}$$

Cette formulation permet au modèle d'apprendre directement à aligner les représentations acoustiques avec les transcriptions, sans annotation temporelle explicite.

4 Expérimentations

Nous évaluons **MonoASR** en le comparant à **MMS**, considéré comme un état de l'art pour la RAP multilingue. L'objectif de cette section est double : (i) valider que le **partage complet des paramètres** via les tokens de langue est compétitif, voire supérieur, aux adaptateurs spécifiques de MMS, et (ii) analyser dans quels contextes MonoASR tire le meilleur parti de son architecture frugale.

4.1 Jeux de données

Nous utilisons trois langues contrastées : le **kabyle** (TutlaytAI, 2024) (25 h, langue peu dotée et morphologiquement complexe), l'**arabe** (Mohamed, 2024) (7 h, langue sous-représentée et script non segmenté), et le **français** (odunola, 2024) (13 h, langue bien dotée).

Afin d'assurer une comparaison cohérente, nous avons :

— construit des vocabulaires au niveau caractère (59 pour le kabyle, 56 pour l'arabe, 43 pour le français), fusionnés en un vocabulaire **unifié** de 118 caractères ;

— rééchantillonné tous les fichiers audio à 16 kHz et défini des partitions train/validation/test homogènes ;

— appliqué un tokenizer commun, garantissant un traitement identique entre langues.

Cette configuration met toutes les langues sur un pied d'égalité, ce qui permet de mieux isoler l'impact de l'architecture.

4.2 Configuration expérimentale

Toutes les expériences sont conduites dans des conditions identiques pour garantir l'équité :

— **Optimisation** : AdamW avec taux d'apprentissage 1×10^{-3}, décroissance de poids 1×10^{-3}, taille de lot effective 32 via accumulation de gradients ;

— **Architecture** : encodeur ULP composé de 4 blocs Transformer, dimension cachée 768, 8 têtes d'attention, dimension de projection 1280 ;

— **Infrastructure** : GPU NVIDIA A100 (80 Go).

Dans MMS, chaque langue l est associée à un adaptateur ϕ_l spécifique. À l'inverse, MonoASR utilise un seul ensemble de paramètres θ pour toutes les langues, conditionné par un **token de langue** $\mathbf{t}_l$. Cette différence structurelle est au cœur de l'analyse.

4.3 Résultats monolingues

Le Tableau 1 présente les performances lorsque chaque langue est entraînée indépendamment.

Analyse. Le WER évalue les erreurs par mot, tandis que BLEU et ROUGE mesurent la similarité via le recouvrement de n-grammes. MonoASR dépasse largement MMS sur le kabyle et le français, montrant qu'un seul espace de paramètres, conditionné par des tokens de langue, capture mieux les spécificités linguistiques que des adaptateurs isolés. Pour l'arabe, les

Langue	Modèle	WER ↓	BLEU ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑
Kabyle	MMS	0.562	0.280	0.691	0.504	0.690
	MonoASR	**0.291**	**0.482**	**0.806**	**0.662**	**0.806**
Arabe	MMS	0.458	0.325	-	-	-
	MonoASR	0.458	0.290	-	-	-
Français	MMS	0.216	0.716	0.928	0.875	0.928
	MonoASR	**0.112**	**0.794**	0.912	0.851	0.911

TAB. 1 – *Résultats monolingues (MMS vs MonoASR). Les scores ROUGE ne sont pas fiables pour l'arabe en raison de la tokenisation.*

deux modèles obtiennent des performances similaires en WER, mais les scores ROUGE sont inexploitables en raison des diacritiques de l'écriture arabe et de l'absence de segmentation explicite, qui compliquent la correspondance des caractères lors de l'évaluation. Le point clé est que MonoASR atteint ces résultats avec **moins de paramètres** : au lieu de réentraîner un adaptateur par langue (MMS), il recycle le même θ guidé par $\mathbf{t}_l$.

4.4 Résultats multilingues simultanés

Lorsque les trois langues sont entraînées conjointement (Tableaux 2, 3), MonoASR obtient en moyenne un WER inférieur de 21.7% à MMS, avec des gains notables en BLEU et ROUGE.

Modèle	**WER ↓**	**BLEU ↑**	**ROUGE-1 ↑**	**ROUGE-2 ↑**	**ROUGE-L ↑**
MMS	0.533	0.276	0.450	0.335	0.450
MonoASR	**0.316**	**0.442**	**0.539**	**0.453**	**0.538**

TAB. 2 – *Résultats multilingues simultanés (moyenne sur toutes les langues).*

Langue	**Modèle**	**WER ↓**	**BLEU ↑**	**ROUGE-1 ↑**	**ROUGE-2 ↑**	**ROUGE-L ↑**
Kabyle	MMS	0.727	0.157	0.593	0.379	0.592
	MonoASR	**0.397**	**0.268**	**0.735**	**0.560**	**0.734**
Arabe	MMS	0.522	0.251	-	-	-
	MonoASR	**0.403**	**0.331**	-	-	-
Français	MMS	0.349	0.421	0.758	0.627	0.758
	MonoASR	**0.147**	**0.728**	**0.881**	**0.800**	**0.881**

TAB. 3 – *Résultats multilingues simultanés détaillés par langue. Les scores ROUGE ne sont pas fiables pour l'arabe en raison de la tokenisation.*

Analyse. Le kabyle et le français bénéficient fortement du partage de paramètres via ULP, tandis que MMS souffre d'une fragmentation induite par les adaptateurs ϕ_l. Pour l'arabe, MonoASR améliore modestement le WER et BLEU, mais les scores ROUGE restent ininterprétables. Ces résultats confirment que les tokens $\mathbf{t}_l$ agissent comme des **vecteurs-routeurs**, évitant les interférences entre langues tout en maximisant le transfert de connaissances.

4.5 Résultats multilingues progressifs

Enfin, nous considérons un scénario incrémental où les langues sont introduites progressivement.

Analyse. MonoASR conserve un avantage net sur MMS dans tous les cas. La clé est que l'introduction d'une nouvelle langue ne nécessite que l'ajout d'un token $\mathbf{t}_l$, alors que MMS doit réentraîner un adaptateur ϕ_l entier, sensible aux faibles volumes de données. Ainsi, MonoASR est naturellement plus extensible et frugal, en particulier dans des scénarios réalistes où de nouvelles langues peu annotées doivent être intégrées.

Langue	Modèle	WER ↓	BLEU ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑
Kabyle	MMS	0.562	0.280	0.691	0.504	0.690
	MonoASR	**0.268**	**0.349**	**0.826**	**0.691**	**0.826**
Arabe	MMS	0.454	0.331	-	-	-
	MonoASR	**0.376**	0.323	-	-	-
Français	MMS	0.215	0.718	0.929	0.877	0.929
	MonoASR	**0.120**	**0.762**	0.900	0.830	0.900

TAB. 4 – *Résultats multilingues progressifs. Les scores ROUGE ne sont pas fiables pour l'arabe en raison de la tokenisation.*

4.6 Résumé des observations

Les résultats expérimentaux mettent en évidence trois points majeurs :

— **Frugalité paramétrique.** MonoASR atteint de meilleures performances que MMS en partageant tous ses paramètres, là où MMS gonfle son architecture avec un adaptateur par langue.
— **Meilleure généralisation.** Les tokens de langue agissent comme des vecteurs-routeurs, permettant de transférer efficacement les connaissances entre langues.
— **Extensibilité.** L'ajout d'une nouvelle langue dans MonoASR n'exige que l'apprentissage d'un token, tandis que MMS nécessite un nouvel adaptateur coûteux et fragile sur peu de données.

En somme, MonoASR établit un compromis favorable entre performance, extensibilité et frugalité, confirmant la pertinence de l'approche ULP pour la RAP multilingue.

5 Analyse qualitative

Nous présentons une analyse qualitative sur des transcriptions en kabyle, arabe et français (voir Figure 2). **MonoASR** se rapproche systématiquement davantage des références que **MMS**. En kabyle, MMS fusionne ou omet des morphèmes, tandis que MonoASR respecte mieux les formes et l'ordre des mots. En arabe, MMS produit des confusions vocaliques, alors que MonoASR reste fidèle à la référence. En français, MMS commet de petites omissions corrigées par MonoASR. Ces observations confirment les résultats quantitatifs : MonoASR réduit les erreurs lexicales et grammaticales, surtout dans les langues faiblement annotées, validant l'efficacité de la Projection Linguistique Universelle (ULP).

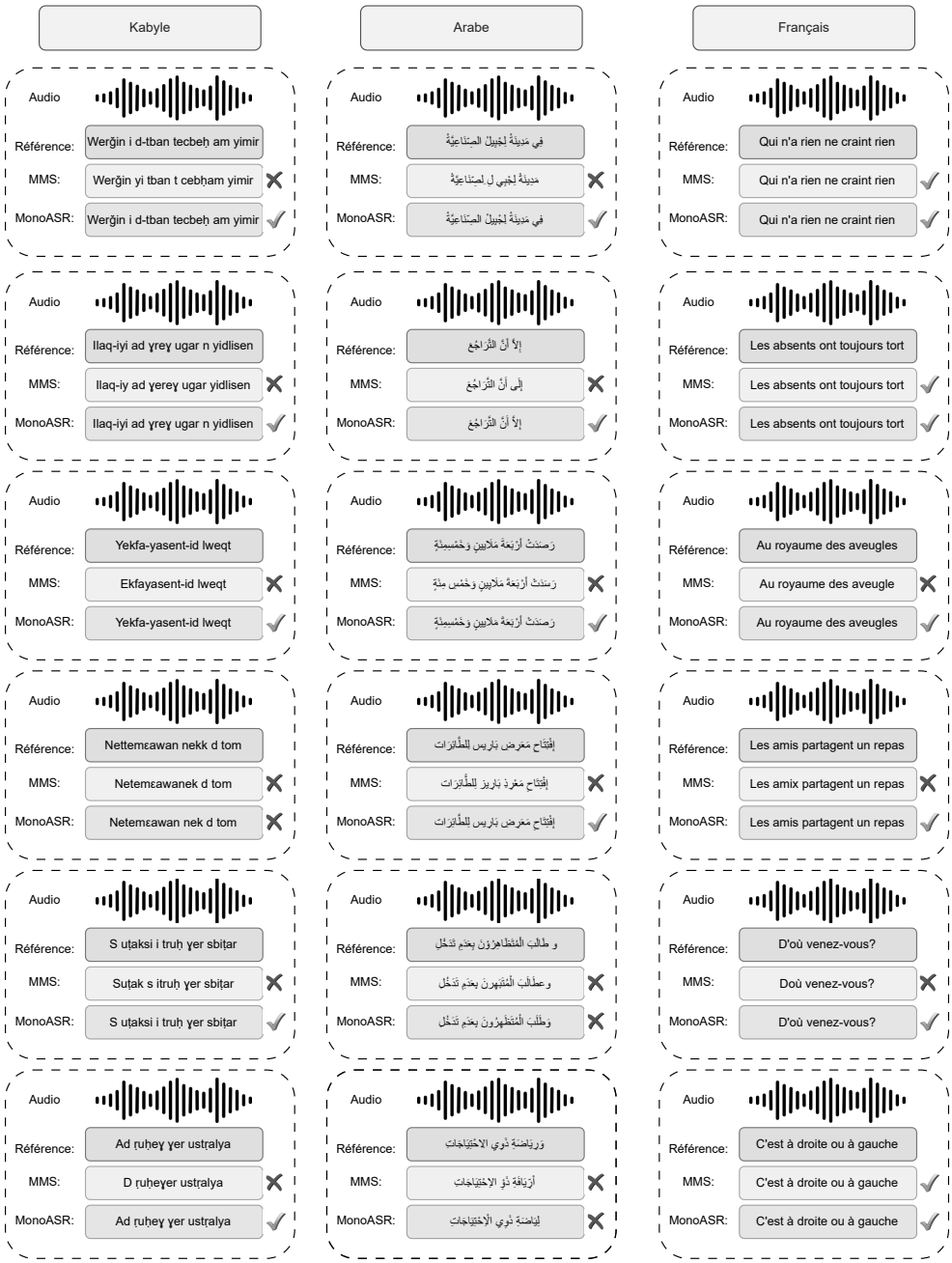

FIG. 2 – *Analyse qualitative : comparaison des transcriptions produites par MMS et MonoASR avec la transcription de référence, c'est-à-dire la version correcte attendue, pour trois langues (kabyle, arabe et français). Cette comparaison illustre les différences de fidélité entre les modèles par rapport au texte de référence.*

6 Conclusion

Dans ce travail, nous avons proposé MonoASR, un modèle de reconnaissance automatique de la parole multilingue reposant sur une architecture unifiée d'encodeur intégrant un mécanisme de Projection Linguistique Universelle (ULP). En conditionnant les représentations audio par des tokens de langue spécifiques, MonoASR parvient à équilibrer efficacement le partage des paramètres et la spécificité linguistique. Nos expériences menées sur trois langues typologiquement diverses — le kabyle (langue amazighe appartenant à la famille des langues nord-africaines), l'arabe et le français — démontrent que MonoASR surpasse systématiquement MMS, en particulier dans les contextes de faibles ressources et d'apprentissage multilingue. Comme perspectives, nous prévoyons d'évaluer MonoASR sur des jeux de données multilingues plus vastes et face à des systèmes de référence plus variés, afin de valider davantage sa capacité à passer à l'échelle. Nous envisageons également d'explorer des stratégies de type « Mélange d'Experts (MoE) » pour mieux intégrer la diversité linguistique, ce qui pourrait renforcer la spécialisation et l'efficacité dans des systèmes de reconnaissance massivement multilingues. Enfin, nous proposons d'investiguer l'utilisation de têtes de modélisation du langage distinctes, chacune adaptée au vocabulaire d'une langue spécifique, plutôt que de recourir à un vocabulaire partagé entre toutes les langues. Cette approche permettrait de réduire l'espace softmax par langue, diminuant ainsi le nombre de scores de probabilité en compétition lors du décodage, ce qui pourrait améliorer les performances et réduire la confusion entre langues.

Références

Baevski, A., Y. Zhou, A. Mohamed, et M. Auli (2020). wav2vec 2.0 : A framework for self-supervised learning of speech representations. *Advances in neural information processing systems 33*, 12449–12460.

Graves, A., S. Fernández, F. Gomez, et J. Schmidhuber (2006). Connectionist temporal classification : labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376.

Lin, J., M. Ge, W. Wang, H. Li, et M. Feng (2024). Selective hubert : Self-supervised pre-training for target speaker in clean and mixture speech. *IEEE Signal Processing Letters*.

Mohamed, Y. (2024). Arabic audio rev3 9643 2021 dataset. `https://huggingface.co/datasets/Yahya-Mohamed/Arabic_Audio_Rev3_9643_2021_Dataset`.

odunola (2024). French audio preprocessed dataset. `https://huggingface.co/datasets/odunola/french-audio-preprocessed`.

Pratap, V., A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, et al. (2024). Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research 25*(97), 1–52.

Radford, A., J. W. Kim, T. Xu, G. Brockman, C. McLeavey, et I. Sutskever (2023). Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR.

Rekesh, D., N. R. Koluguri, S. Kriman, S. Majumdar, V. Noroozi, H. Huang, O. Hrinchuk, K. Puvvada, A. Kumar, J. Balam, et al. (2023). Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE.

Sudo, Y., M. Shakeel, Y. Fukumoto, Y. Peng, et S. Watanabe (2024). Contextualized automatic speech recognition with attention-based bias phrase boosted beam search. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10896–10900. IEEE.

TutlaytAI (2024). Kabyle asr dataset. `https://huggingface.co/datasets/TutlaytAI/kabyle_asr`.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, et I. Polosukhin (2017). Attention is all you need. *Advances in neural information processing systems 30*.

Xue, H., K. Huang, Z. Zhou, S. Huang, et S. Shang (2025). The tea-aslp system for multilingual conversational speech recognition and speech diarization in mlc-slm 2025 challenge. *arXiv preprint arXiv :2507.18051*.

Xue, H., Q. Shao, K. Huang, P. Chen, J. Liu, et L. Xie (2024). Sshr : Leveraging self-supervised hierarchical representations for multilingual automatic speech recognition. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE.

Yan, B., V. Pratap, S. Watanabe, et M. Auli (2025). Improving multilingual asr in the wild using simple n-best re-ranking. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE.

Yu, Z., Y. Zhang, K. Qian, C. Wan, Y. Fu, Y. Zhang, et Y. C. Lin (2023). Master-asr : achieving multilingual scalability and low-resource adaptation in asr with modular learning. In *International Conference on Machine Learning*, pp. 40475–40487. PMLR.

Zhang, B. et R. Sennrich (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems 32*.

Summary

Automatic Speech Recognition (ASR) converts spoken language into text and remains a major challenge. Recent models, such as Massively Multilingual Speech (MMS), cover hundreds of languages but require the addition of language-specific adapters, which increases parameter cost and hinders scalability, especially for low-resource languages. We introduce MonoASR, a frugal and unified multilingual system that avoids such adapters through a Universal Language Projection (ULP). ULP associates a learned language token with acoustic representations, enabling the same model and parameters to handle different languages. Evaluated on French (a high-resource language), Arabic, and Kabyle [2] (underrepresented and complex languages), MonoASR achieves lower word error rates (WER) than MMS, demonstrating its robustness, generalization ability, and suitability for low-cost multilingual transcription. Code is available at : `https://github.com/ilyesqlm/MonoASR`

2. It is one of the Tamazight languages, part of the North African language family.