

THE SPEECH-LLM TAKES IT ALL: A TRULY FULLY END-TO-END SPOKEN DIALOGUE STATE TRACKING APPROACH

Nizar El Ghazal*, Antoine Caubrière, Valentin Vielzeuf

Orange Innovation
firstname.lastname@orange.com

ABSTRACT

This paper presents a comparative study of context management strategies for end-to-end Spoken Dialog State Tracking using Speech-LLMs. We systematically evaluate traditional multimodal context (combining text history and spoken current turn), full spoken history, and compressed spoken history approaches. Our experiments on the SpokenWOZ corpus demonstrate that providing the full spoken conversation as input yields the highest performance among models of similar size, significantly surpassing prior methods. Furthermore, we show that attention-pooling-based compression of the spoken history offers a strong trade-off, maintaining competitive accuracy with reduced context size. Detailed analysis confirms that improvements stem from more effective context utilization.

Index Terms— Speech-LLM, SpokenDST, Multimodal, Context Propagation

1. INTRODUCTION

Dialog State Tracking (DST) is a vital component in task-oriented dialog (TOD) systems [1, 2], enabling them to understand and maintain the context of a conversation over multiple turns. By accurately tracking user intents and relevant information, DST allows systems to reason over dialog states and effectively fulfill user requests. However, in the context of spoken dialog, Spoken DST remains a relatively immature research area, with current system performance significantly lagging behind those achieved in written dialog scenarios [3]. One of the most common recent approaches is the cascade system. It typically involves an Automatic Speech Recognition (ASR) module followed by an eventual ASR correction module and then a written DST component [4], often based on models such as T5 [5]. This pipeline approach leverages the strengths of existing text-based DST models and was notably popular in the DSTC11 challenge [6], where it was used by the winning system, OLISIA [7].

Despite its success, the cascade approach faces inherent limitations, as it is highly susceptible to error propagation from the ASR stage, which can degrade the overall accuracy of the system [8]. This issue is even more pronounced in real-world scenarios, where ASR systems often struggle with proper nouns and domain-specific terminology, elements that are very frequent in DST slot values [9].

End-to-end (E2E) systems have emerged as a promising alternative, as they may potentially mitigate the error propagation inherent in cascade systems. In particular, [10] demonstrated the effectiveness of E2E approaches, particularly in fully spoken contexts without access to ground-truth transcriptions, such as the SpokenWOZ [3] dataset. In these settings, E2E models have been shown to outperform traditional cascade systems. Concurrently, speech-aware

large language models (LLMs), which are also considered end-to-end (E2E) systems, have gained increasing popularity in a variety of spoken language tasks, including automatic speech recognition (ASR) and response generation [11, 12]. Recent work [13] applied speech-aware LLMs to the spoken DST task, achieving state-of-the-art performance in the SpokenWOZ dataset.

A notable advantage of E2E systems is their flexibility in context management, as they can seamlessly integrate written and spoken information. For instance, [10] and [13] both utilize the spoken representation of the user’s last turn, but differ in how they handle the rest of the context: the former combines the spoken user turn with the written previous state, while the latter combines it with the written representations of all previous turns. This raises an important question. What would happen if we relied solely on spoken context, either by feeding the system the speech representations for the entire conversation or by condensing them using an intermediate module?

In this paper, we explore these possibilities for context management when using a Speech-LLM model. Our contributions are three-fold: **(a)** we validate the use of Speech-LLMs as an accurate approach for spoken DST **(b)** we propose two context management approaches reaching the SOTA and **(c)** our best performing approach demonstrates a simple yet effective method: feeding the entire spoken conversation to the model without additional compression or modality mixing.

2. METHODOLOGY

In task-oriented dialogue (TOD) systems, the role of Spoken Dialog State Tracking (DST) is to condense the user’s intent and relevant information into a structured, machine-readable format. More formally, given as input a sequence of spoken dialogue turns $U_1, A_2, \dots, A_{t-1}, U_{t-1}$, our goal is to predict a set of k relevant domains ($domain_1, domain_2, \dots, domain_k$) and n slot-value pairs ($slot_1 = value_1, slot_2 = value_2, \dots, slot_n = value_n$), which are then represented as a JSON structure.

The Figure 1 illustrates our proposed systems, composed of three main components: a speech encoder, a connector, and a Large Language Model (LLM). In order to reduce the context length, we optionally add a “compression module” between the connector and LLM. The speech encoder processes the entire dialog history and computes dense representations for each turn. These representations are then down-sampled, using x6 stride, and passed to the connector module, which maps the speech features into the LLM’s input space. They may be passed through the compression module for the approaches that need it. Finally, the LLM generates the dialogue state in an auto-regressive manner.

*The author performed the work while at an internship

2.1. Context Management

As represented in Figure 1, we explore several strategies for handling the dialog context.

Multimodal Context Following [13], we provide as input the spoken user utterance U_n^{spoken} and the written dialogue history together. The model then predicts the transcription of the user’s utterance U_n^{text} , the active domains D_n and the dialogue state S_n . The LLM is trained on the prompt:

$h_n \{ \text{"history": } Context_n, \text{"user_last_turn": } U_n^{\text{text}}, \text{"domains": } D_n, \text{"predicted_state": } S_n \}$

where we have:

$$Context_n = \text{USER: } U_1; \text{AGENT: } A_2; \dots; \text{AGENT: } A_{n-1}$$

$$h_n = \text{Connector}(\text{Encoder}(U_n))$$

In practice, the speech representation h_n is concatenated with embeddings that represent the prompt’s text, yielding a multimodal input sequence. During inference, the model autoregressively completes the prompt starting from the field "user_last_turn". The generated ASR hypothesis U_n^{text} is then fed back to construct the textual context $Context_{n+1}$ for subsequent turns.

Full Spoken Context With this context-management strategy, $Context_n$, corresponding to the full spoken conversation, is provided to the model. The model predicts the active domain D_n and the dialogue state S_n . The prompt employed for this strategy is:

$Speech_Emb \{ \text{"domains": } D_n, \text{"predicted_state": } S_n \}$

where :

$$Context_n = (U_1^{\text{spoken}}, A_2^{\text{spoken}}, \dots, U_n^{\text{spoken}})$$

$$h_{2i+1} = \text{Connector}(\text{Encoder}(U_{2i+1}))$$

$$h_{2i} = \text{Connector}(\text{Encoder}(A_{2i}))$$

$$Speech_Emb = (h_1 || h_2 || \dots || h_n)$$

As in the multimodal context setting, the sequence of speech embeddings $Speech_Emb$ is pre-pended to the embeddings of the textual part of the prompt before being fed to the LLM. During inference, the model receives the speech embeddings as input and autoregressively generates the remaining fields of the prompt.

Compressed Spoken Context The only difference with full spoken context is how $Speech_Emb$ is obtained. Instead of using the entire sequences h_i , we introduce a set of N_{queries} trainable query vectors Q and compute z_i through query-based pooling using a TransformerDecoder architecture:

$$z_i = \text{TransformerDecoder}(Q, h_i)$$

$$Speech_Emb = (z_1 || z_2 || \dots || h_n)$$

In this formulation, the decoder treats Q as the target sequence and z_i as the memory. Each decoder layer first applies *self-attention* over the query tokens, allowing them to interact and share information. It then applies *cross-attention*, where the queries attend to the speech sequence z_i , extracting the most relevant aspects from it. The final output is a set of N_{queries} vectors that serve as a compressed representation of the turn. These vectors are concatenated and used in downstream dialogue modeling.

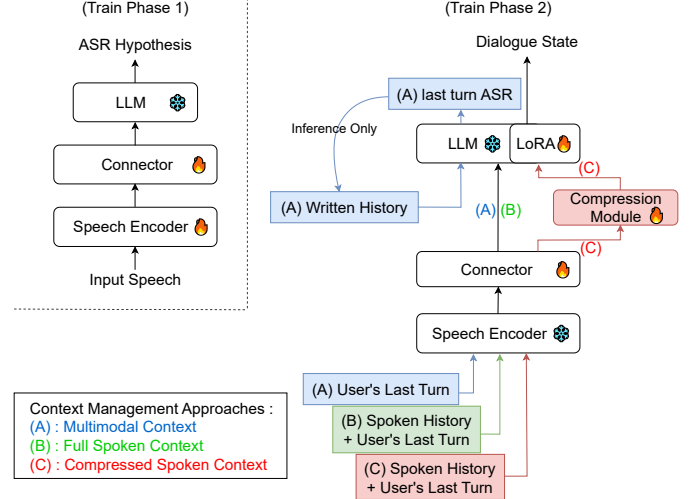


Fig. 1: An overview of our system. to the left, the ASR pretraining stage. To the right finetuning for dialog state tracking

2.2. Training

We train our models in two stages, as described in Figure 1. The first stage is ASR pre-training, where we freeze the LLM and train the speech encoder and connector to produce speech representations that align with the LLM’s input space. Specifically, we task the LLM with generating the transcription from the speech embeddings, propagating the LLM gradients back to the encoder and connector. This approach allows us to leverage the large-scale ASR datasets that are publicly available, resulting in robust alignment between the speech and text modalities.

The second stage is DST fine-tuning. In this phase, we freeze the speech encoder and train the connector, the optional compression module, and a small LoRA module for the LLM. The objective is to produce a JSON string in the format described in 2.1. Training is performed by minimizing the cross-entropy loss between the generated output and the ground-truth dialog state annotations.

3. RESULTS

3.1. Datasets

For the ASR pre-training stage, we train our model on a combination of the Loquacious Medium dataset (2,500 hours) [14], the Fisher corpus (1,960 hours) [15], and the train split from SpokenWOZ dataset (200 hours) [3]. Although SpokenWOZ does not provide ground-truth transcripts, we include it in the ASR pre-training phase because the speech encoder is frozen during DST fine-tuning, and we want the encoder to be exposed to the characteristics of SpokenWOZ data. To address the lack of transcripts on SpokenWOZ, we use Whisper-large-v3¹ [16] to generate automatic transcriptions for SpokenWOZ audio. These generated transcripts are also used later for the multimodal context method in the DST stage.

For DST fine-tuning, we primarily use the SpokenWOZ dataset for both training and evaluation. As in [10, 13] we remove the nine corrupted dialogues from the SpokenWOZ test set², and report the Joint Goal Accuracy (JGA) [17] on both the dev and test sets.

¹<https://huggingface.co/openai/whisper-large-v3>

²<https://github.com/AlibabaResearch/DAMO-ConvAI/issues/87>

3.2. Implementation details

For our component selection, we use W2v-BERT³ [18] as the speech encoder. The connector module is implemented as a single-layer Transformer encoder with a hidden dimension of 1024 and 16 attention heads. Similarly, we employ a one-layer Transformer Decoder with a hidden dimension of 1024, 16 heads, and a trainable number of queries ($N_{queries}$) as the compression module. This module is also used for attention pooling by setting $N_{queries} = 1$. For the language model, we use OLMo 2 1B⁴ [19]. We apply a LoRA adapter with a rank of 16 and an alpha value of 1, as determined by grid search. During inference, we employ beam search with 5 beams, which was also selected based on grid search results. During ASR pre-training, we use a virtual batch size of 256, a learning rate of 1×10^{-4} , and 5,000 warm-up steps. Training proceeds until the word error rate (WER) on the combined validation sets of all datasets ceases to improve. For DST fine-tuning, we maintain the same virtual batch size of 256, use a learning rate of 2×10^{-4} , and 500 warm-up steps. The model is trained until the JGA on the validation set no longer improves. All our experiments⁵ were performed using SpeechBrain toolkit⁶ [20]

Model	SWOZ test
SPACE+WavLMalign [3]	25.65%
E2E (Whisper+T5) [10]	24.10%
UBAR + GenWOZ [21]	25.90%
WavLM + conn. + OLMo-1B [13]	34.66%
Compressed Spoken Context (Ours)	36.49%
Full Spoken Context (Ours)	39.32%
WavLM + conn. + Gemma-2-9B-Instruct [13]	42.17%

Table 1: Comparison of our two best models with prior work.

3.3. Best Model Analysis

For fair comparison with prior work, the reported JGA for our model in Table 1 uses post-processing, which includes (i) canonicalizing time expressions to 24-hour format and (ii) case-insensitive fuzzy matching for open/proper-noun slots with a Levenshtein ratio ≥ 0.90 , applied symmetrically to predictions and references. Table 1 presents a comparison between published results on the SpokenWOZ test set and our two best systems: the compressed context method using 10 queries and the full spoken context method. For our systems the post-processing yields a 3 points JGA increase, which is comparable to the post-processing reported in [13]. Our approach substantially outperforms other systems of comparable size. To the best of our knowledge, the only system that surpasses our results is the Gemma-2-9B variant reported in [13]. We did not opt to train a Gemma-based variant of our model due to its high computational requirements, as our primary objective is to demonstrate the effectiveness of our method on small and compact models. Furthermore, as shown in Section 3.4, when using the same model components, our context management strategy significantly outperforms that of previous work.

To further analyze our best model, we selected the six slots with the highest error counts. In Figure 2, blue bars represent the

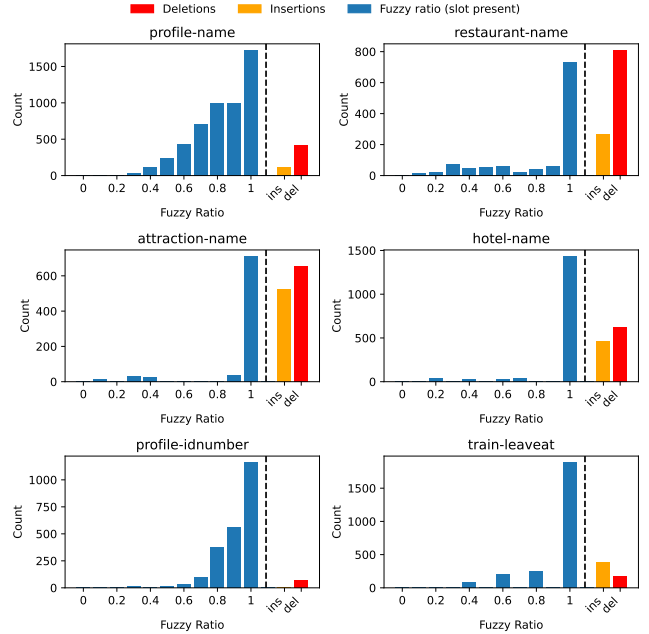


Fig. 2: Distribution of Levenshtein (fuzzy) ratios for the six most error-prone slots, with counts of insertions (orange) and deletions (red). High fuzzy ratios indicate near-correct predictions.

Levenshtein (fuzzy) ratio for slot values present in both prediction and reference, while orange and red bars indicate the counts of insertions and deletions, respectively. Most predictions achieve high fuzzy ratios (above 0.8), suggesting that when the model predicts a slot present in the reference, it usually gets the value nearly correct. Interestingly, for *restaurant-name*, *attraction-name*, and *hotel-name*, the number of substitutions (fuzzy ratio < 1) is very low, with most errors arising from insertions and deletions. This indicates that the model is generally able to correctly predict these proper nouns when it attempts them. In contrast, *profile-name*, *profile-idnumber* remain highly challenging due to their variable content and frequent spelling across multiple turns. Finally, although the error rate for *train-leaveat* is relatively low compared to its total occurrences, its high frequency means it still contributes substantially to the overall error count.

	SWOZ Dev	SWOZ Test
Multimodal Context (baseline)	31.85%	32.06%
Full Spoken Context	36.89%	36.29%
Compressed Spoken Context		
1 query	31.03%	30.99%
10 queries	34.26%	33.51%

Table 2: JGA Evaluation of different context management approaches on SpokenWOZ.

3.4. Context Management Methods Comparison

All subsequent analyses use JGA with no post processing. Table 2 shows the JGA score on SpokenWOZ dev and test splits for each

³<https://huggingface.co/facebook/w2v-bert-2.0>

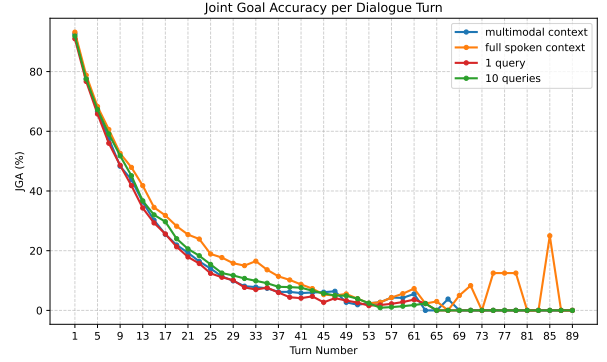
⁴<https://huggingface.co/allenai/OLMo-2-0425-1B-Instruct>

⁵The source code will be released after acceptance

⁶<https://github.com/speechbrain/speechbrain>



(a) Slot type analysis



(b) JGA per turn

Fig. 3: (a) Slot value F1 score analysis per category. (b) JGA score analysis per dialogue turn.

method. Overall, both the full spoken context and the 10-queries-per-turn methods outperformed the baseline. In particular, the full spoken context approach achieved a significantly higher JGA, demonstrating the effectiveness of leveraging the entire spoken conversation as input. The competitive performance of the 10-queries method further suggests that a substantial portion of the speech representations is redundant, and that it is possible to reduce the input size without a significant loss in performance, provided that a sufficient number of queries is used. We next provide a fine-grained comparison based on slot group and dialogue turn analyses.

Slot Group Analysis We categorize slots into four groups: categorical, time, open, and profile. Categorical slots have a fixed set of values (e.g., yes/no, area, price range). Time slots correspond to temporal expressions (e.g., departure time). Open slots can take a wide range of values such as place names, while profile slots, which are treated separately for finer analysis, contain personal information (e.g., names, IDs, emails) and are often spelled out across multiple turns. Figure 3a shows the average F1 score by slot type. All models perform well on categorical slots, with full spoken context slightly ahead. Performance drops for time and open slots, where full spoken context and 10-query compression clearly outperform the others. Profile slots are the hardest: full spoken context again leads, while the 1-query model performs worst, indicating that compressing each turn to a single embedding discards too much information.

Dialogue Turn Analysis Figure 3b displays the evolution of Joint Goal Accuracy (JGA) across dialogue turns. All models perform well in the early turns (1–5), but accuracy declines quickly in the mid turns (5–30) and approaches zero by turn 40. This drop can be attributed to the increasing length and complexity of dialogue states, combined with the strictness of the JGA metric, as well as the limited capacity of the relatively small LLM used in our experiments. The full spoken context method consistently outperforms the others, particularly during the mid turns. In the very late turns, it shows occasional performance peaks, though these are difficult to interpret given the small sample size. The 10-query attention pooling method remains competitive, but still underperforms compared to full spoken context in the late turns, even though it benefits from a much smaller context size.

3.5. Additional Experiences and Discussion

Additional Experiences To further understand the contributions of individual components and design choices in our system, we

conducted a series of ablation studies and supplementary experiments. Specifically, we investigated the impact of ASR pretraining data, the connector, the compression module, and DST data preprocessing. For ASR pretraining, we compared using the LibriSpeech dataset [22] alone versus the mixed dataset described in Section 3.1. In baseline experiments with the multimodal method, we observed that when the encoder is unfrozen during DST finetuning, the choice of ASR pretraining data has little impact. However, when freezing the encoder (which is a more practical setup for the Full/Compressed Spoken Context methods), we found that relying solely on LibriSpeech resulted in up to a 3-point drop in JGA compared to using the mixed dataset. During ASR pretraining, we also experimented with different numbers of layers (1, 2, and 4) in the encoder. We found that a single layer provided the fastest convergence and the best performance. For the compression module, we varied the number of layers and found that increasing to three layers led to a 2% absolute drop in JGA. We attribute this to the limited amount of DST finetuning data, as the compression module is only initialized at this stage. Finally, for the multimodal context method, we normalized Whisper transcripts using NeMo Inverse Text Normalization (ITN) [23], along with additional processing for time expressions. This preprocessing yielded a 1% absolute gain in JGA.

Limitations and discussion While our full spoken context approach achieves the highest performance, it could become computationally demanding for very long dialogues. The compressed context method offers a good compromise, with strong results and reduced input size. Additionally, we did not scale our experiments to larger LLMs such as Gemma-2-9B. Both directions will be explored in future work.

4. CONCLUSION

In this paper, we have proposed a fully E2E approach to Spoken Dialog State Tracking, drawing inspiration from Speech-LLMs. In contrast to traditional multimodal context approaches, we show that it is possible to use the entire spoken conversation as input (until the current turn) and achieve state-of-the-art results. We also have performed a fine-grained analysis to illustrate the causes of improvements brought by using a full spoken context: less error propagation through the dialog and better performance on the most challenging slots. In future work, a more sophisticated and compact handling of the spoken context may be explored. Moreover, scaling the used model would be a promising extension.

5. REFERENCES

- [1] David Suendermann and Roberto Pieraccini, “Slu in commercial and research spoken dialogue systems,” *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 171–194, 2011.
- [2] Jason D. Williams, Antoine Raux, and Matthew Henderson, “The Dialog State Tracking Challenge Series: A Review,” *Dialogue & Discourse*, 2016.
- [3] Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li, “SpokenWOZ: A Large-Scale Speech-Text Benchmark for Spoken Task-Oriented Dialogue Agents,” in *NeurIPS Datasets and Benchmarks Track*, 2023.
- [4] Jason D Williams, Antoine Raux, and Matthew Henderson, “The dialog state tracking challenge series: A review,” *Dialogue & Discourse*, vol. 7, no. 3, pp. 4–33, 2016.
- [5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.
- [6] Hagen Soltau, Izhak Shafran, Mingqiu Wang, Abhinav Rastogi, Wei Han, and Yuan Cao, “DSTC-11: Speech aware task-oriented dialog modeling track,” in *Proceedings of The Eleventh Dialog System Technology Challenge*. 2023, Association for Computational Linguistics.
- [7] Léo Jacqmin, Lucas Druart, Valentin Vielzeuf, Lina Maria Rojas-Barahona, Yannick Estève, and Benoît Favre, “OLISIA: a Cascade System for Spoken Dialogue State Tracking,” in *Proceedings of The Eleventh Dialog System Technology Challenge*. 2023, Association for Computational Linguistics.
- [8] Deyuan Wang, Tiantian Zhang, Caixia Yuan, and Xiaojie Wang, “Joint modeling for asr correction and dialog state tracking,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [9] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic, “Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling,” in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [10] Lucas Druart, Valentin Vielzeuf, and Yannick Estève, “Is one brick enough to break the wall of spoken dialogue state tracking?,” *arXiv preprint arXiv:2311.04923*, 2023.
- [11] Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al., “Wavchat: A survey of spoken dialogue models,” *arXiv preprint arXiv:2411.13577*, 2024.
- [12] Haitian Lu, Gaofeng Cheng, Liuping Luo, Leying Zhang, Yanmin Qian, and Pengyuan Zhang, “Slide: Integrating speech language model with llm for spontaneous spoken dialogue generation,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [13] Šimon Sedláček, Bolaji Yusuf, Ján Švec, Pradyoth Hegde, Santosh Kesiraju, Oldřich Plchot, and Jan Černocký, “Approaching dialogue state tracking via aligning speech encoders and llms,” *arXiv preprint arXiv:2506.08633*, 2025.
- [14] Titouan Parcollet, Yuan Tseng, Shucong Zhang, and Rogier van Dalen, “Loquacious set: 25,000 hours of transcribed and diverse english speech recognition data for research and commercial use,” 2025.
- [15] Christopher Cieri, David Miller, and Kevin Walker, “The fisher corpus: A resource for the next generations of speech-to-text,” in *LREC*, 2004, vol. 4, pp. 69–71.
- [16] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28492–28518.
- [17] Victor Zhong, Caiming Xiong, and Richard Socher, “Global-locally self-attentive encoder for dialogue state tracking,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.
- [18] Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenhaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al., “Seamless: Multilingual expressive and streaming speech translation,” 2023.
- [19] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al., “2 olmo 2 furious,” 2025.
- [20] Mirco Ravanelli, Titouan Parcollet, Adel Moumen, Sylvain de Langen, Cem Subakan, Peter Plantinga, et al., “Open-source conversational ai with speechbrain 1.0,” *Journal of Machine Learning Research*, vol. 25, no. 333, 2024.
- [21] Haris Gulzar, Monikka Roslianna Busto, Akiko Masaki, Takeharu Eda, and Ryo Masumura, “Leveraging llms for written to spoken style data transformation to enhance spoken dialog state tracking,” in *Proc. Interspeech 2025*, 2025, pp. 1743–1747.
- [22] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [23] Yang Zhang, Evelina Bakhturina, Kyle Gorman, and Boris Ginsburg, “Nemo inverse text normalization: From development to production,” *arXiv preprint arXiv:2104.05055*, 2021.