

A TOOL FOR AUTOMATICALLY TRANSCRIBING IN-TERVIEWS WITH SPEAKER RECOGNITION

Lukáš Kleveta¹, Jaromír Landa¹

¹Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, Czech Republic

DOI: https://doi.org/10.11118/ecos2026_012

ABSTRACT

The present paper presents the design and implementation of a software tool for automatically transcribing audio recordings, supplemented by speaker recognition (diarization). The tool integrates two open-source components: The implementation of OpenAI's Whisper AI model for speech-to-text conversion and the pyannote.audio library or, alternatively, NVIDIA NeMo - for speaker recognition. The architecture of the system is implemented as a linear pipeline with modules for audio preprocessing, diarization, transcription, and result merging. It is possible to modify key processing parameters (model, language, thresholds, clustering method) via a configuration file without modifying the code. The functionality was verified on two test sets. The first was a short English recording with a single speaker. In this case, a quantitative comparison of Whisper AI models was made using the Levenshtein metric. The second was a one-minute recording with two speakers. In this case, a comparison was made of twenty configurations in terms of transcription, diarization, and computational complexity. The medium and large-v2 models demonstrated the highest degree of mutual agreement among transcriptions (97.79%), while the lowest was exhibited by the tiny and large-v3 models (82.88%). With regard to overall balance, the configuration employing the large-v2 model and the NeMo diarization module with spectral clustering was identified as the optimal solution. The findings substantiate the practical viability of open-source instruments for automated transcription. However, they concurrently underscore the constraints associated with multilingual recordings, the computational intricacy, and the susceptibility of diarization to variations in audio quality.

Keywords: automatic speech transcription, speaker diarization, Whisper, pyannote.audio, NVIDIA NeMo, audio processing, open-source, Levenshtein distance, pipeline

JEL Code: C88, L86, O33



This article is licensed under CC BY-SA 4.0 Attribution-ShareAlike 4.0 International license.
<https://creativecommons.org/licenses/by-sa/4.0/deed.en>

1 INTRODUCTION

In recent years, the field of automatic speech processing has witnessed substantial advancements, largely attributable to the emergence of machine learning models. The capacity to transform spoken language into a structured text transcript, inclusive of speaker identification, engenders a plethora of pragmatic applications. These encompass the indexing of audiovisual archives, the analysis of conference recordings, the facilitation of journalistic and editorial endeavours, and the processing of qualitative data in research (Miró *et al.*, 2012; Park *et al.*, 2022).

An open-source model that has fundamentally pushed the boundaries of accessibility in this field is OpenAI’s Whisper. The model is an encoder-decoder system with a Transformer architecture that has been trained on 680,000 hours of predominantly multilingual audio, a fact that has enabled it to deliver robust performance across a range of accents and background noise (Radford *et al.*, 2023). Whisper AI is distributed as open-source, allowing for free deployment in both academic and commercial projects without the need to adapt the model to specific input.

Speech transcription naturally involves the task of speaker identification, known as diarization. Diarization provides an answer to the question of the speaker’s identity and temporal segmentation of the recording based on the speakers’ identities (Tranter and Reynolds, 2006). Current approaches encompass classical clustering algorithms based on i-vectors or x-vectors, as well as end-to-end neural models (Park *et al.*, 2022). An open-source tool for this task is the Python library pyannote.audio, which combines neural segmentation with agglomerative clustering of speaker vectors (Bredin *et al.*, 2020).

Notwithstanding advances in individual sub-tasks, the successful deployment of integrated transcription and diarization systems remains non-trivial. The following persistent challenges have been identified: short speech segments, overlapping speech, noise, and sensitivity to sampling rate and audio compression (Miró *et al.*, 2012). The practical implementation of this technology is further complicated by differences in speaker characteristics due to factors such as age, gender, and accent (Church *et al.*, 2017). Furthermore, the synchronisation of the outputs of two independent models – one for transcription and one for diarisation – generates a distinct class of errors related to the temporal alignment of segments.

This paper summarises the results of a bachelor’s thesis whose objective was to design, implement, and test a functional open-source tool that combines these components into a coherent pipeline. The emphasis was placed on three aspects: The system under discussion features the following characteristics:

- Modular architecture, which allows individual processing steps to be swapped without affecting the rest of the system.
- Configurability of key transcription and diarization parameters via an external YAML file.
- Robustness when processing longer recordings using overlapping segmentation.

The impetus for this endeavour is rooted in the mounting demand for automated tools to facilitate the transcription of textual data derived from qualitative research methods. The transcription of an hour-long interview, a task typically undertaken by a qualified transcriber, traditionally necessitates several hours of labour. The implementation of an automated solution has the potential to markedly reduce the time investment required, albeit at the expense of necessitating human review of the generated transcription. The reliability of such automation is contingent upon a combination of the transcription model, the diarization model, and the processing parameter settings.

The proposed paper will address the following research questions. The first question is to what extent the transcriptions generated by different Whisper AI model sizes differ for the same recording. The second question pertains to the identification of the most effective combinations of configuration parameters (Whisper AI model size, temperature parameter value, choice of diarization module, and clustering method) that yield optimal results in terms of transcription quality, diarization accuracy, and computational complexity. The objective of this research is not to attain state-of-the-art results on standard benchmark data, but rather to ascertain the viability of constructing a practically usable tool from freely available components and to delineate its vulnerabilities.

The remainder of the paper is structured as follows. The Methodology and Data chapter provides a comprehensive overview of the pipeline architecture, the technologies employed, audio preprocessing, the principles of diarization and transcription, and the evaluation metrics. The Results chapter presents a quantitative comparison of Whisper AI models using the Levenshtein metric and a comparative test of twenty configurations on a longer recording. The final section summarises the main findings, the limitations of the chosen approach, and directions for further research.

2 METHODOLOGY AND DATA

The proposed system is to be implemented as a linear pipeline consisting of four main modules. The initial module, designated as AudioProcessor, is responsible for the preliminary processing of the input audio. The second module (Diarizer) performs speaker recognition. The third module, designated as Transcriber, facilitates the conversion of speech into written form. The fourth module (ResultsMerger) functions to merge the diarization and transcription outputs, thereby generating the final structured output in the selected format (TXT, JSON, SRT). The communication infrastructure is characterized by the implementation of clearly defined data interfaces, facilitating seamless interaction between modules. A comprehensive logging system is in place to ensure meticulous documentation of each phase, thus facilitating diagnostic processes. The architecture diagram is shown in Figure 1.

Audio preprocessing can be delineated in three distinct stages. The initial procedure entails the implementation of noise reduction through the utilization of the spectral subtraction method, with the ‘prop_decrease’ parameter set to 0.75. This parameter serves to specify the proportion of the identified noise that is to be subtracted from the signal. It has been demonstrated that an increase in value results in enhanced intelligibility; however, it is important to note that overly aggressive reduction can lead to the distortion of speech segments. The subsequent stage involves amplitude normalization along the channel dimension of the tensor, thereby ensuring equilibrium of the volume. The third step in the process is the conversion of the audio file to a standardised WAV format, with a sampling rate of 16 kHz and a single channel (mono). This is a prerequisite for the Whisper AI model. For recordings exceeding a specified size, a segmentation technique with configurable overlap is implemented. This approach serves to mitigate the risk of context loss at boundaries, thereby ensuring the integrity of the recorded data.

The diarization module has been designed to allow for extensibility. The abstract class, designated as BaseDiarizer, establishes an interface that is subsequently implemented by specific modules. Two such implementations are available within the system: the first is labelled ‘PyAnnoteDiarizer’, and employs the ‘pyannote.audio’ library with the pre-trained model ‘pyannote/speaker-diarization-3.1’; the second is labelled ‘NemoDiarizer’, and is built on top of the NVIDIA NeMo library.

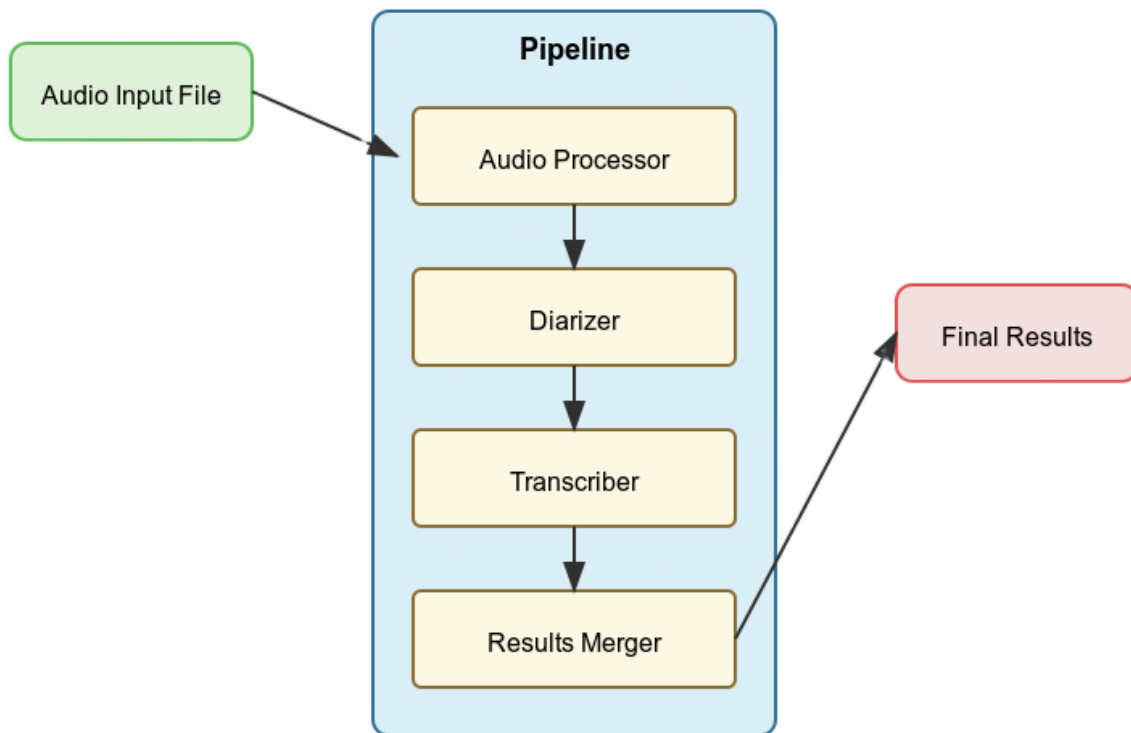


Fig. 1: Diagram of a linear pipeline for audio processing with diarization and transcription (Audio Input → AudioProcessor → Diarizer → Transcriber → ResultsMerger → Final Results) (Kleveta, 2025)

The module is selected through the use of a configuration file and the `DiarizationFactory` class. The present version of the PyAnnote pipeline (Bredin *et al.*, 2020) operates in three phases: local segmentation using five-second sliding windows, speaker vector extraction, and global agglomerative clustering (Bredin, 2023). NeMo employs a similar methodology, utilizing its own models for voice activity detection (`vad_multilingual_marblenet`), embedding extraction (`titanet_large`), and spectral clustering. The output in both cases is a list of time segments with assigned speaker identifiers.

The transcription module utilizes Whisper AI models of varying sizes, ranging from the compact model (39 million parameters) to the large-v3 model (1.55 billion parameters), in addition to an optimized turbo variant. The key parameters that must be considered include the task selection (transcribe or translate), temperature (which affects output variability), `compression_ratio_threshold` (for detecting excessive repetitiveness), and `no_speech_threshold` (which sets the threshold for detecting silence). By default, the temperature is set to 0, thereby ensuring the output is deterministic. In the absence of user input regarding the desired language, the system will automatically detect the language of the audio. In order to utilize GPU acceleration via CUDA technology, the system automatically switches to the hardware variant if available.

The evaluation of transcription quality was conducted through the utilization of two complementary metrics. The standard metric employed is the Word Error Rate (WER), defined as the ratio of the sum of word substitutions, insertions, and deletions to the number of words in the reference transcription (Ali and Renals, 2018). For languages with complex morphology or ambiguous word boundaries, the Character Error Rate (CER), which operates at the character level, is

more suitable (Thennal *et al.*, 2024). In the comparison performed, the Levenshtein distance was additionally utilized. This is defined as the minimum number of insertions, deletions, and character substitutions required to transform one string into another (Haldar and Mukhopadhyay, 2011). Computationally, it is implemented using dynamic programming on a matrix of dimensions $(m+1) \times (n+1)$. This metric enables a direct comparison of transcription pairs without the necessity for a reference “true” transcription. The Diarization Error Rate (DER) is a metric employed for the evaluation of diarization processes. It calculates the time proportions of false detections, omitted speech, and speaker identity mix-ups (Bredin, 2017). Supplementary metrics, including purity and coverage, facilitate the discernment of errors arising from excessive or insufficient segmentation.

The test data consists of two recordings. The initial recording is a concise, approximately 10-second recording in English featuring a sole speaker. It is characterized by clarity and absence of substantial noise, and is employed for the purpose of comparison of Whisper AI models through the utilization of the Levenshtein metric. The second recording is approximately one minute in duration and features two speakers, with the first speaker dominating for approximately 90% of the time. The audio is characterized by clarity and fluidity in the enunciation of American English, with subtle indications of Canadian influences. A total of twenty distinct configuration files were subjected to rigorous testing on this recording, with a particular emphasis on alterations to transcription parameters (temperature, thresholds), the selection of diarization module, embedding model, and clustering method. The limitations of the test set are discussed in the concluding chapter.

3 RESULTS

The initial experiment focused on comparing six multilingual Whisper AI models (tiny, base, small, medium, large-v2, large-v3) using a 10-second English test recording. For each pair of models, the Levenshtein distance between their transcriptions and the corresponding percentage match was calculated.

The highest mutual match was observed between the medium and large-v2 pair (97.79%, Levenshtein distance equal to 3). The transcriptions differed only in the ending of the last sentence. In contrast, the lowest match was observed between the tiny and large-v3 models (82.88%, distance 25). This outcome merits further consideration. It is a commonly held belief that a larger model should always provide a more accurate transcription. However, a qualitative analysis reveals a more complex picture.

In the test recording, the tiny model terminated the transcription at the word “world”, which corresponded to the actual end of the utterance. In contrast, the large-v3 model incorporated the sentence “had been destroyed,” which is not present in the original recording. This phenomenon is referred to as a hallucination, where the model generates coherent but contextually unsupported text. Conversely, smaller models tend to exhibit a distinct category of errors, including the generation of nonsensical word forms (e.g., “und” in the transcription of the tiny model) and punctuation errors.

This analysis suggests the following partial conclusion: for brief, linguistically well-defined recordings, increasing the model size does not guarantee an improvement in the quality of the output. Whilst larger models have been shown to excel in the identification of punctuation, sentence structure and complex vocabulary, there is also an increased risk of hallucinations with increasing capacity. With regard to consistency, the medium and large-v2 pair demonstrated the greatest stability.

The second experiment focused on evaluating the impact of configuration parameters on a more complex test recording featuring two speakers. Twenty configuration variants were prepared, which sequentially varied the size of the Whisper AI model, the value of the temperature parameter, the diarization module used (pyannote vs. NeMo), the embedding model (titanet_large vs. resnet), and the clustering method (spectral, sc, ahc). The evaluation was based on four criteria: content consistency of the transcription, diarization quality, output stability, and computational complexity. Table 1 provides a synopsis of the five optimal configurations.

The most balanced results were achieved by the Config_14 configuration, which was the only one to achieve the maximum score in both the transcription and diarization categories, albeit at the cost of a slightly lower performance score. The most accurate transcription was achieved by the Config_11 configuration, which combined the large-v2 model with a low temperature parameter value and optimized thresholds for `compression_ratio_threshold` and `logprob_threshold`. Although the Config_19 configuration showed comparable diarization quality, it failed to detect the secondary speaker, and its performance score was the lowest among the five configurations compared.

Three observations emerge from the analysis. First, lower values of the temperature parameter (typically 0 to 0.2) lead to more stable and contextually consistent transcriptions, while higher values increase output variability and, in some cases, generate less accurate formulations. Second, the NeMo diarization module achieved higher quality in the tests than the baseline pyannote.audio, particularly in distinguishing between two speakers on a recording with an unbalanced speaking share. Third, the choice of clustering method significantly affects the diarization output: spectral clustering (sc) and agglomerative hierarchical clustering (ahc) outperformed simpler approaches.

Drawing upon the findings of individual case studies, the following proposed configuration was put forward: the Whisper AI large-v2 model, the transcribe task, a temperature set to 0.2, the NeMo diarization module with the titanet_large embedding model, spectral clustering, and VAD parameters set to `threshold=0.45` and `window_length_in_sec=0.15`. This configuration combines the strengths of successful variants from the test set; however, it cannot be claimed that this is the universally best setting for any type of recording. The generalisability of the findings is limited by two factors. Firstly, the size of the test set is small. Secondly, the predominance of the English language in the test data is also a factor (see Discussion and Conclusions section).

Tab. 1 Comparison of the best five configurations out of twenty tested. Values on a scale of 1–5; the “Second speaker” column indicates whether the presence of a second speaker was correctly detected. (Kleveta, 2025)

Configuration	Transcription	Diarization	Performance	Number of speakers	Second speaker	Time (s)
Config_14	5.0	5.0	4.0	2	Yes	12.5
Config_11	4.6	4.9	4.5	2	Yes	11.2
Config_03	4.8	4.7	4.5	2	Yes	10.3
Config_19	4.0	4.7	2.5	2	No	13.7
Config_20	4.4	4.5	3.8	2	Yes	10.9

4 DISCUSSION AND CONCLUSIONS

The findings of the experiments demonstrate that by integrating the open-source components Whisper AI and pyannote.audio, or NVIDIA NeMo, it is feasible to construct a functional tool for automatic speech transcription with speaker recognition, yielding a practical output quality for concise, intelligible recordings of spoken English. The modular pipeline architecture ensures that individual components can be exchanged without affecting other parts of the system, and the external YAML configuration allows for parameter tuning without modifying the source code.

Concurrently, the experiments conducted revealed several significant limitations. Of these, the size of the test set itself is of the greatest importance. The initial experiment is founded upon a solitary 10-second recording, whilst the subsequent experiment is predicated upon a 60-second recording. Consequently, any conclusions that are drawn about the relative quality of the models and configurations must be generalised with the understanding that they may not be universally applicable. In order to perform a more robust comparison, it would be necessary to use a more extensive and linguistically diverse test set, ideally an established benchmark with a standardized reference. In this sense, the present study is more of an exploratory feasibility study than a rigorous evaluation.

A secondary constraint pertains to the system's reliance on two distinct models: one for transcription and the other for diarization. This approach is not without risk of temporal mismatches between the outputs of the two modules, which can lead to inaccurate assignment of text segments to individual speakers, particularly near speaker-switch boundaries. This problem is well-documented in the academic literature and has motivated the development of integrated approaches such as end-to-end neural diarization (Park *et al.*, 2022) or post-processing diarization using language models (Wang *et al.*, 2024). However, the implementation of such approaches would result in a substantial increase in the requirements for training data and computational power.

The third limitation pertains to the behaviour of Whisper AI models when confronted with multilingual content. In the course of the testing process, it was repeatedly observed that the model consistently identified the language of the entire recording as English, even in cases where the recording contained segments in other languages. The model then proceeded to translate or completely mis-transcribe these segments. In certain instances, the phenomenon of “stuck” behaviour manifested, wherein the subsequent segments of transcription became disassociated from the underlying content following the occurrence of a problematic segment. This issue persisted even when the transcribe task was explicitly set. In multilingual contexts, which are prevalent in the European environment, this behaviour poses a significant obstacle to practical deployment. Current developments in the field of massively multilingual ASR are paving the way for models covering hundreds of languages (Pratap *et al.*, 2024). However, a separate task beyond the scope of this paper is the systematic comparison of these models with Whisper AI models on recordings containing Central European languages.

The fourth limitation pertains to computational complexity. It has been demonstrated that larger Whisper AI models (large-v2, large-v3) require GPU acceleration when combined with NeMo diarization in order to process data within a reasonable timeframe. In a purely CPU-based environment, processing time increases to such an extent that practical application to longer recordings is rendered unfeasible. The necessity to divide extended recordings into overlapping segments gives rise to a further source of error, namely the potential loss of contextual information at segment boundaries, which may consequently lead to inconsistencies in the identification of the same speaker across adjacent segments.

From the perspective of practical application, these results have several implications. The utilisation of automated transcription with diarization facilitates the reduction of the time required for the preparation of text transcripts in qualitative research, journalism, podcast production, and audiovisual archive management. However, none of the tested systems provides an output that could be published directly without human review. Errors in punctuation, capitalisation of proper names, and hallucinations in larger models require final editing. The potential for real-world time savings is therefore contingent on the availability of an editor proficient in the target language and the subject matter of the recording.

It is possible to identify a number of potential avenues for the tool's future development. The initial approach involves the implementation of an end-to-end model that integrates diarization and transcription, with the objective of resolving the issue of temporal alignment (Park *et al.*, 2022). The second is the implementation of a visual user interface that will enable interactive correction of the transcript and re-assignment of segments to speakers. The third direction is deployment in a cloud environment with flexible allocation of GPU resources, which would lower the barrier to use for users without their own powerful hardware. The fourth direction is systematic evaluation on a larger and more linguistically diverse dataset, which would allow for more reliable conclusions about the relative performance of the configurations.

In conclusion, it can be stated that the proposed pipeline fulfils the objective of this bachelor's thesis—namely, to verify whether it is possible to build a practical tool for automatic transcription of recordings with speaker recognition using freely available open-source components. The experiments conducted confirm that the choice of the Whisper AI model and the configuration of the diarization module have a significant impact on output quality. The most balanced results on the test data were achieved by a configuration combining the Whisper AI large-v2 model with the NeMo diarization module and spectral clustering. The identified limitations – chiefly the modest proportions of the test set, challenges pertaining to multilingual content, computational intricacy, and errors at segment boundaries – delineate the agenda for future research. In view of the rapid advances being made in the field of speech models, it is reasonable to assume that some of the aforementioned limitations will be addressed in the near future by subsequent generations of open-source models.

Acknowledgements

This paper was supported by the project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists, this is co-financed from Operational Programme Research, Development and Education.

Several AI tools were used in the preparation of this article. They assisted with summarizing source material, translating passages between languages, and identifying typographical and stylistic issues in the text. All output produced by these tools was reviewed and verified. Summaries were checked against the original sources, translations were assessed for accuracy, and suggested corrections were evaluated before being accepted. The tools served as an aid to the writing process; the final content reflects the authors' judgment and remains the authors' responsibility.

REFERENCES

- ALI, A., RENALS, S. 2018. Word error rate estimation for speech recognition: e-WER. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne: Association for Computational Linguistics, pp. 20–24.
- BREDIN, H. 2017. pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. In: *Proceedings of Interspeech 2017*. Stockholm: ISCA. <https://doi.org/10.21437/interspeech.2017-411>
- BREDIN, H. 2023. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In: *Proceedings of Interspeech 2023*. Dublin: ISCA.
- BREDIN, H. *et al.* 2020. pyannote.audio: neural building blocks for speaker diarization. In: *ICASSP 2020 – IEEE International Conference on Acoustics, Speech and Signal Processing*. Barcelona: IEEE. <https://doi.org/10.1109/ICASSP40776.2020.9052974>
- CHURCH, K. *et al.* 2017. Speaker diarization: a perspective on challenges and opportunities from theory to practice. In: *ICASSP 2017 – IEEE International Conference on Acoustics, Speech and Signal Processing*. New Orleans: IEEE. <https://doi.org/10.1109/ICASSP.2017.7953098>
- HALDAR, R., MUKHOPADHYAY, D. 2011. Levenshtein distance technique in dictionary lookup methods: an improved approach. arXiv preprint. *arXiv:1101.1232*. <https://doi.org/10.48550/arXiv.1101.1232>
- KLEVETA, L. 2025. *Nástroj pro automatické přepisy rozhovorů s rozpoznáním řečníků*. Bachelor's thesis. Brno: Mendel University in Brno, Faculty of Business and Economics.
- MIRÓ, X. A. *et al.* 2012. Speaker diarization: a review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*. 20(2), 356–370. <https://doi.org/10.1109/TASL.2011.2125954>
- PARK, T.J., KANDA, N., DIMITRIADIS, D. 2022. A review of speaker diarization: recent advances with deep learning. *Computer Speech & Language*. 72, 101317. <https://doi.org/10.1016/j.csl.2021.101317>
- PRATAP, V., TJANDRA, A., SHI, B. *et al.* 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*. 25.
- RADFORD, A., KIM, J. W., XU, T., BROCKMAN, G., MCLEAVEY, C., SUTSKEVER, I. 2023. Robust speech recognition via large-scale weak supervision. In: *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*. PMLR, vol. 202.
- THENNAL, D. K., JAMES, J., GOPINATH, D., ASHRAF, M. 2024. Advocating character error rate for multilingual ASR evaluation. arXiv preprint. *arXiv:2410.07400*. <https://doi.org/10.48550/arXiv.2410.07400>
- TRANTER, S. E., REYNOLDS, D. A. 2006. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*. 14(5), 1557–1565. <https://doi.org/10.1109/TASL.2006.878256>
- WANG, Q., HUANG, Y., ZHAO, G., CLARK, E., XIA, W., LIAO, H. 2024. DiarizationLM: speaker diarization post-processing with large language models. arXiv preprint *arXiv:2401.03506*. <https://doi.org/10.48550/arXiv.2401.03506>

Contact information

Lukáš Kleveta: e-mail: xkleveta@node.mendelu.cz

Jaromír Landa: e-mail: jaromir.landa@mendelu.cz