Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

Calame: An Open Source Transcription Software

Thomas Soulas*, Yves Ferstler, Valentyna Tsilinchuk, Yassine Chahdi,
Catherine Lavoie, Gaëlle Laperrière*, Marie-Jean Meurs*
Université du Québec à Montréal (UQAM), Montréal, QC, Canada

Abstract

While research on automatic speech processing is very active, its outcomes remain mainly inaccessible to people without programming skills or expertise. Moreover, studies focus mostly on high-resource languages and conventional setups, preventing a wider adoption and social impact of these technologies. Automatic speech processing systems can be needed in a variety of use cases, such as automatic transcription of meetings, interviews, or even conferences. They can also be useful for subtitling and dictation, or to interact with voice assistants. 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.

Keywords: Natural Language Processing, Automatic Speech Recognition, Transcription, Free and Open Source, Low-resource Languages, Verbatim

1. Introduction

Although automatic speech processing is a highly active area of research, with powerful open-source toolkits such as ESPnet, Kaldi, NeMo and SpeechBrain [1–5] available, resulting systems often remain inaccessible to users without necessary programming or Natural Language Processing (NLP) knowledge, while the accessible ones are mostly tailored to mainstream applications and languages with abundant resources. In response to this, we introduce a user-friendly pipeline that simplifies the use of state-of-the-art NLP models for both high and low-resource languages: Calame[1], a transcription and diarization Graphical User Interface (GUI) dedicated to research communities with or without technical skills.

This paper presents the first official release of Calame. The software has already been beta tested and actively used by more than ten people from different environments and contexts, primarily collaborative researchers seeking robust solutions to produce and manage verbatim transcripts on resource-constrained devices. Target users include all research teams working with qualitative data, who require secured, efficient, sustainable and user-friendly tools to handle transcription workflows. Calame further focuses on offering support for low-resource languages, starting with Quebec French.

Section 2 presents a literature review of manual annotation techniques, as well as the dominant commercial solutions on the market. In section 3, we introduce Calame and share the reflections that shaped its conception. We take a closer look at the software architecture, the strategic decisions made during its design and development processes. In section 4, we present the current status of the project, with experimental results on the Quebec French dataset CEREALES [6] – this dialect being the first low-resource language considered for the software – before sharing early users feedback from Calame targeted audience. Finally, we outline the evolutions envisaged for Calame, highlighting the tool future prospects.

[1] https://calame.tech

* soulas.thomas_david@courrier.uqam.ca, laperriere.gaelle@courrier.uqam.ca, meurs.marie-jean@uqam.ca

2. Related Work

As of February 2026 and to the extent of our knowledge, the only Free and Open Source (FOSS) software similar to Calame are Amical[2], Vexa[3], Murmure[4] and Scriberr[5]. Amical is still in the early stages of development and focuses on note taking, using generative models to improve said notes. Vexa is a freemium self-hostable tool dedicated to online meeting transcription. Murmure is a local tool developed mainly for dictation. Scriberr provides a wide range of NLP tools, with state-of-the-art systems. However, its applications are limited to generic use cases and do not support research on low-resource languages or specialized domains. Its installation also remains challenging for non-programmers, since no graphical interface is available to replace terminal commands. Like Amical, Calame has additional modules in development, with a stable version ready for practical use. It is designed for real life interviews, long audios, and provides useful features for various contexts, such as diarization and anonymization. In contrast with Scriberr, our work focuses on developing more targeted solutions, aiming to advance NLP research for low-resource languages and specialized domains, while also providing a guided GUI installer currently available for Windows and Linux (macOS installer in development). Appendix Table 3 summarizes NLP and software features of Calame, Scriberr, Vexa, Murmure, and Amical.

2.1. Commercial Automatic Transcription Softwares

Several ready-to-use solutions are available for both individuals and companies that do not wish to develop their own system, meeting their needs to transcribe speech recordings. Some of these solutions are easily accessible via a user-friendly web interface, offering a wide range of correction and editing features. Others are available via an Application Programming Interface (API) and require minimal programming skills. Most of them propose a monthly subscription and are limited to a few hours of audio processing per month. Once the threshold is reached, usage incurs additional charges. Appendix Table 4 summarizes these applications (web, API and desktop usage), highlighting pricing and supported languages (from 1 to 139). While many applications claim broad multilingual coverage, this support often focuses primarily on high-resource languages or relies on generalized multilingual systems that lack language-specific specialization.

3. Calame Overview

Since Calame might be used to transcribe confidential information, we wanted the user to have the choice between local or remote data processing. Specific development considerations emerged from this decision, given that Calame would have been available for direct use and configurable to a certain extent. In addition, Calame needed to be able to work on CPU or GPU and on most Operating Systems (OS), Windows and Linux being our top priority.

3.1. Software Architecture

To alleviate development cost and facilitate implementation, Calame is bundled in a Docker environment, which alleviates development costs and facilitate multi-OS deployment. While Docker offers convenience, it does come with the trade-off of a larger, more demanding environment compared to native execution.

Figure 1 shows the complete architecture of Calame, with the use of Python for the backend and Angular for the frontend. Even if Python can be considered rather slow for backend deployment, it can easily handle more than 100 people at a time, while allowing

[2] https://github.com/amicalhq/amical
[3] https://github.com/Vexa-ai/vexa
[4] https://github.com/Kieirra/murmure
[5] https://scriberr.app

easy implementation of deep learning solutions. CPU and Memory constraints, particularly VRAM or RAM, are expected to be the main bottlenecks.

The backend is constituted of modules that communicate between each other. The architectural pattern is a typical Controller, Service, Data Access Object (DAO) with an authentication wrapper on the controller. Controllers handle routing, services handle tasks and DAO methods handle queries to the database. To execute tasks such as transcription or diarization, we use Redis[6] with Celery[7] to avoid overloading the machine. The current environment file allows the user to configure the number of tasks that can be done in parallel. We recommend 8GB RAM or VRAM for one task. The backend also comes with an API, meaning that the frontend is swappable with any other frontend that fit with the API.

The frontend is a typical Angular application accessible from any browser. A proxy is bundled with the software to redirect each request. For example, `https://localhost` will connect to the application, and `https://localhost/api` will redirect to the API.

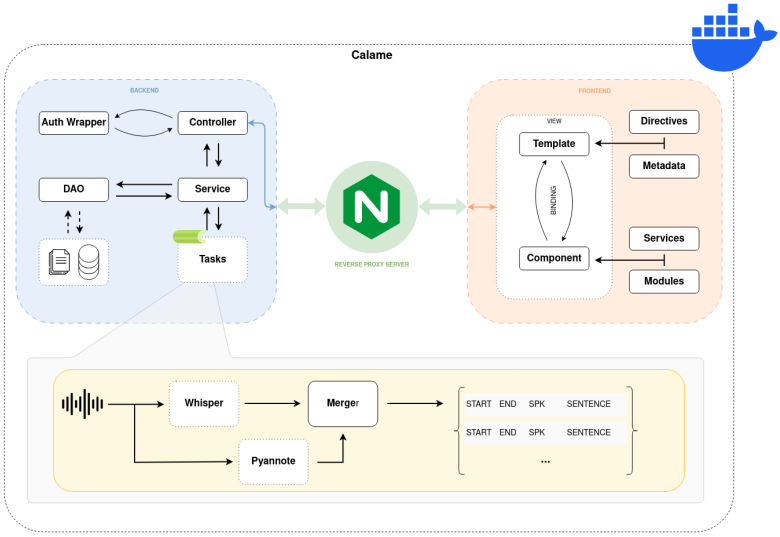

Figure 1. Calame architecture.

3.2. **Features and Implementation**

Calame is primarily a transcription tool with diarization capabilities. Throughout its development and prior to its first release, we took the decision to use former state-of-the-art and ready-to-use models such as Whisper medium [7] for automatic speech recognition, and Pyannote [4] 3.1 for automatic speaker diarization. Whisper is an encoder-decoder Transformer trained on 680k hours of labeled speech data. Its performances on French and English Common Voice 9 [8] dataset are respectively of 16.0 WER and 11.2 WER.

Pyannote is a toolkit for speaker diarization, voice activity detection, and other audio segmentation tasks. We use Pyannote 3.1 to separate speakers segments and link the right transcription to the right speaker. Its Diarization Error Rates (DER) are reported by Bredin [4] for the AMI [9] English corpus, with 18.8 DER on "Single Distant Microphone" data, and 22.7 DER on "Individual Headset Microphone" data. Although Pyannote remains one of the top-performing toolkits for ready-to-use diarization, we continue to evaluate and benchmark emerging state-of-the-art methods for the next release.

The current pipeline uses Whisper and Pyannote separately, each task being in its own environment. This decision allows users to select either the transcription module alone,

[6] https://redis.io/
[7] https://docs.celeryq.dev
[8] https://commonvoice.mozilla.org/fr/datasets
[9] https://groups.inf.ed.ac.uk/ami/corpus/

or the combined transcription and diarization modules. Therefore, the diarization task retrieves the preceding transcription to attribute each speaker to their correct text segments. If no transcription has been previously made, launching the diarization task queues up a transcription task, meaning that the diarization task is dependent on transcription.

Figures 2 and 3 present Calame UI, with the first figure showcasing project management in the software, and the second showcasing an example of automatic transcription and diarization on Quebec French speech.

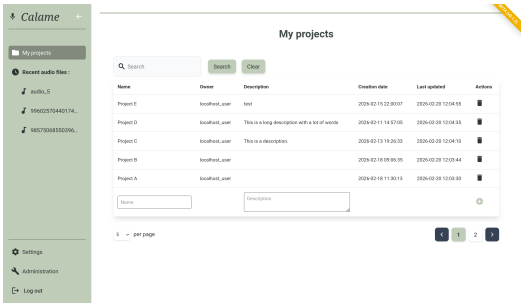

Figure 2. Project management in Calame.

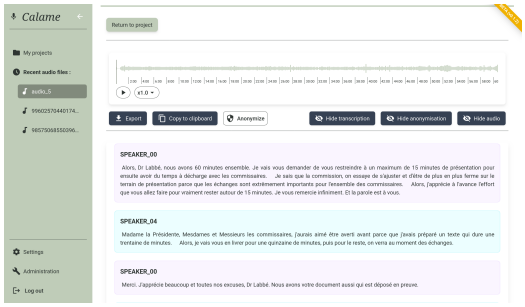

Figure 3. A transcribed file in Calame.

4. Qualitative and quantitative evaluation

This section analyses the processing time of Calame two main tasks: transcription and diarization, sharing experimental ASR results on the Quebec French CEREALES dataset, and discussing qualitative feedback from beta users.

4.1. Computational performance

For Calame to be deployed on a broad spectrum of hardware, it is crucial to evaluate its computational performances on different configurations. As such, we considered the following:

- RTX 4060 with 8GB VRAM, Ryzen 7 7700 (Desktop under Windows)
- RTX 2070 with 8GB VRAM, i7-10875H (Laptop under Windows)
- i7-1260P with 32GB RAM (Laptop under Arch Linux)

Table 1 shows that processing times are slightly lower with a 7 years old RTX 2070 GPU than a CPU, but really slow compared to a more recent RTX 4060 GPU. Apart from considerations related to the hardware age, this can be explained by the GPU architectures (respectively Turing and Ada Lovelace).

Hardware	File	t_{TRS}	t_{DIA}	t
i7-1260P	5 min	5.51	3.41	**8.92**
	30 min	26.34	22.72	**49.06**
	60 min	51.31	39.45	**90.76**
RTX 2070	5 min	3.24	0.39	**3.63**
	30 min	24.99	4.30	**29.29**
	60 min	38.32	13.24	**51.56**
RTX 4060	5 min	1.02	0.24	**1.26**
	30 min	3.72	1.36	**5.08**
	60 min	6.99	2.83	**9.82**

Table 1. Processing time (minutes) for transcription (t_{TRS}), diarization (t_{DIA}), and in total (t) for audio files of different durations.

A quick look to `htop` shows an increase of 6 to 7 GB of RAM usage when a task is in progress. It is, however, not possible to run tasks in parallel with sufficient RAM. Due to a CPU bottleneck in the evaluated configurations, all tasks are treated consecutively by default, preventing a consequent augmentation of the processing time.

4.2. **ASR experimental results**

The Quebec French CEREALES dataset [6] originates from the Commission Viens[10] data, resulting from a public inquiry on relations between Indigenous Peoples and certain public services in Quebec between June 2017 and December 2018. It is a large, gender-balanced corpus of 346 hours of spontaneous speech, from approximately 300 different Quebec French speakers, which is freely accessible for academic research. Two conventional metrics are considered for the evaluations : Word Error Rate (WER) and Character Error Rate (CER). A relevant WER improvement on CEREALES test set should differ by 0.5 points, on top of a possible 0.2 variation of Error Rates observed with 5 ASR trainings. Note that since CPUs cannot use single-precision floating-point format (FP32), the Error Rates can also be higher using a CPU (up to 0.2 WER points from our experimentation), while trainings are performed on a single A100-40G GPU.

Fine-tuning monolingual self-supervised learning (SSL) speech encoders like LeBenchmark [8] and multilingual ones like XLS-R [9] and w2v-BERT 2.0 [10] have led to state-of-the-art ASR results for challenging low-resource datasets [11, 12]. To improve Quebec French ASR in Calame, we fine-tuned these models and Whisper, initially chosen for Calame, on 50 hours of CEREALES training set. Speech encoder fine-tuning was performed using the SpeechBrain toolkit, while Whisper fine-tuning was carried out with Hugging Face library[11].

For comparison consistency, we used an identical set of hyper-parameters for all experiments with speech encoders. Decoding layers are composed of 3 randomly initialized fully-connected layers of 1024 neurons, activated with LeakyReLU, with a final Softmax layer. The fine-tuning is done using a CTC loss function, optimized by Adam with 0.00001 learning rate for speech encoders, and Adadelta with 1.0 learning rate for linear layers. Dynamic Batching methods from SpeechBrain were used to improve training efficiency and stability on CEREALES variable-length audio segments. Table 2 presents ASR experimental results. Whisper medium initial performances on CEREALES test set were of 19.26 WER. Fine-tuning Whisper large-v3 on a subset of CEREALES training set achieves a state-of-the-art **13.38** WER. However, w2v-BERT far better CER indicates its suitability for Calame, improving significantly ASR performances for Quebec French of more than 5 WER points, with a lot less parameters and therefore processing times equivalent to the ones shared for Whisper medium in Table 1.

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 2. Experimental results of low-resource fine-tuning of ASR speech encoders on Quebec French CEREALES test set.

4.3. **User feedback**

Since July 2025, Calame has been shared with research communities including Humanities, Social Sciences, and Law. The software reached over 10 beta users, some with only basic computer skills. Users report that the software reduces manual workload by a time factor of two to three, even on low-end devices. Ease of use, synchronization between audio and

[10] https://www.quebec.ca/gouvernement/portrait-quebec/premieres-nations-inuits/commission-viens

[11] https://huggingface.co

transcription, as well as the ability to control the playback speed are especially welcomed by the target users.

5. Conclusion

State-of-the-art NLP tools are rarely accessible to non-experts. Moreover, studies mostly focus on high-resource languages and conventional setups, preventing their wider adoption. Existing commercial solutions are often limited and costly, making it essential to provide user-friendly tools for both researchers and the general public. This paper presents Calame, a "Free and Open-Source" 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. First impressions and results are encouraging, with 13.38 WER and 6.98 CER state-of-the-art contributions to the recently introduced low-resource Quebec French CEREALES dataset. For future releases, Calame is set to offer a modular architecture, enabling the possibility to switch between models and functionalities. This approach will allow users to install specific modules depending on their needs. The recent community version of Pyannote, Diarizen [5], and other newly state-of-the-art approaches are currently being evaluated to enhance speaker diarization while targeting low-resource language processing through fine-tuning.

Acknowledgements

This research was enabled by support provided by Calcul Québec and the Digital Research Alliance of Canada. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [MJ Meurs, NSERC Grant # 2025-07163] and the Fonds de recherche du Québec (FRQ) Chaire de recherche du Québec sur la découvrabilité des contenus scientifiques en français [MJ Meurs, Grant # 2025-0QCDM-356468].

References

[1] M. Ravanelli et al. "Open-source conversational AI with Speechbrain 1.0". In: *Journal of Machine Learning Research* 25.333 (2024), pp. 1–11. DOI: 10.48550/arXiv.2407.00463.

[2] S. Watanabe et al. "ESPnet: End-to-End Speech Processing Toolkit". In: *Interspeech 2018*. 2018. DOI: 10.21437/Interspeech.2018-1456.

[3] O. Kuchaiev et al. "NeMo: A Toolkit for Building AI Applications Using Neural Modules". In: *arXiv* (2019). DOI: 10.48550/arXiv.1909.09577.

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

[5] J. Han et al. "Leveraging self-supervised learning for speaker diarization". In: *ICASSP 2025*. 2025. DOI: 10.1109/ICASSP49660.2025.10889475.

[6] L. Maison et al. "CEREALES : a new dataset of Quebec French accented speech with applications to speech recognition". In: *Interspeech 2025*. 2025, pp. 4058–4062. DOI: 10.21437/Interspeech.2025-1934.

[7] A. Radford et al. "Robust Speech Recognition via Large-Scale Weak Supervision". In: *ICML 2023*. 2023. URL: https://dl.acm.org/doi/10.5555/3618408.3619590.

[8] T. Parcollet et al. "LeBenchmark 2.0: A standardized, replicable and enhanced framework for self-supervised representations of French speech". In: *Computer Speech & Language* 86 (2024). DOI: 10.1016/j.csl.2024.101622.

[9] A. Babu et al. "XLS-R: Self-supervised cross-lingual speech representation learning at scale". In: *Interspeech 2022*. 2022. DOI: 10.21437/Interspeech.2022-143.

[10] Y.-A. Chung et al. "W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training". In: *ASRU 2021*. 2021, pp. 244–250. DOI: 10.1109/ASRU51503.2021.9688253.

[11] S. Mdhaffar et al. "Performance Analysis of Speech Encoders for Low-Resource SLU and ASR in Tunisian Dialect". In: *ArabicNLP 2024*. 2024, pp. 130–139. DOI: 10.18653/v1/2024.arabicnlp-1.12.

[12] A. Conneau et al. "FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech". In: *SLT 2022*. 2022, pp. 798–805. DOI: 10.1109/SLT54892.2023.10023141.

Appendix A. **FOSS feature comparison**

Software	NLP features				Software features				Free
	Transcription	Diarization	Post-processing	Lang. Coverage	Local&Distant	Record&Upload	User-friendly*	Systems	
Amical	✔		✔	generic multilingual	✔	Record	✔	MacOS, Windows	Freemium
Vexa	✔	✔		generic multilingual	✔	Record	Shell install	Built-in	Freemium
Murmure	✔	✔	✔	generic multilingual	Local	Record	Shell install	MacOS, Linux, Windows	✔
Scriberr	✔	✔	✔	generic multilingual	Self-hostable	✔	Shell install	MacOS, Linux	✔
Calame	✔	✔		fine-tuning low-resource	✔	✔	✔	Linux, Windows	✔

Table 3. Non-exhaustive automatic NLP and software features comparison of Free and Open Source (FOSS) Automatic Transcription Softwares, April 2026. Does not include future or incoming features. (*refers to general public without computation skills)

Appendix B. **Proprietary softwares evaluation**

Software	Usage	Languages	Pricing
AmberScript	Web	French, English, 37 others	10$ / hour
Authôt	Web	French, English, 30 others	0.1$ / min
Descript	Web	French, English, 20 others	16-24$ / month for 10-30 hours
Dictation	Web	French, English, 123 others	free
HappyScribe	Web	French, English, 64 others	0.15$ / min
Otter.ai	Web	English	8.33$ / month
Sonix	Web	French, English, 40 others	15$ / month
Speechmatics	Web	French, English, 48 others	0.08$ / min
Temi (Rev)	Web	English	0.25$ / min
Trint	Web	French, English, 40 others	80$ / month
Verbit	Web	English and Spanish	24$ / month
Vook.ai	Web	French, English, 4 others	3$ / hour
AssemblyAI	API	French, English, 18 others	0.15$ / hour
Amazon Transcribe	API	French, English, 123 others	0.024$ / min
Deepgram	API	French, English, 30 others	0.37$ / hour
IBM Watson S2T	API	French, English, 10 others	0.02$ / min
Microsoft Azure S2T	API	French, English, 137 others	0.485-1.615$ / hour
Google S2T	API	French, English, 123 others	0.024$ / min
Whisper	API	French, English, 100 others	0.006$ / min
f4x	Desktop	French, English, 18 others	15$ / hour
Nuance Dragon	Desktop	French, English, 13 others	999$ license
Nvivo Transcription	Desktop	French, English, 40 others	30$ / hour

Table 4. Overview of proprietary Automatic Transcription Softwares, October 2025 (prices in USD).