TRANSCRIBING AND ALIGNING CONVERSATIONAL SPEECH: A HYBRID PIPELINE APPLIED TO FRENCH CONVERSATIONS

Hiroyoshi Yamasaki[1], *Jérôme Louradour*[2], *Julie Hunter*[2], *Laurent Prévot*[1,3]

[1] Aix Marseille Université & CNRS, LPL, Aix-en-Provence, France
[2] LINAGORA Labs, Toulouse, France
[3] CEFC, CNRS & MEAE, Taipei, Taiwan

ABSTRACT

With the advent of transformer based models, the use of fully automated ASR-pipelines in a real-world context has come close to a reality. However, for faithful transcription of conversational speech, there remain challenges both in terms of the content predicted by these models (hallucinations, unintended normalizations of disfluencies and transcriptions of background noises) and in terms of alignment accuracy. In this paper we present a hybrid ASR-pipeline which augments transformer models with other algorithms in order to transcribe conversational data. Through experiments on two French datasets, we show that: 1) VAD preprocessing can significantly improve transcription quality as well as word level temporal alignment, 2) prompting can reduce unintended normalizations of disfluencies, 3) heuristic-based detection of untranscribed sounds can further improve alignment quality. We conclude that our hybrid pipeline is an efficient way to improve and augment existing ASR-models.

***Index Terms*—ASR, Transformer, French, Conversation**

I. INTRODUCTION

While automatic speech recognition (ASR) has steadily improved over the past few decades for a variety of tasks and domains, conversational speech has remained a challenge. Until recently, word error rate (WER) had been so high that for tasks requiring accurate transcripts, manual transcription was often more efficient than correcting ASR output. The situation changed dramatically with the introduction of Whisper [1], a transformer-based, end-to-end model trained on massive amounts of speech data. Other models followed, including models supporting additional languages [2], [3], and diverse extensions of the Whisper model [4], [5].

This significant improvement in ASR models promises to greatly facilitate the production of conversational datasets, eliminating the central bottleneck for downstream linguistic study and modeling of conversational speech. Still, applying Whisper to conversational corpora presents certain challenges: Whisper (i) easily registers and transcribes background voices in cases where speakers have individual microphones but are in the same room (the standard scenario for conversational corpora such as ICSI [6], AMI [7] and CID [8]), (ii) hallucinates content (inserts words not in the audio signal), (iii) frequently omits disfluencies and discourse markers, which are common in spontaneous speech, and (iv) either lacks word-level alignment (original model) or struggles with alignment when there are untranscribed disfluencies or other noises (more recent implementations).

In this paper, we present a pipeline designed to address the above challenges and facilitate transcription of conversational speech with a model like Whisper. We illustrate the impact of our pipeline on two datasets in French, a language which is well represented in the multilingual models but without large manually transcribed or corrected conversational corpora available for developing and training conversational models like AMI [7] or Switchboard [9] for English. We hope this will open the door to the development of further conversational corpora and benchmarks in a wide variety of languages.

II. DATASET DESCRIPTION

We evaluate our pipeline on two French datasets: the Corpus of Interactional Data (CID) [8] and the newly created SUMM-RE corpus. CID features eight 1-hour long conversations between two friends talking about a given topic but without other constraints. SUMM-RE includes 300 roughly 20-minute conversations between 3-4 speakers, most of whom did not know each other before recording.[1]

Participants in SUMM-RE were instructed to enter into a loosely guided role-play to simulate meeting-style interactions. Both corpora contain highly spontaneous interactional speech with a high rate of disfluencies [10] and a complex mix of monologic (e.g., telling a story) and dialogic (e.g., negotiating the next topic to address) contributions.

CID was manually transcribed in its entirety and double-checked. In addition, each transcript was manually segmented based on silences, yielding gold timestamps for the beginning and end of each speech segment. While most SUMM-RE transcripts have been produced automatically

WE GRATEFULLY ACKNOWLEDGE SUPPORT FROM THE ANR-FUNDED PROJECT, SUMM-RE (ANR-20-CE23-0017)

[1]While most recordings were made in a studio, Covid restrictions forced us to collect some data (<20%) via Zoom.

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

Name	Duration	# speakers / conv.	total # unique speaker
CID	8h08	2	16
SUMM-RE-sm	3h33	3-4	28

Table I. Basic statistics for our two datasets

using the pipeline described in this paper, a small subset of 10 meetings, SUMM-RE-sm, has been manually corrected at both the transcription and segment-timestamp level. Transcription and correction for both corpora were carried out with the Praat phonetic annotation tool [11]. A summary of the data sets is provided in Table I. We will release SUMM-RE-sm with the paper and the rest of the SUMM-RE dataset at the end of the associated research project.

III. CHALLENGES

Background voices While participants in both CID and SUMM-RE wore individual headsets, their microphones often picked up the voices of other speakers when the participants were all in the same room and the ASR system thus transcribed the words of secondary speakers alongside those of the main speaker. The first challenge that we faced was thus to isolate the utterances of the main speaker in each individual speaker recording.

Hallucinations Like other transformer-based models, Whisper has a tendency to hallucinate. We found, for instance, numerous cases in which Whisper predicts "sous-titres réalisés par [x].org" ("subtitles created by [x].org"). Other types of hallucinations involve getting stuck in loops of strings of words or just more standard cases of transcribing words that were not uttered in the conversation. Our second challenge was thus to find ways to keep Whisper from hallucinating.

Transcript "cleaning" Spontaneous conversation contains disfluencies and conversation-specific uses of discourse markers, as illustrated by (1-a):

(1) a. donc euh on a euh enfin j'ai contacté euh notre fournisseur
so um we have um well I have contacted um our supplier

b. j'ai contacté notre fournisseur
I have contacted our supplier

However, as shown in (1-b), Whisper has a tendency to delete discourse markers (in cyan) and disfluencies, including filled pauses (in blue) and fragments (in orange). This not only has a negative impact on the evaluation of ASR output (by increasing deletions and complicating word-level alignment) but also removes information valuable for downstream linguistic tasks requiring richer representations of the dialogue dynamics. Our third challenge was therefore to find ways to help Whisper more faithfully transcribe conversational speech.

Signal alignment Precise word-level[2] timestamps are useful for a variety of downstream tasks that require multimodal, text/audio representations, including discourse segmentation and dialogue act tagging. The original Whisper model, however, only provides timestamps for speech segments. And more recent implementations that add word-level timestamps struggle when there are untranscribed noises in the signal. Omitted disfluencies and discourse markers, for example, complicate the the idenification of timestamps for nearby words. Our fourth challenge, then, was to develop a means of accurately predicting word-level timestamps.

IV. PIPELINE

At a high level, the first part of our pipeline is designed to isolate the contributions of the main speaker in order to reduce the impact of both background voices and silences, which are correlated with hallucinations, on ASR quality. The next part focuses on ASR, exploiting three important additions to the original Whisper model: a prompt that encourages Whisper to transcribe disfluencies and discourse markers, word-level timestamps, and a method for mitigating the impact of untranscribed words and sounds on word-level alignment. Motivation for our pipeline choices comes from tests on CID and SUMM-RE-sm, but also on a small subset of SUMM-RE-sm consisting of roughly 60-90 second extracts of 10 of the files from SUMM-RE-sm. This smaller subset was used to reduce the search space for what we wanted to explore in more depth on the larger datasets.

IV-A. Isolating the main speaker

To isolate contributions from the main speaker, we tested three approaches: (i) inter-pausal-unit (IPU) detection using the SPPAS annotation toolkit [12], (ii) repurposing the Pyannote speaker diarization model [13], [14] for main speaker detection, and finally, (iii) combining SPPAS and Pyannote.

The SPPAS IPU-detection algorithm [15] allows for an optimized threshold value for each file. First, it calculates the root-mean-square (RMS) of the intensity inside a sliding time window of duration 20 ms and then calculates the threshold value $\Theta = \min + \mu - 1.5\sigma$ where min is the minimum, μ is the mean, and σ is the standard deviation for RMS values. Once Θ is calculated, IPU intervals are determined to be those that exceed both Θ and a specified minimum duration (500ms in our case). Silences longer than a given length (100ms in our case) and IPUs that fall under the minimum are treated as silences.

The second approach uses Pyannote to predict speaker turns and labels. We identify the loudest speaker as the main speaker, except in two types of cases that complicate this assumption. First, background speakers sometimes make short but very loud utterances. We tried to eliminate these by applying a minimum duration of 0.3 seconds. Second,

[2]For "word"-level timestamps, we ignore here the difference between words and contractions of words such as "j'ai" (*I've*).

all speakers with a relative intensity above a given threshold (0.8 dBFS) were taken as candidates for main speaker, but sometimes, a single speaker could be assigned two different labels. To remove these cases, we added a secondary filter with intensity threshold of 0.5 dBFS and duration threshold of 0.2 seconds. For the current data set we additionally performed manual verification/correction of the main speaker assignment (2 files out of 39 modified).

Our third approach exploits the complementary benefits of SPPAS and Pyannote. For each case in which SPPAS predicts an interval that overlaps an interval predicted by Pyannote, we take the IPU to be the maximum interval on which the models agree. This approach shows significantly fewer false positives than either system alone, where errors in SPPAS are due to background voices with an intensity too close to that of the main speaker and false positives in Pyannote are linked to similarity in voice quality (e.g., similar age and gender, etc.).[3]

IV-B. Passing segments to Whisper

As preliminary investigations showed that hallucinations are often triggered by silences or non-speech sounds, we decided to feed Whisper segments that had been separately identified as speech (either IPUs or segments selected by a VAD). While such segments can be processed independently by Whisper, we found that it was more efficient to first glue together the relevant segments of audio, separated by intervals of silence, and then pass the whole audio to Whisper (later recovering the original timestamps). We also applied post-processing to remove non-Latin characters, emojis, URLs and any remaining sentences about subtitles.

IV-C. Prompting Whisper

Like other transformer-based models, Whisper can condition its prediction on a given prompt. As it has the particularity of chunking audio files into 30-second segments for transcription, we developed a customized prompt and injected it for the first chunk and then continually reinjected it for the other chunks to maintain its effect throughout the audio. We tried dozens of different prompts that illustrated a variety of disfluencies in order to encourage Whisper to output the disfluencies that were present in the audio signal. The following prompt, involving discourse markers, repetitions, and filled pauses, was the most successful: *Bon. Ben je crois euh je vois ce que euh tu veux dire. Hum tu tu tu euh ben tu me diras*.[4]

[3]Note that as Pyannote detects mainly speech, non-speech sounds such as laughs, coughs etc. are not recognized by this approach, although they are generally included in conversation/dialogue-processing pipelines.

[4]The prompt can be translated as follows:

Bon. Ben je crois euh je vois ce que euh tu veux dire.
Well. Well I think um I see what um you mean.
(literally: *what you want to say*)

Hum tu tu tu euh ben tu me diras.
Um you you you uh well you'll tell me
(literally: *you me will tell*)

IV-D. Signal alignment

Several strategies are possible to recover plausible word-level timestamps. We considered three:

1) Using the external Wav2Vec2 [16] model that predicts character probabilities for each speech frame, and performing an alignment based on Dynamic Time Warping (DTW) [17] on these probabilities for the characters predicted by Whisper. This is the approach adopted by WhisperX [5].
2) Performing DTW on combined cross-attention weights (of the decoder attending to the encoded speech signal) for each predicted token. This is the approach adopted by Whisper-Timestamped [18], Faster Whisper [4] and OpenAI-Whisper [1].
3) Analyzing the probability distribution over timestamps at the end of each word, as in [19], [20].

Preliminary experiments showed that (3) produces very inaccurate timestamps. We therefore restricted our attention in subsequent experiments to the systems using (1) or (2).

Even when using the prompt described above in IV-C, untranscribed disfluencies remain. These together with other non-transcribed acoustic noises degrade the prediction of the timestamps for nearby words that are successfully transcribed. Based on the observation that combined cross-attention weights often form a concave curve around the prediction of each transcribed token, Whisper-timestamped [18] proposes an option to make timestamps more accurate around untranscribed disfluencies and other noises. The heuristics consist in applying a peak detection on the portion of cross-attention weights that correspond to the first token of each word. If several peaks are detected in the vicinity of a single token, the region corresponding to the first peaks is marked as silence, and the start of the last peak is chosen as the start of the word. We adopt this approach to further refine predictions of start timestamps.

V. EVALUATION METRICS

In our experiments, described in Section VI, we evaluated transcription accuracy in terms of Word Error Rate (WER), and its decomposition into Deletion (Del), Insertion (Ins) and Substitution (Sub) rates. We also used the two following metrics to take into account timestamp predictions.

V-A. F1-score

To measure both word errors and timestamp accuracy with a single metric, we use the harmonic mean of precision and recall metrics introduced in [5]. For each word in the ground truth transcript, we extend the interval determined by its start and end time by adding a collar of 20 ms[5] before and after it. If the model correctly predicts the word and predicts that the word overlaps this extended gold interval, then it is

[5]The original paper used 200ms, but a higher degree of precision is needed for downstream applications exploiting both signal and transcripts.

considered a true positive. If the word falls entirely out of the expanded window or if the model fails to predict the word, it is classified as a false negative. Any word predicted by the ASR and for which there is no corresponding word in the gold transcript is counted as a false positive.[6]

V-B. Average timestamp difference T-δ

To evaluate the quality of word timestamps, we first identified the set of words that were correctly predicted. Using DTW [17], we aligned the tokens based on the distance between timestamped words, which we calculated as the sum of the Levenshtein edit distance and the absolute difference between start/end timestamps (in seconds). We found that using DTW with such a hybrid distance calculation is more robust than a simple Levenshtein alignment, for instance in cases where some words are repeated but the repetition only appears in the ground truth or in the prediction.

We then calculated the absolute difference between the predicted start/end time and the real start/end time for each of these correctly predicted words. The average timestamp difference, T-δ, is simply the average of these values.

VI. RESULTS

Table II shows, in the form of an ablation study, the results obtained from testing our pipeline on CID and SUMM-RE-sm. The systems we compare all use Whisper large-v2 as the core model. Our reference system uses IPUs obtained by combining SPPAS and Pyannote predictions (cf. Section IV-A) and then glued together as described in IV-B. It also uses the custom prompt introduced in IV-C, disfluency detection heuristics (cf. IV-D) and then Whisper-Timestamped as the ASR implementation.

Results of Table II:A show a clear reduction in insertions when Whisper receives information on IPUs (Reference) relative to the case where Whisper receives no information on speech segments (No VAD). We also see a notable improvement with the state of the art VAD Silero [21], further underscoring the value of speech segments.

Insertions can be due to a) transcription of background speaker voices or b) hallucinations from Whisper. To distinguish these possibilities, we used gold IPUs to identify intervals of background speech and intervals of silences, as the latter are associated with hallucinations. We then used these intervals to filter (i.e., remove words from) transcripts that had been produced in three different settings: (1) without any VAD, (2) with the VAD Silero, and (3) with our pipeline. In particular, we compare results when (i) background voices are filtered out but silences remain, and (ii) both background voices and silences are filtered, leaving only the main speaker contributions. Figure 1 shows that (i) leads to a significant decrease in insertions when no particular attempt

[6]This metric does not measure a model's capacity for recognizing zones of silence, as there is no penalty for predicting that a word not only overlaps the desired (gold) interval but also covers the silences around that word.

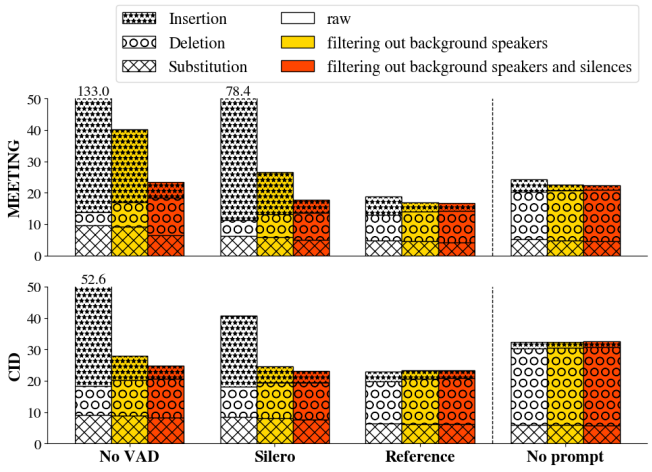

Fig. 1. WER details for different pipeline settings and with different types of filtering for the predicted words.

is made to isolate main speaker contributions (settings (1) and (2)), suggesting that most insertions can be traced to background speakers. Still, the reduction of insertions in these settings when we additionally filter silences suggests that a portion of the errors come from hallucinations.

Table II:B shows that using our custom prompt (cf. IV-C) significantly reduces the number of deletions, as more disfluencies and discourse markers are transcribed. It also slightly increases the insertion rate, but Figure 1 (right column) shows that this is not due to an increase in hallucinations. Whisper also offers the possibility of conditioning its prediction for a given chunk on that of the previous chunk. Table II:B shows that considering the previous chunk (previous) slightly lowers performance on all metrics except insertions, where we see only a subtle improvement.

Heuristics on cross-attention weights to refine the start times of words uttered after untranscribed sounds greatly improve the accuracy of T-δ, while they have no influence on other metrics. Using no prompt and no conditioning on the previous chunk slightly improves T-δ. This is because words recovered by use of our custom prompt are harder to align, which calls into question the reliability of T-δ when comparing models with different WER.

In terms of ASR implementation (Table II:C), WhisperX outperforms others in WER while Whisper-Timestamped has the best T-δ and overall F1-score. Faster Whisper has better T-δ than WhisperX but a poor WER. Differences in WER are most likely due to details relating to decoding heuristics.

We also tested the impact of applying IPUs as a post-processing step to filter out words from ASR transcripts, instead of providing them as input to Whisper. Taking Silero (represented with (✻) in Table II:D) as the only VAD during pre-processing, we tested IPUs predicted by (i) SPPAS, (ii) Pyannote and (iii) our combined approach. We also checked

Pipeline	SUMM-RE						CID					
	F1	**T-δ (ms)**	**WER (%)**	Del (%)	Ins (%)	Sub (%)	**F1**	**T-δ (ms)**	**WER (%)**	Del (%)	Ins (%)	Sub (%)
Our Reference	**0.81**	108	**18.8**	8.1	5.8	4.8	**0.81**	**78**	**22.9**	13.5	3.0	6.4
(A) Speech detection												
Silero (✻)	0.63	117	78.4	5.1	*67.1*	6.3	0.74	82	40.8	9.7	*22.7*	8.4
No VAD	0.49	196	133.0	4.3	*119.1*	9.6	0.68	101	52.6	9.3	*34.3*	9.1
(B) ASR details												
No prompt & no previous	**0.80**	**94**	23.6	*14.1*	4.7	4.8	0.77	**71**	29.1	*20.9*	2.2	5.9
No prompt + previous	0.77	114	24.2	*15.0*	4.0	5.2	0.70	114	32.3	*24.3*	2.1	5.9
No heuristics	**0.82**	204	**18.8**	8.1	5.8	4.8	**0.81**	105	**22.9**	13.5	3.0	6.4
(C) ASR implementation												
WhisperX	**0.80**	157	**18.3**	7.9	6.1	4.3	**0.80**	90	**21.4**	13.3	3.5	5.7
Faster Whisper	**0.80**	120	24.0	*15.1*	4.3	4.7	0.77	82	30.0	*22.2*	1.7	6.1
(D) Post-processing												
Filter: (✻) SPPAS IPU	0.72	109	44.8	6.9	31.8	6.0	0.79	**76**	27.6	11.4	8.1	8.1
Filter: (✻) Pyannote IPU	0.79	108	26.4	8.2	12.6	5.6	**0.80**	**78**	24.8	13.5	4.2	7.0
Filter: (✻) combined IPU	**0.81**	105	21.5	8.8	7.3	5.3	**0.80**	**76**	**23.8**	14.0	3.0	6.8
Filter: combined IPU	**0.81**	107	**19.2**	9.0	5.4	4.9	**0.81**	**77**	**23.5**	14.5	2.8	6.2
Combined IPU + Julius	0.72	113	**18.8**	8.3	5.7	4.8	0.72	82	**22.8**	13.9	2.6	6.4

Table II. Evaluation of different ASR pipelines with word-level timestamps for SUMM-RE-sm and CID.

(iv) filtering with combined IPUs after preprocessing with combined IPUs. To see if we could further improve alignment, we evaluated the impact of (v): supplementing case (iv) with a dedicated alignment tool, Julius [22]. We chose the SPPAS [12] wrapper for Julius and an alignment model that had been trained on conversational French.

Table II:D shows that using combined IPU after Silero yields a lower WER than using SPPAS or Pyannote alone. Using combined IPU post-processing does not lead to significant differences, suggesting that IPUs can be applied before or after the ASR model. Forced alignment with Julius also failed to improve performance.

VII. DISCUSSION

In our experiments, accurately detecting the main speaker had the greatest effect on both transcription accuracy and word-level alignment for all models. This was particularly evident on SUMM-RE, most likely because background speakers were more active than in dyadic CID. Post-processing experiments suggest that this improvement holds regardless of whether IPUs are given as input to the ASR system or applied as filtering after the fact. The use of a custom prompt significantly reduces deletions, and heuristics for dealing with untranscribed noises clearly improve alignment. Noteworthy is the very high WER for out-of-the-box models suggesting that mainstream systems are not suitable for meetings and other natural, conversational data without modification. Our results show that various strategies and heuristics can be applied to enhance state of the art models.

VIII. CONCLUSION AND FUTURE WORK

The pipeline described in this paper is designed to address certain challenges that arise when applying high-performance, transformer-based ASR models to conversational speech. Our pipeline first exploits a combination of an intensity-based VAD and a slightly modified speaker diarization model in order to isolate the conversational contributions of the main speaker in a situation in which speakers have individual microphones. We show that this step allows for a significant reduction in insertions, which can be traced to background voices and hallucinations, and overall improvement in WER, F1-scores and word alignment. Our pipeline then employs a prompt that leads to improvements on ASR metrics by helping the model to predict disfluencies and discourse markers in the audio signal. Finally, we apply additional heuristics to reduce the impact of untranscribed words and other noises on word-level alignment.

Our approach is fairly generic and can be applied to any language. Moreover, the modular nature of our pipeline allows for a flexible combination of various components (ASR, IPU-detection, alignment) as new and improved models become available. Thus, the pipeline can be constantly improved even as the individual models become obsolete.

Future work could consider laughs, coughs and other paralinguistic sounds that are not currently addressed by our pipeline. These signals are not only of interest to researchers of human communication but may also confound ASR models and lead to erroneous transcriptions. Addressing this issue could thus further improve the performance.

IX. ACKNOWLEDGEMENTS

We would like to thank Roxane Bertrand for manually correcting transcripts and Océane Granier for correcting transcripts and collecting the original data.

X. REFERENCES

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[2] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi *et al.*, "Scaling speech technology to 1,000+ languages," *arXiv preprint arXiv:2305.13516*, 2023.

[3] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang, Z. Meng, K. Hu, A. Rosenberg, R. Prabhavalkar, D. S. Park, P. Haghani, J. Riesa, G. Perng, H. Soltau, T. Strohman, B. Ramabhadran, T. Sainath, P. Moreno, C.-C. Chiu, J. Schalkwyk, F. Beaufays, and Y. Wu, "Google usm: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[4] G. Klein, "Faster whisper transcription with ctranslate2," *GitHub repository*, 2023. [Online]. Available: https://github.com/guillaumekln/faster-whisper

[5] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," *arXiv preprint arXiv:2303.00747*, 2023.

[6] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The ICSI meeting corpus," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP'03)*, vol. 1. IEEE, 2003, pp. I–I.

[7] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[8] P. Blache, R. Bertrand, G. Ferré, B. Pallaud, L. Prévot, and S. Rauzy, "The corpus of interactional data: A large multimodal annotated resource," *Handbook of linguistic annotation*, pp. 1323–1356, 2017.

[9] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, speech, and signal processing, ieee international conference on*, vol. 1. IEEE Computer Society, 1992, pp. 517–520.

[10] E. E. Shriberg, "Preliminaries to a theory of speech disfluencies," Ph.D. dissertation, Citeseer, 1994.

[11] P. Boersma and V. Van Heuven, "Speak and unspeak with praat," *Glot International*, vol. 5, no. 9/10, pp. 341–347, 2001.

[12] B. Bigi, "Sppas-multi-lingual approaches to the automatic annotation of speech," *The Phonetician. Journal of the International Society of Phonetic Sciences*, vol. 111, no. ISSN: 0741-6164, pp. 54–69, 2015.

[13] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[14] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Interspeech*, 2021.

[15] B. Bigi and B. Priego-Valverde, "Search for inter-pausal units: application to cheese! corpus," in *9th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, 2019, pp. 289–293.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[17] T. Giorgino, "Computing and visualizing dynamic time warping alignments in R: The dtw package," *Journal of Statistical Software*, vol. 31, no. 7, 2009.

[18] J. Louradour, "whisper-timestamped," *GitHub repository*, 2023. [Online]. Available: https://github.com/linto-ai/whisper-timestamped

[19] jianfch, "Stabilizing timestamps for whisper," *GitHub repository*, 2023. [Online]. Available: https://github.com/jianfch/stable-ts

[20] G. Gerganov, "Whisper.cpp," *GitHub repository*, 2023. [Online]. Available: https://github.com/ggerganov/whisper.cpp

[21] T. Silero, "Silero vad: pre-trained enterprise-grade voice activity detector," *GitHub repository*, 2021. [Online]. Available: https://github.com/snakers4/silero-vad

[22] A. Lee, T. Kawahara, K. Shikano *et al.*, "Julius-an open source real-time large vocabulary recognition engine." in *INTERSPEECH*, 2001, pp. 1691–1694.