

ÚFAL-PMCG SUBMISSION TO THE MIREX 2026 AUTOMATIC LYRICS TRANSCRIPTION CHALLENGE: SIMPLE ASR BEATS AUDIO-LLMS

Tomáš Sourada Jan Hajič jr.
Institute of Formal and Applied Linguistics
Charles University
Prague, Czech Republic
{sourada,hajicj}@ufal.mff.cuni.cz

ABSTRACT

We describe the ÚFAL-PMCG submission to the MIREX 2026 Automatic Lyrics Transcription task. We do not train or fine-tune anything. Instead, we compare two pretrained ASR models, Whisper and Qwen3-ASR, with open-weight audio large language models (audio-LLMs) that fit into 32 GB of GPU memory. On a 500-song multilingual development subset of DALI, Qwen3-ASR has the lowest word error rate (28.9%). The best audio-LLM system, Gemma 4 E4B given the ASR transcripts as hints, reaches 33.8% and stays behind both ASR models. We submit four systems: Whisper, Qwen3-ASR, Gemma 4 E4B with ASR hints, and a combination in which a text-only LLM (Qwen3-8B) merges the Whisper and Qwen3-ASR transcripts clip by clip and then inserts section breaks. The combination’s WER (29.7%) is close to that of Qwen3-ASR, and it has the best section-break F-score.

1. INTRODUCTION

This extended abstract describes the systems submitted by the Prague Music Computing Group (PMCG) at the Institute of Formal and Applied Linguistics (ÚFAL), Charles University, to the MIREX 2026 Automatic Lyrics Transcription (ALT) task.

Cířka et al. [1] showed that a general-purpose ASR model, Whisper [2], transcribes lyrics fairly well without any adaptation. Audio-LLMs have many more parameters and are trained on a wide range of audio tasks, so one might expect them to do better. The only evaluation of open-weight audio-LLMs on lyrics transcription we know of, CMI-Bench [3], found all tested models unusable, but it covered older models. We test recent open-weight audio-LLMs against Whisper and Qwen3-ASR [4], all used off the shelf. In our development experiments, the ASR models outperform the audio-LLMs. Merging the two ASR transcripts with a text-only LLM gives a WER close to the

best single ASR model and the best section structure. We submit the two ASR models, their LLM combination, and the best audio-LLM system.

2. TASK AND EVALUATION

The input is a polyphonic music recording (44.1 kHz stereo WAV); the output is the lyrics as plain text, organized into lines, with an empty line between sections. The test sets are Jam-ALT [1] (79 songs in English, French, German, and Spanish, revised from the multilingual version [5] of JamendoLyrics [6]), MUSDB-ALT [7] (39 English songs from MUSDB18 [8]), and an undisclosed set of 10 English songs. Any training data not overlapping with the test sets may be used (DALI [9] is recommended); each submission must run within 24 h on a single GPU with at most 32 GB of VRAM. The main criteria are WER and CER. Following Cířka et al. [1], the `alt-eval` toolkit [10] reports WER, the case-sensitive WER’, and F-scores for punctuation (P), parentheses (B), line breaks (L), and section breaks (S); it does not compute CER, so we compute CER with our own code. We select systems by WER and CER; the other metrics are less reliable on our development data (Section 3).

3. DATA

We do not train or fine-tune any model, so we use no training data. The pretrained models we use were trained on large amounts of data, which are not published.

We use DALI [9], the recommended training set of the task, for development. Its version 1.0 contains 5,358 commercial songs with semi-automatically aligned lyrics (paragraphs, lines, words, and notes) and a link to a matching YouTube video for each song. DALI’s download tool relies on the defunct `youtube_dl`, so we downloaded the audio with `yt-dlp` (best audio, converted to 192 kbps MP3) for the 5,086 entries marked as working, one at a time with randomized delays, logging failures without stopping. We obtained 4,280 songs (84%); the other 806 are no longer available (deleted, private, terminated account, copyright takedown, or region or age restriction). We exported the annotations of all 5,358 songs with the DALI package (v1.1) into plain-text lyrics files (lines sorted by start time, blank lines between paragraphs)



© T. Sourada and J. Hajič jr.. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** T. Sourada and J. Hajič jr., “ÚFAL-PMCG Submission to the MIREX 2026 Automatic Lyrics Transcription Challenge: Simple ASR Beats Audio-LLMs”, extended abstract in *MIREX 2026 Automatic Lyrics Transcription Task*, 2026.

and JSON files readable without the package, and built a per-song index (artist, title, language, YouTube id, availability, and word, line, and paragraph counts). The lyrics cover 30 languages: English 75%, German 8.1%, French 4.0%, Italian 3.6%, Spanish 3.5%.

We estimate the ratio of English, French, German, and Spanish songs in the final MIREX ALT test sets as roughly 69:20:20:20 and, from the 4,280 songs in DALI with both audio and lyrics, drew two subsets with this ratio, based on the language label of the DALI index (we did not check the actual language of the lyrics). Songs are shuffled per language with a fixed seed, so the smaller subset is contained in the larger one: *multiling25* (13 English, 4 French, 4 German, 4 Spanish; 1.6 h of total audio) for preliminary experiments, and *multiling500* (268/78/77/77; 31.6 h) for the final comparison. The DALI lyrics are all lowercase, so case-sensitive metrics are not meaningful on these subsets.

4. SYSTEM DESCRIPTIONS

Given the limit of 32 GB of GPU memory, we use Whisper [2] (as Cífka et al. [1]) and Qwen3-ASR [4], and the audio-LLMs Qwen2-Audio-7B-Instruct [11], Qwen2.5-Omni-3B and -7B [12], and Gemma 4 E2B, E4B, and 12B [13] (larger Gemma 4 models have no audio input). All systems first run Whisper on the whole song; the others reuse its timestamped segments to cut the song into clips of at most 30 s. Post-processing strips whitespace, drops lines of music symbols only (such as “♪”), and collapses multiple blank lines; casing, punctuation, and parentheses are not normalized.

4.1 ASR-Based Systems

Whisper. We run Whisper large-v2, which Cífka et al. [1] found better than v3 on Jam-ALT with the language provided (27.9% vs. 32.6% WER), with faster-whisper [14] in float16. As Cífka et al. [1], we use long-form transcription with beam size 5 and a line break between timestamped segments; unlike them, we detect the language from the first 30 s. Conditioning on the previous text caused catastrophic failures in preliminary experiments: once a 30 s window was decoded as “♪”, the symbol propagated through the rest of the song, and some songs were transcribed as “♪” only. We therefore disable it. Lines of music symbols only, which Whisper still occasionally outputs, add spurious line breaks, so post-processing drops them.

Qwen-ASR. We run Qwen3-ASR-1.7B with Hugging Face transformers (bfloat16, greedy decoding) on clips cut in the middle of the gaps between Whisper segments, detecting the language per clip. As it outputs no line breaks, we split each clip’s text into as many lines as there are Whisper segments in the clip, in proportion to their word counts.

Whisper+Qwen-ASR+Qwen3-8b. For each clip, a text-only LLM, Qwen3-8B [15] (chosen for its memory footprint; greedy, thinking off), receives the Whisper and

Qwen3-ASR transcripts, the detected language, and all previously combined lines. It picks the more likely reading of each part or combines the two, using only words from the candidates and correcting only clearly misheard words; empty, truncated, or overly long answers fall back to Qwen3-ASR. Then the same LLM receives the numbered lines of the whole song and names the lines that end a section, where we insert blank lines. This improves the section-break F-score and leaves the words, and hence WER, unchanged. Unlike LyricWhiz [16], which combines several Whisper runs, we combine two different ASR systems.

4.2 Audio-LLM-Based Systems

Most open-weight audio-LLMs accept at most 30 s of audio, so the audio-LLM transcribes each clip (consecutive Whisper segments merged up to 30 s) separately, following the instructions of Cífka et al. [1, Sec. 2], with greedy decoding. In preliminary experiments on *multiling25*, adding the text of previous clips as context hurt Gemma 4 E4B, so we do not use it. Against repetition loops, we apply a repetition penalty to the tokens generated for the current clip; of the values 1.05 to 1.5, we chose 1.1. A final text-only pass to fix inconsistencies did not help in a single trial. Optionally, the audio-LLM also gets the Whisper and Qwen3-ASR transcripts of the clip as two unnamed drafts, similar to the multi-ASR fusion of Prakash et al. [17] on speech.

In preliminary experiments (mostly on *multiling25*), Qwen2.5-Omni-7B and Gemma 4 E4B were the best, while the other models reached WER above 100% and seemed to need further prompt engineering. In the final setup (no context, penalty 1.1), Qwen2.5-Omni-7B (46.1% WER) and Gemma 4 E4B (57.8%) remained the best without hints (63.8–131.6% for the others); with hints, Gemma 4 12B (36.7%) came close to Gemma 4 E4B (34.1%). Manual inspection of a few songs suggests that some errors of Gemma 4 E4B come from mishearing and from long repetitions of vocalizations (“oh oh oh”) that may or may not belong in the transcript. We compare Qwen2.5-Omni-7B and Gemma 4 E4B, each with and without hints (QwenOmni2.5-7b-nohint, QwenOmni2.5-7b-ASR-hints, Gemma-e4b-nohint, Gemma-e4b-ASR-hints); QwenOmni2.5-7b-ASR-hints is missing from the results because its run failed on the day of the deadline.

5. DEVELOPMENT RESULTS

Table 1 shows the results on *multiling500*. In WER and CER, Qwen-ASR and the ASR combination are the best, with Whisper slightly behind. Among the audio-LLM systems, Gemma with ASR hints has the lowest WER and CER, but its WER is still higher than those of both its hints, so it is unclear whether the hints help it recognize anything or whether it merely avoids spoiling them too much. Whisper and the ASR combination have the best punctuation, the audio-LLM systems slightly better line breaks, and the ASR combination the best section breaks, closely followed by QwenOmni2.5-7b-nohint. We do not discuss WER’, as

System	WER ↓	CER ↓	WER' ↓	F_P ↑	F_B ↑	F_L ↑	F_S ↑
Whisper	33.3	21.8	46.1	24.2	–	70.0	0.1
Qwen-ASR	28.9	16.9	40.6	18.9	–	53.1	–
Whisper+Qwen-ASR+Qwen3-8b	29.7	18.1	43.2	20.9	0.0	64.7	23.1
QwenOmni2.5-7b-nohint	42.5	28.9	55.0	14.5	0.0	75.6	21.7
Gemma-e4b-nohint	52.5	31.3	61.3	0.7	0.3	73.5	14.8
Gemma-e4b-ASR-hints	33.8	21.1	47.0	2.8	0.2	76.4	15.6

Table 1. Results on *multiling500* (500 songs, 31.6 h), all in %. F_T is the F-score for punctuation (P), parentheses (B), line breaks (L), and section breaks (S); “–” means no tokens of that type in the output.

the DALI references are lowercase.

Overall, ASR outperforms the audio-LLMs. Qwen-ASR alone has the lowest WER and CER, but the LLM combination of two ASR systems comes close while giving the best section breaks and good punctuation, i.e., the best balance over all metrics.

CMI-Bench [3] evaluated mostly older audio-LLMs (Qwen2-Audio, used in our preliminary experiments, is the only overlap) on DSing [18], English solo singing without accompaniment, which is much simpler than our multilingual polyphonic data. All models had WER above 100%, usually far above, and the authors conclude that none is usable. On DALI, with suitable prompting, recent audio-LLMs get much closer to the ASR systems, with WER safely below 100%; a safe comparison would require evaluating them on DSing too. Still, much smaller ASR models outperform open-weight general-purpose audio-LLMs on lyrics transcription.

6. SUBMITTED SYSTEM

We submit four systems: Whisper as **UFAL-PMCG-Whisper**, Qwen-ASR as **UFAL-PMCG-Qwen-ASR**, Whisper+Qwen-ASR+Qwen3-8b as **UFAL-PMCG-Whisper+Qwen-ASR+Qwen3-8b**, and Gemma-e4b-ASR-hints as **UFAL-PMCG-Gemma-e4b-ASR-hints**. None of them uses training data or training. Table 2 gives their sizes and resource use.

7. LIMITATIONS AND CONCLUSION

The pretrained models may have seen the evaluation data during training; their training data are not published, so we cannot check this.

On a multilingual subset of DALI, off-the-shelf ASR models transcribe lyrics better than audio-LLMs that fit into 32 GB of GPU memory. ASR hints narrow the gap but do not close it. Merging two ASR transcripts with a text-only LLM keeps WER close to the best single model and adds section structure; this is our primary system.

8. ACKNOWLEDGMENTS

This work was supported by the project “Human-centred AI for a Sustainable and Adaptive Society” (reg. no.:

CZ.02.01.01/00/23_025/0008691), co-funded by the European Union, and partially by the SVV project number 260 821. The work described herein has been using services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (<https://lindat.cz>), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

9. AI USAGE STATEMENT

We used Claude Code (Anthropic) to write the code of our experiments and to help draft, edit, and shorten this paper. The authors checked all content, results, and references.

10. REFERENCES

- [1] O. Cífka, H. Schreiber, L. Miner, and F.-R. Stöter, “Lyrics transcription for humans: A readability-aware benchmark,” in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, CA, USA, 2024, pp. 737–744.
- [2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. of the 40th Int. Conf. on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202. PMLR, 2023, pp. 28 492–28 518.
- [3] Y. Ma, S. Li, J. Yu, E. Benetos, and A. Maezawa, “CMI-Bench: A comprehensive benchmark for evaluating music instruction following,” in *Proc. of the 26th Int. Society for Music Information Retrieval Conf.*, Daejeon, South Korea, 2025, pp. 416–425.
- [4] X. Shi, X. Wang, Z. Guo, Y. Wang, P. Zhang *et al.*, “Qwen3-ASR technical report,” arXiv:2601.21337, 2026.
- [5] S. Durand, D. Stoller, and S. Ewert, “Contrastive learning-based audio to lyrics alignment for multiple languages,” in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
- [6] D. Stoller, S. Durand, and S. Ewert, “End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model,” in *Proc. of the IEEE*

System	Params (B)	Time (s/h)	GPU (GB)
Whisper	1.55	139	5.3
Qwen-ASR	3.6	349	5.3
Whisper+Qwen-ASR+Qwen3-8b	11.8	631	17.3
Gemma-e4b-ASR-hints	11.6	760	16.1

Table 2. Submitted systems: total parameters of all models in the pipeline (Whisper large-v2 1.55 B, Qwen3-ASR-1.7B 2.04 B, Qwen3-8B 8.19 B, Gemma 4 E4B 8.00 B including per-layer embeddings), inference time per hour of audio, and peak GPU memory, measured on *multiling500* on an NVIDIA A40.

- Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 181–185.
- [7] J. Syed, I. M. Higgs, O. Cífka, and M. Sandler, “Exploiting music source separation for automatic lyrics transcription with Whisper,” in *Proc. of the IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW)*, Nantes, France, 2025.
- [8] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “MUSDB18 – a corpus for music separation,” Zenodo, Dec. 2017.
- [9] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “DALI: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm,” in *Proc. of the 19th Int. Society for Music Information Retrieval Conf.*, Paris, France, 2018, pp. 431–437.
- [10] AudioShake, “alt-eval: Readability-aware automatic lyrics transcription (ALT) evaluation toolkit,” <https://github.com/audioshake/alt-eval>, 2023.
- [11] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, Y. Leng, Y. Lv, J. He, J. Lin, C. Zhou, and J. Zhou, “Qwen2-Audio technical report,” arXiv:2407.10759, 2024.
- [12] J. Xu, Z. Guo, J. He, H. Hu, T. He, S. Bai, K. Chen, J. Wang, Y. Fan, K. Dang, B. Zhang, X. Wang, Y. Chu, and J. Lin, “Qwen2.5-Omni technical report,” arXiv:2503.20215, 2025.
- [13] Gemma Team, “Gemma 4 technical report,” arXiv:2607.02770, 2026.
- [14] SYSTRAN, “faster-whisper: Faster Whisper transcription with CTranslate2,” <https://github.com/SYSTRAN/faster-whisper>, GitHub repository.
- [15] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui *et al.*, “Qwen3 technical report,” arXiv:2505.09388, 2025.
- [16] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. Li, G. Zhang, S. Liu, R. B. Dannenberg, J. Fu, C. Lin, E. Benetos, W. Chen, W. Xue, and Y. Guo, “LyricWhiz: Robust multilingual zero-shot lyrics transcription by whispering to ChatGPT,” in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023, pp. 343–351.
- [17] J. Prakash, B. Kumar, K. Hacioglu, B. Sharma, S. Gopalan, M. Chetlur, S. Venkatesan, and A. Stolcke, “Better pseudo-labeling with multi-ASR fusion and error correction by SpeechLLM,” in *Proc. Interspeech*, Rotterdam, The Netherlands, 2025.
- [18] G. Roa Dabike and J. Barker, “Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system,” in *Proc. Interspeech*, Graz, Austria, 2019, pp. 579–583.