Actes de CORIA-TALN 2026
Nantes, France, 29 juin au 3 juillet 2026

Calame : Un logiciel de transcription en code source libre

Thomas Soulas Yves Ferstler Valentyna Tsilinchuk Yassine Chahdi
Catherine Lavoie Gaëlle Laperrière Marie-Jean Meurs
Université du Québec à Montréal (UQAM), Montréal, QC, Canada
{soulas.thomas_david,laperriere.gaelle}@courrier.uqam.ca,
meurs.marie-jean@uqam.ca

RÉSUMÉ

Bien que la recherche en traitement automatique de la parole soit très active, ses résultats demeurent peu accessibles aux personnes non-expertes. Les systèmes qui en sont issus ont pourtant de nombreuses applications, telles la transcription de réunions, d'entretiens et de conférences. Les non-spécialistes ont ainsi souvent recours à des solutions commerciales coûteuses, manquant généralement de modularité et fonctionnalités, sans protection des données sensibles et exposées aux cybermenaces. Afin de rendre le traitement automatique de la parole accessible à la fois à la communauté scientifique et au grand public, nous avons lancé Calame : un logiciel gratuit en code source libre, utilisable en local et à distance. Calame permet actuellement de traiter l'anglais et le français de France, le français québécois ainsi que d'autres langues à faibles ressources étant progressivement intégrés grâce à l'affinage de modèles à l'état de l'art.

ABSTRACT

Calame : An Open Source Transcription Software

While research on automatic speech processing is very active, its outcomes remain mainly inaccessible to people without programming skills or expertise. Automatic speech processing systems can be needed in a variety of use cases, such as automatic transcription of meetings, interviews, or even conferences. Non-experts may rely on commercial solutions, but these typically lack modularity, offer only partial functionalities, increase exposure to cyber threats, and impose significant financial barriers for potential users. As automatic transcription techniques improve, it becomes crucial to make these tools accessible to both the research community and the general public. To make language technology more inclusive, we released Calame, a free, open-source, and accessible software for automatic multilingual speech processing, available for both local and remote use. Its current language coverage includes English and French, with Quebec French and other low-resource languages being gradually incorporated with state-of-the-art fine-tuned models.

MOTS-CLÉS : Traitement automatique du langage naturel, traitement automatique de la parole, transcription automatique, code source libre, langues peu dotées, verbatim.

KEYWORDS: Natural Language Processing, Automatic Speech Recognition, Transcription, Free and Open Source, Low-resource languages, Verbatim.

Actes de la session industrielle de CORIA-TALN 2026, pages 2–12.

1 Introduction

Bien que la recherche sur le traitement de la parole soit très active, notamment autour de boîtes à outils en code source libre telles que ESPnet, Kaldi, NeMo et SpeechBrain (Ravanelli *et al.*, 2024; Watanabe *et al.*, 2018; Kuchaiev *et al.*, 2019; Bredin, 2023; Han *et al.*, 2025), les systèmes qui en résultent restent souvent inaccessibles aux personnes n'ayant pas les connaissances nécessaires en programmation ou en traitement du langage naturel (TALN), alors que des solutions commerciales continuent d'être développées pour les langues largement dotées en ressources. Pour remédier à ce manque, nous présentons Calame [1], un outil de transcription et de séparation de locuteurs.trices gratuit, en code source libre, simple d'utilisation, permettant de faciliter l'utilisation de modèles à l'état de l'art en TALN pour des langues bien dotées comme peu dotées.

Cet article présente la première version stable de Calame. Le logiciel a déjà été testé en version bêta et est activement utilisé par des groupes de recherche issus d'environnements et de contextes différents. Travaillant majoritairement en sciences humaines, ces personnes utilisatrices ont besoin de solutions robustes pour produire et gérer des transcriptions mot-à-mot sur des appareils aux ressources parfois limitées. Le développement et la diffusion de l'outil bénéficient par ailleurs du soutien de Calcul Québec [2], qui agit à la fois comme partenaire et comme preneur d'un outil distribué destiné à permettre un déploiement à large échelle. Calame cible les équipes de recherche travaillant avec des données qualitatives devant être transcrites, et ayant un fort besoin d'outils efficaces, durables, sécurisés et faciles d'accès. Calame s'attache de plus à offrir un soutien aux langues disposant de peu de ressources, en commençant par le français québécois.

L'article présente en section 2 les principaux outils logiciels existants comparables à Calame. La section 3 présente Calame et expose les réflexions qui ont guidé sa conception avant de détailler son architecture logicielle. La section 4 décrit l'état actuel du projet et rapporte les résultats expérimentaux obtenus sur l'ensemble de données en français québécois CEREALES (Maison *et al.*, 2025), cette variété de français étant la première langue à faibles ressources prise en compte par le logiciel. Elle présente également les premiers constats des personnes utilisatrices de Calame. Les évolutions envisagées et les perspectives d'avenir de l'outil sont présentées en conclusion.

2 Logiciels de transcription automatique

2.1 Logiciels en code ouvert

À notre connaissance, en mars 2026, les seuls logiciels libres et en code ouvert (FOSS) similaires à Calame sont Amical [3], Vexa [4], Murmure [5] et Scriberr [6].

Amical, encore en début de développement, se focalise sur la prise de notes et utilise des modèles génératifs pour leur correction et amélioration. Vexa est un outil "*freemium*" hébergeable localement, dédié à la transcription de réunions en ligne. Murmure est un outil local développé principalement pour la dictée. Scriberr fournit une large gamme d'outils de TALN, mais ses applications sont limitées à des cas d'utilisation génériques et ne prennent pas en charge les langues à faibles ressources ou

1. https://calame.tech
2. https://www.calculquebec.ca
3. https://github.com/amicalhq/amical
4. https://github.com/Vexa-ai/vexa
5. https://github.com/Kieirra/murmure
6. https://scriberr.app

les domaines spécialisés. Son installation reste également difficile pour les personnes n'ayant pas de connaissance en programmation puisqu'elle se fait par le biais de lignes de commande.

2.2 Logiciels propriétaires

Plusieurs solutions commerciales prêtes à l'emploi sont disponibles pour les particuliers et les entreprises qui ne souhaitent pas développer leur propre système, répondant ainsi à leurs besoins en matière de transcription d'enregistrements vocaux. La table 1 donne un aperçu de ces solutions. Certaines sont facilement accessibles via une interface web, offrant un large éventail de fonctionnalités de correction et d'édition. D'autres sont disponibles via une interface de programmation (API) et nécessitent des compétences minimales en programmation. La plupart d'entre elles proposent un abonnement mensuel et sont limitées à quelques heures de traitement audio par mois, entraînant des frais supplémentaires en cas de seuil atteint.

La table 1 met en évidence ces coûts ainsi que les langues prises en charge. Bien que de nombreuses applications offrent une large couverture multilingue (de 1 à plus d'une centaine de langues), on remarque une surreprésentation des langues grandement dotées ainsi que des systèmes multilingues généralistes ne proposant pas de spécialisation pour les langues couvertes. Cette absence de spécialisation implique des limitations dans le traitement des langues peu dotées, avec une uniformisation selon des bases grammaticales majoritairement anglophones.

De part leur nature, ces outils imposent pour la majorité de transmettre les données à des tiers, les rendant inutilisables pour le traitement de données sensibles, confidentielles ou personnelles.

Logiciel	**Usage**	**Langues**	**Prix**
AmberScript	Web	Français, Anglais, 37 autres	10€ / heure
Authôt	Web	Français, Anglais, 30 autres	0,1€ / min
Descript	Web	Français, Anglais, 20 autres	16-24€ / mois pour 10-30 heures
Dictation	Web	Français, Anglais, 123 autres	gratuit
HappyScribe	Web	Français, Anglais, 64 autres	0,15€ / min
Otter.ai	Web	Anglais	8,33€ / mois
Sonix	Web	Français, Anglais, 40 autres	15€ / mois
Speechmatics	Web	Français, Anglais, 48 autres	0,08€ / min
Temi (Rev)	Web	Anglais	0,25€ / min
Trint	Web	Français, Anglais, 40 autres	80€ / mois
Verbit	Web	Anglais and Spanish	24€ / mois
Vook.ai	Web	Français, Anglais, 4 autres	3€ / heure
AssemblyAI	API	Français, Anglais, 18 autres	0,15€ / heure
Amazon Transcribe	API	Français, Anglais, 123 autres	0,024€ / min
Deepgram	API	Français, Anglais, 30 autres	0,37€ / heure
IBM Watson S2T	API	Français, Anglais, 10 autres	0,02€ / min
Microsoft Azure S2T	API	Français, Anglais, 137 autres	0,485-1,615€ / heure
Google S2T	API	Français, Anglais, 123 autres	0,024€ / min
Whisper	API	Français, Anglais, 100 autres	0,006€ / min
f4x	Desktop	Français, Anglais, 18 autres	15€ / heure
Nuance Dragon	Desktop	Français, Anglais, 13 autres	999€ license
Nvivo Transcription	Desktop	Français, Anglais, 40 autres	30€ / heure

TABLE 1 – Aperçu des applications propriétaires de transcription automatique, mars 2026.

3 Calame

Calame est conçu pour les entretiens réels, les longs enregistrements audio, et offre des fonctionnalités utiles pour divers contextes, notamment dans le domaine universitaire et de la recherche, telles que la séparation de locuteurs.trices et l'anonymisation. Notre travail se concentre sur le développement de solutions ciblées, visant à faire progresser la recherche en TALN pour les langues à faibles ressources et les domaines spécialisés, tout en fournissant un programme d'installation avec interface graphique actuellement disponible pour Windows et Linux (installateur macOS en cours de développement).

Afin que Calame puisse être utilisé pour transcrire des informations confidentielles, nous l'avons conçu pour que les personnes utilisatrices aient le choix entre un traitement des données local ou distant. En raison des possibles contraintes matérielles de nos communautés utilisatrice, Calame fonctionne sur CPU ou GPU et sur la plupart des systèmes d'exploitation (OS), Windows et Linux étant notre priorité.

3.1 Architecture logicielle

Calame est intégré dans un environnement Docker[7], ce qui réduit les coûts de développement et facilite le déploiement multi-OS. Bien que Docker offre une grande commodité, il présente l'inconvénient d'un environnement plus volumineux et plus exigeant que l'exécution native. En effet, Docker introduit plusieurs couches d'abstraction supplémentaires : chaque conteneur embarque non seulement l'application, mais aussi toutes ses dépendances, bibliothèques et fichiers système nécessaires à son fonctionnement.

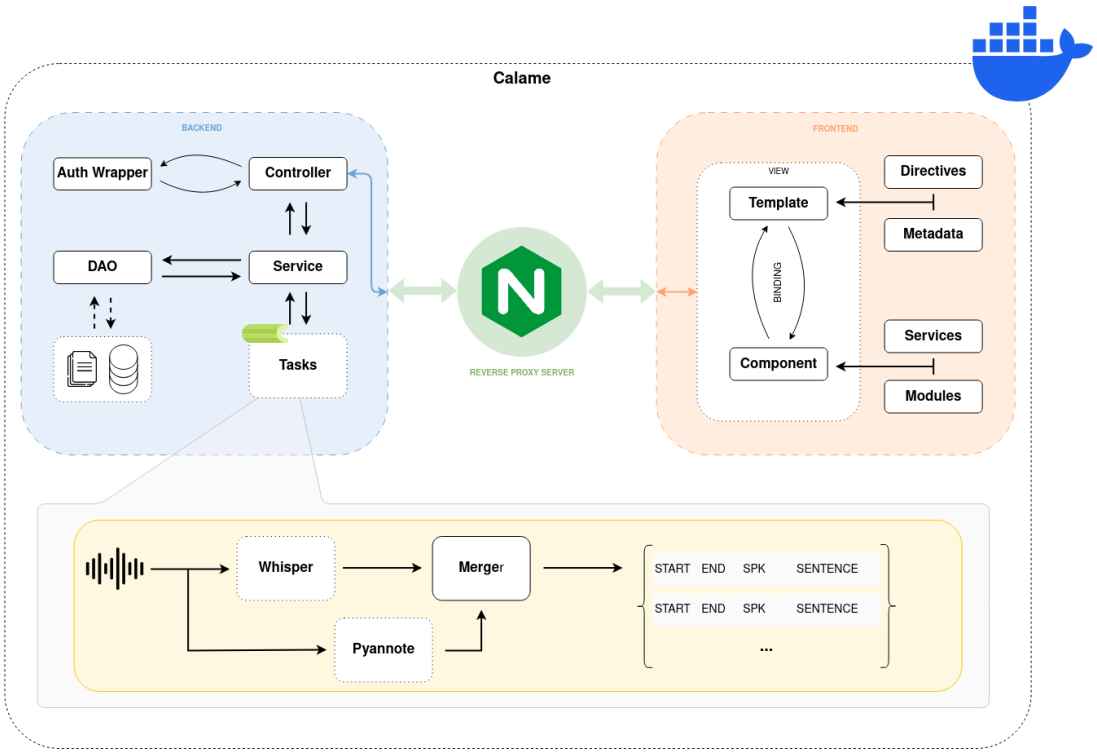

FIGURE 1 – Architecture de Calame.

7. https://www.docker.com/

La figure 1 montre l'architecture complète de Calame, avec l'utilisation de Python pour la partie serveur (*backend*) et d'Angular pour la partie cliente (*frontend*). Même si Python peut être considéré comme plutôt lent pour le déploiement côté serveur, il peut facilement gérer plusieurs centaines de personnes à la fois, tout en permettant une mise en oeuvre aisée des modèles d'apprentissage profond utilisés. Plusieurs instances de Calame peuvent également être déployées pour la gestion d'un plus grand nombre de personnes utilisatrices. Les contraintes liées au CPU et à la mémoire, en particulier la VRAM ou la RAM, constituent les principaux goulots d'étranglement.

La partie serveur est composée de modules qui communiquent entre eux. Le modèle architectural est un modèle classique de contrôleur, service, objet d'accès aux données (DAO) avec une couche d'authentification sur le contrôleur. Les contrôleurs gèrent le routage, les services gèrent les tâches, et les méthodes DAO gèrent les requêtes vers la base de données qui recense toutes les données dynamiques de Calame (personnes utilisatrices, projets, transcriptions, etc.) Pour gérer l'exécution des tâches telles que la transcription ou la séparation de locuteurs.trices, nous utilisons Redis [8] avec Celery [9], des solutions matures permettant la gestion de tâches asynchrones. Le fichier d'environnement actuel permet à la personne utilisatrice de configurer le nombre de tâches pouvant être effectuées en parallèle. Il est recommandé de prévoir 8 Go de RAM ou de VRAM pour une tâche. La partie serveur est également fournie avec une API, ce qui signifie que la partie cliente est interchangeable avec tout autre partie cliente compatible avec cette API.

3.2 Implémentation et fonctionnalités

Calame est avant tout un outil de transcription doté de capacités de séparation de locuteurs.trices. Tout au long de son développement et avant sa première version, nous avons pris la décision d'utiliser des modèles prêts à l'emploi tels que Whisper medium (Radford *et al.*, 2023) pour la reconnaissance automatique de la parole, et Pyannote (Bredin, 2023) 3.1 pour la séparation automatique des locuteurs.trices. Whisper est un encodeur-décodeur Transformer entraîné sur 680 000 heures de données vocales étiquetées. Whisper medium s'est montré être un bon compromis initial, entre simplicité de mise en place et consommation computationnelle (RAM et CPU particulièrement). Sa version large est moins adaptée à des machines ayant peu de ressources computationnelles. Ses taux d'erreur de mots (WER) sur les ensembles de données Common Voice 9 en français et en anglais [10] sont respectivement de $16,0$ et $11,2$.

Pyannote est une boîte à outils pour la séparation de locuteurs.trices, la détection de l'activité vocale et d'autres tâches de segmentation audio. Nous utilisons Pyannote 3.1 pour séparer les segments des locuteurs.trices et relier la transcription correcte au locuteur ou à la locutrice approprié.e. Ses taux d'erreur de séparation (DER) sont rapportés par Bredin (2023) pour les données AMI [11] composées de parole en anglais, avec un DER de $18,8$ pour les données "unique microphone distant" et un DER de $22,7$ pour les données "micro-casque individuel". Bien que Pyannote reste l'un des outils prêt à l'emploi les plus performants pour la séparation de locuteurs.trices dans des langues bien dotées, nous continuons à évaluer et à comparer les méthodes à l'état de l'art récentes pour les prochaines versions de Calame, avec une spécialisation prévue sur les langues peu dotées ciblées par le logiciel.

Le pipeline actuel utilise Whisper et Pyannote séparément. Cette décision permet aux personnes

8. https://redis.io/
9. https://docs.celeryq.dev
10. https://commonvoice.mozilla.org/fr/datasets
11. https://groups.inf.ed.ac.uk/ami/corpus/

utilisatrices de sélectionner soit le module de transcription seul, soit les modules combinés de transcription et de séparation de locuteurs.trices. La tâche de séparation de locuteurs.trices récupère alors la transcription réalisée afin d'attribuer à chaque locuteur ses segments de texte. Dans le contexte de l'application, la tâche de séparation est dépendante de la tâche de transcription.

Les figures 2 et 3 présentent l'interface d'utilisation de Calame, la première figure illustre l'interface de gestion de projets du logiciel, mettant en évidence les principales fonctionnalités offertes à la personne utilisatrice : création et organisation des projets, suppression des projets et accès aux projets. La seconde figure présente quant à elle un exemple concret de transcription automatique appliquée à un discours public en français québécois, accompagné d'une démonstration de la séparation automatique des locuteurs.trices. La personne utilisatrice peut ici modifier manuellement les transcriptions et l'attribution de locuteur.trice après les traitements automatiques.

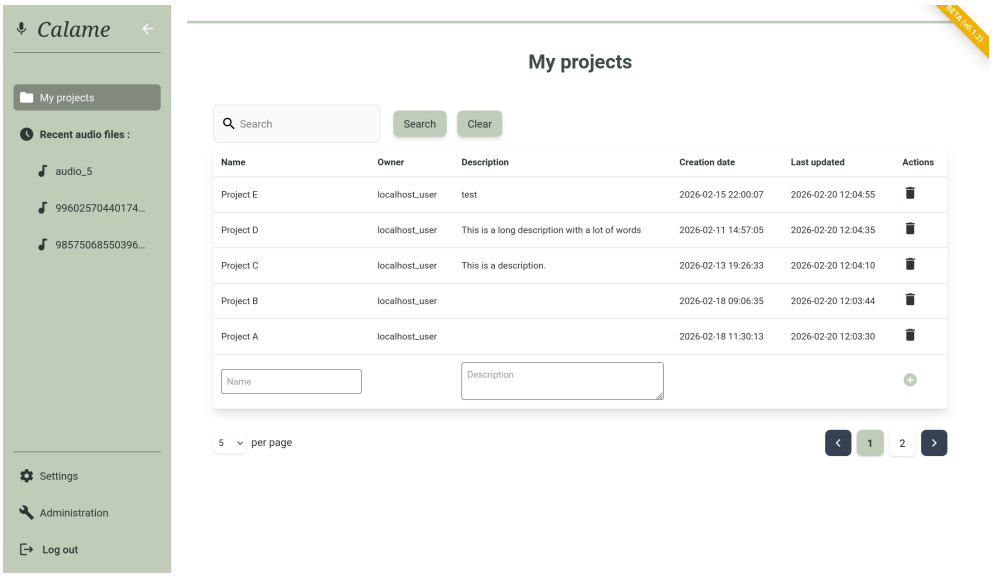

FIGURE 2 – Gestion de projets dans Calame.

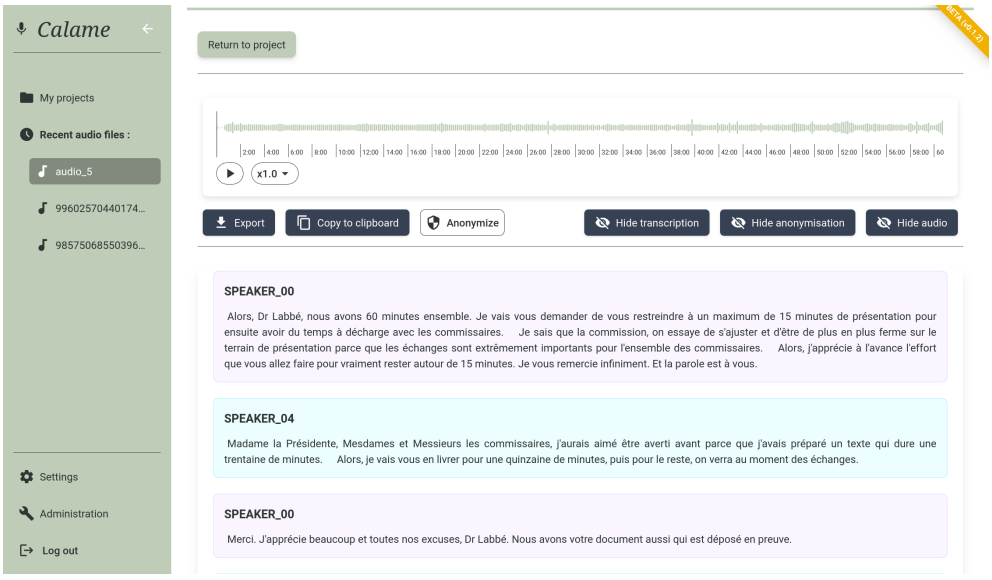

FIGURE 3 – Un fichier transcrit dans Calame.

4 Évaluations qualitative et quantitative

Cette section analyse le temps de traitement des deux tâches principales de Calame : la transcription et la séparation de locuteurs.trices. Elle fournit les résultats expérimentaux de la reconnaissance automatique de la parole (ASR) sur l'ensemble de données CEREALES en français québécois et présente les commentaires qualitatifs des personnes utilisatrices du logiciel.

4.1 Performances de calcul

Calame a été conçu pour pouvoir être déployé sur un large éventail de matériel, dont du matériel aux capacités modestes. Nous avons donc évalué ses performances de calcul avec différentes configurations matérielles :

— RTX 4060 avec 8 Go de VRAM, Ryzen 7 7700 (ordinateur de bureau sous Windows)
— RTX 2070 avec 8 Go de VRAM, i7-10875H (ordinateur portable sous Windows)
— i7-1260P avec 32 Go de RAM (ordinateur portable sous Arch Linux)

La table 2 montre que les temps de traitement sont légèrement inférieurs avec un GPU NVidia RTX 2070 vieux de 7 ans qu'avec notre CPU de 3 ans, mais vraiment lents par rapport à un GPU NVidia RTX 4060 plus récent. Outre les considérations liées à l'âge du matériel, cela peut s'expliquer entre autre par les formats de précision numérique supportés et le comportement arithmétique des Tensor Cores, qui varient selon les générations. En effet, l'architecture Turing (RTX 2070) ne supporte que les formats de précision FP16 et FP32 alors que l'architecture Ada Lovelace supporte FP8, FP16, bfloat16 et TF32. Ces écarts, bien que souvent infimes pour une seule opération, se propagent et s'amplifient à chaque calcul, pouvant aboutir à des sorties numériquement différentes d'une architecture à l'autre, même avec un modèle strictement identique.

Matériel	**Fichier**	t_{TRS}	t_{SEP}	t
i7-1260P	5 min	$5,51$	$3,41$	$\mathbf{8,92}$
	30 min	$26,34$	$22,72$	$\mathbf{49,06}$
	60 min	$51,31$	$39,45$	$\mathbf{90,76}$
RTX 2070	5 min	$3,24$	$0,39$	$\mathbf{3,63}$
	30 min	$24,99$	$4,30$	$\mathbf{29,29}$
	60 min	$38,32$	$13,24$	$\mathbf{51,56}$
RTX 4060	5 min	$1,02$	$0,24$	$\mathbf{1,26}$
	30 min	$3,72$	$1,36$	$\mathbf{5,08}$
	60 min	$6,99$	$2,83$	$\mathbf{9,82}$

TABLE 2 – Temps de traitement (en minutes) pour la transcription (t_{TRS}), la séparation de locuteurs.trices (t_{SEP}) et au total (t) pour des fichiers audio de différentes durées.

L'outil interactif en ligne de commande `htop` montre une augmentation de 6 à 7 Go de l'utilisation de la mémoire vive lorsqu'une tâche est en cours d'exécution. Toutefois, posséder plus de mémoire vive ne permettrait pas nécessairement d'exécuter plus de tâches en parallèle. En effet, dans les configurations évaluées, le processeur présente lui-même un goulot d'étranglement. De ce fait, toutes les tâches sont traitées consécutivement par défaut.

4.2 Résultats expérimentaux

L'ensemble de données CEREALES en français québécois (Maison *et al.*, 2025) provient de la Commission Viens [12] et résulte d'une enquête publique sur les relations entre les peuples autochtones et certains services publics au Québec entre juin 2017 et décembre 2018. Il s'agit d'un vaste corpus équilibré sur le plan du genre, comprenant 346 heures d'enregistrements de discours spontanés provenant d'environ 300 locutrices et locuteurs québécois différents, librement accessible à des fins de recherche académique.

Deux mesures conventionnelles sont utilisées pour les évaluations : le taux d'erreur sur les mots (WER) et le taux d'erreur sur les caractères (CER). Concernant l'intervalle de confiance du WER sur l'ensemble de test de CEREALES, une amélioration est jugée significative lorsque supérieure à $0,5$ points, selon le test de Student. Une variation des taux d'erreur de $0,2$ points a également été observée sur 5 entraînements ASR dû aux initialisations des paramètres neuronaux. Il convient de noter que, comme les processeurs ne peuvent pas utiliser le format à virgule flottante simple précision (FP32), les taux d'erreur peuvent également être plus élevés lors de l'utilisation des modèles d'ASR dans Calame avec un processeur (jusqu'à $0,2$ points de WER d'après nos expériences). Les entraînements ASR présentés dans cette section sont eux effectués sur un GPU A100-40G.

L'affinage d'encodeurs vocaux pré-appris de manière auto-supervisée (SSL) monolingues tels que LeBenchmark (Parcollet *et al.*, 2024) et multilingues tels que XLS-R (Babu *et al.*, 2022) et w2v-BERT 2.0 (Chung *et al.*, 2021) a démontré des résultats de pointe en matière de reconnaissance automatique de la parole (ASR) pour des ensembles de données complexes et pauvres en ressources (Mdhaffar *et al.*, 2024; Conneau *et al.*, 2022). Afin d'améliorer la reconnaissance automatique de la parole en français québécois dans Calame, nous avons affiné ces modèles en complément de Whisper, initialement choisi pour Calame, sur 50 heures de données d'entraînement du jeu de données CEREALES. L'affinage de l'encodeur vocal a été effectué à l'aide de la boîte à outils SpeechBrain, tandis que l'affinage de Whisper a été réalisé avec la bibliothèque Hugging Face [13].

Pour garantir une cohérence lors de l'analyse des résultats expérimentaux, un ensemble identique d'hyper-paramètres a été utilisé pour l'ensemble des expériences intégrant un encodeur de parole. L'encodeur de parole est suivi d'un bloc de décodage composé de 3 couches linéaires de 1024 neurones initialisées aléatoirement et activées avec LeakyReLU, lui-même suivi d'une couche Softmax finale. L'affinage est effectué à l'aide d'une fonction de coût CTC, optimisée par l'optimiseur Adam (Kingma & Ba, 2015) avec un taux d'apprentissage de $0,00001$ pour les encodeurs de parole, et Adadelta (Zeiler, 2012) avec un taux d'apprentissage de $1,0$ pour le bloc de décodage. Les méthodes de traitement par lots dynamiques de SpeechBrain ont été utilisées pour améliorer l'efficacité et la stabilité de l'entraînement sur les segments audio de longueur variable de CEREALES.

La table 3 présente les résultats ASR expérimentaux. Les performances initiales moyennes de Whisper sur l'ensemble de test CEREALES étaient de $19,26$ WER. Le réglage fin de Whisper large-v3 sur un sous-ensemble de l'ensemble d'entraînement CEREALES permet d'atteindre un WER de pointe de **13,38**. Cependant, le CER nettement inférieur de w2v-BERT indique son adéquation pour Calame, améliorant considérablement les performances ASR pour le français québécois de plus de 5 points de WER, avec beaucoup moins de paramètres et donc des temps de traitement équivalents à ceux partagés pour Whisper medium dans la table 2.

12. https://www.quebec.ca/gouvernement/portrait-quebec/premieres-nations-inuits/commission-viens
13. https://huggingface.co

Model	#Param.	CER	WER
LeBenchmark 3k large		7,93	16,22
LeBenchmark 7k large	319M	7,68	15,81
LeBenchmark 14k large		7,55	14,96
XLS-R	319M	8,51	17,79
w2v-BERT 2.0	584M	**6,98**	**13,61**
Whisper medium	769M	9,32	14,61
Whisper large-v3-turbo	809M	8,83	13,81
Whisper large-v3	1,550M	8,68	**13,38**

TABLE 3 – Résultats expérimentaux de l'affinage de modèles d'ASR sur le corpus de test de l'ensemble de données françaises québécoises CEREALES.

4.3 Retours des personnes utilisatrices

Depuis juillet 2025, Calame est partagé avec des communautés de recherche, notamment en sciences humaines, en sciences sociales et en droit. Le logiciel a été testé par plus de 10 personnes utilisatrices aux profils variés, la plupart ne possédant que des compétences informatiques de base (bureautique). Les personnes utilisatrices rapportent que le logiciel réduit la charge de travail manuelle d'un facteur deux à trois, même sur des appareils bas de gamme. Pour un temps de transcription manuelle initialement prévue à 10 heures, cela signifie un temps final de travail rapporté à 5 heures voire 3 heures environ, grâce à la transcription automatique et la séparation de locuteurs.trices réalisées par Calame. La facilité d'utilisation, la synchronisation entre l'audio et la transcription, ainsi que la possibilité de contrôler la vitesse de lecture sont également particulièrement appréciées. Le déploiement de Calame à plus large échelle rend maintenant possible la collecte de commentaires qualitatifs et de données quantitatives sur les gains par rapport aux transcriptions manuelles réalisées par les équipes utilisatrices. Ces informations permettront une évaluation rigoureuse des services rendus par l'outil.

5 Conclusion

Les outils de TALN de pointe sont rarement accessibles aux non-spécialistes. De plus, les études se concentrent principalement sur les langues riches en ressources et les configurations conventionnelles, ce qui empêche une adoption plus large de ces technologies. Les solutions commerciales existantes sont souvent limitées, coûteuses et non respectueuses de la souveraineté des données, ce qui rend indispensable la mise à disposition d'outils conviviaux tant pour les équipes de recherche que pour le grand public. Cet article présente Calame, un logiciel libre et en code source ouvert pour le traitement automatique de la parole multilingue, disponible pour une utilisation locale et à distance. Aujourd'hui, Calame encapsule Whisper medium afin de permettre son usage sur la plupart des machines personnelles récentes. Les premiers comptes-rendus des personnes utilisatrices sont encourageants, de même que les résultats à l'état de l'art de **13,38** WER et **6,98** CER obtenus sur l'ensemble de données CEREALES, récemment introduit pour le français québécois, une langue peu dotée. Des analyses approfondies étendront ces apprentissages sur l'ensemble de données CommissionsQC (Serrand *et al.*, 2025), également introduit récemment pour le français québécois.

La prochaine version stable de Calame permettra de choisir le modèle d'ASR le plus pertinent pour la personne utilisatrice. Cette approche permettra aux personnes utilisatrices d'installer des modules spécifiques en fonction de leurs besoins.

La version communautaire récente de Pyannote, Diarizen (Han *et al.*, 2025) et d'autres approches sont actuellement en cours d'évaluation afin d'améliorer la séparation de locuteurs.trices tout en ciblant le traitement des langues à faibles ressources.

Remerciements

Les travaux de recherche présentés dans cet article ont été rendus possibles grâce au soutien et aux ressources fournis par Calcul Québec et par l'Alliance de recherche numérique du Canada. Nous remercions également le Conseil de recherches en sciences naturelles et génie du Canada (CRSNG) [MJ Meurs, subvention CRSNG # 2025-07163] et le Fonds de recherche du Québec (FRQ) Chaire de recherche du Québec sur la découvrabilité des contenus scientifiques en français [MJ Meurs, subvention # 2025-0QCDM-356468].

Références

Babu A., Wang C., Tjandra A., Lakhotia K., Xu Q., Goyal N., Singh K., von Platen P., Saraf Y., Pino J. *et al.* (2022). XLS-R : Self-supervised cross-lingual speech representation learning at scale. In *Interspeech 2022*. DOI : 10.21437/Interspeech.2022-143.

Bredin H. (2023). pyannote.audio 2.1 speaker diarization pipeline : principle, benchmark, and recipe. In *Interspeech 2023*. DOI : 10.21437/Interspeech.2023-105.

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 *ASRU 2021*, p. 244–250. DOI : 10.1109/ASRU51503.2021.9688253.

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. In *SLT 2022*, p. 798–805. DOI : 10.1109/SLT54892.2023.10023141.

Han J., Landini F., Rohdin J., Silnova A., Diez M. & Burget L. (2025). Leveraging self-supervised learning for speaker diarization. In *ICASSP 2025*. DOI : 10.1109/ICASSP49660.2025.10889475.

Kingma D. P. & Ba J. (2015). Adam : A method for stochastic optimization. In *ICLR 2015*. DOI : 10.48550/arXiv.1412.6980.

Kuchaiev O., Li J., Nguyen H., Hrinchuk O., Leary R., Ginsburg B., Kriman S., Beliaev S., Lavrukhin V., Cook J. *et al.* (2019). NeMo : A toolkit for building AI applications using Neural Modules. *arXiv*. DOI : 10.48550/arXiv.1909.09577.

Maison L., Soulas T. & Meurs M.-J. (2025). CEREALES : a new dataset of Quebec French accented speech with applications to speech recognition. In *Interspeech 2025*, p. 4058–4062. DOI : 10.21437/Interspeech.2025-1934.

Mdhaffar S., Elleuch H., Bougares F. & Estève Y. (2024). Performance analysis of speech encoders for low-resource SLU and ASR in Tunisian dialect. In *ArabicNLP 2024*, p. 130–139. DOI : 10.18653/v1/2024.arabicnlp-1.12.

Parcollet T., Evain S., Nguyen H., Le H., Boito M. Z., Mdhaffar S., Alisamir S., Tong Z., Tomashenko N., Dinarelli M., Parcollet T. *et al.* (2024). Lebenchmark 2.0 : A standardized, replicable and enhanced framework for self-supervised representations of french speech. *Computer Speech & Language*, **86**. DOI : 10.1016/j.csl.2024.101622.

Radford A., Kim J. W., Xu T., Brockman G., McLeavey C. & Sutskever I. (2023). Robust speech recognition via large-scale weak supervision. In *ICML 2023*.

Ravanelli M., Parcollet T., Moumen A., de Langen S., Subakan C., Plantinga P., Wang Y., Mousavi P., Della Libera L., Ploujnikov A. *et al.* (2024). Open-source

conversational AI with Speechbrain 1.0. *Journal of Machine Learning Research*, **25**(333), 1–11. DOI : 10.48550/arXiv.2407.00463.

SERRAND C., MORSLI A. & BOULIANNE G. (2025). Commissionsqc : a québec french speech corpus for automatic speech recognition. In *Interspeech 2025*. DOI : 10.21437/Interspeech.2025-2490.

WATANABE S., HORI T., KARITA S., HAYASHI T., NISHITOBA J., UNNO Y., SOPLIN N. E. Y., HEYMANN J., WIESNER M., CHEN N., RENDUCHINTALA A. & OCHIAI T. (2018). ESPnet : End-to-End Speech Processing Toolkit. In *Interspeech 2018*. DOI : 10.21437/Interspeech.2018-1456.

ZEILER M. D. (2012). Adadelta : An adaptive learning rate method. *arXiv*.