The Speech-LLM Takes It All: A Truly Fully End-to-End Spoken Dialog State Tracking Approach

Nizar El Ghazal[*], **Antoine Caubrière, Valentin Vielzeuf**
Orange Research
4 Rue du Clos Courtel, 35510 Cesson-Sévigné, France
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.

Keywords: Speech-LLM, SpokenDST, Multimodal, Context Propagation

1. Introduction

Dialog State Tracking (DST) is a vital component in task-oriented dialog (TOD) systems (Suendermann and Pieraccini, 2011; Williams et al., 2016), 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 (Si et al., 2023).

One of the most common recent approaches for spoken DST is the cascade system. The process generally begins with an Automatic Speech Recognition (ASR) module that transcribes spoken language into text. This is often followed by an ASR correction module to improve the accuracy of the transcription, and then a written DST component, which is frequently based on models such as *T5* (Raffel et al., 2020). This pipeline approach leverages the strengths of existing text-based DST models. And it was notably popular in the DSTC-11 challenge (Soltau et al., 2023), where a variant was used by the winning system, OLISIA (Jacqmin et al., 2023).

Despite its success, the cascade approach encounters inherent limitations. It is highly vulnerable to error propagation originating from the initial ASR stage. This can significantly affect the overall accuracy and reliability of the entire system (Wang et al., 2023). 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 (Budzianowski et al., 2018).

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, (Druart et al., 2023) demonstrated the effectiveness of E2E approaches, particularly in fully spoken contexts without access to ground-truth transcriptions, such as the SpokenWOZ (Si et al., 2023) dataset. In these settings, E2E models have been shown to outperform traditional cascade systems.

Concurrently, speech-aware large language models (speechLLMs), which are also considered E2E systems, have gained increasing popularity in a variety of spoken language tasks, including ASR and response generation (Ji et al., 2024; Lu et al., 2025). Recent work (Sedláček et al., 2025) applied speech-aware LLMs to the spoken DST task, achieving state-of-the-art performance in the SpokenWOZ dataset (Si et al., 2023).

A notable advantage of E2E systems is their remarkable flexibility when it comes to managing context, as they are capable of seamlessly integrating both written and spoken information within a single framework. For example, in recent studies such as (Druart et al., 2023) and (Sedláček et al., 2025), both approaches utilize the spoken representation of the user's last turn to inform the system's response.

However, they differ significantly in how they handle the rest of the context: the former combines the spoken user turn with the written previous state, while the latter integrates it with the written representations of all previous turns, providing a large and more comprehensive context. This difference in approach raises an important and intriguing ques-

[*] The author performed the work while at an internship

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2629–2637
11–16 May 2026.

tion.

What would happen if we relied solely on spoken context? Specifically, what would be the effects of feeding the system the speech representations for the entire conversation, or alternatively, condensing these spoken representations using an intermediate module before processing?

In this paper, we explore these possibilities for context management when using a Speech-LLM model. Our contributions are three-fold.

- We validate the use of Speech-LLMs as an accurate approach for spoken DST.
- We propose two context management approaches reaching the SOTA.
- 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 dialog (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 dialog turns U_1, A_2, ..., A_{t-1}, U_{t-1}, our goal is to predict a set of k relevant domains ($domain_1$, $domain_2$, ..., $domain_k$) and n slot-value pairs ($slot_1 = value_1$, $slot_2 = value_2$, ..., $slot_n = value_n$), which are then represented as a JSON structure.

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, by stacking every 6 successive frames and concatenating their embeddings, reducing the sequence length and increasing the feature dimension. They are then 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 dialog state in an auto-regressive manner.

2.1. Context Management

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

Multimodal Context Following (Sedláček et al., 2025), we provide as input the spoken user utterance U_n^{spoken} and the written dialog history together. The model then predicts the transcription of the user's utterance U_n^{text}, the active domains D_n and the dialog state S_n. The LLM is trained on the prompt:

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

where we have:

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

$$h_n = Connector\big(Encoder(U_n)\big)$$

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 dialog state S_n. The prompt employed for this strategy is:

Speech_Emb {"domains": D_n *,"predicted_state":* S_n *}*

where :

$$Context_n = (U_1^{\text{spoken}}, A_2^{\text{spoken}}, \ldots, U_n^{\text{spoken}})$$
$$h_{2i+1} = Connector\big(Encoder(U_{2i+1})\big)$$
$$h_{2i} = Connector\big(Encoder(A_{2i})\big)$$
$$Speech\_Emb = (h_1||h_2||\ldots||h_n)$$

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 auto-regressively 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

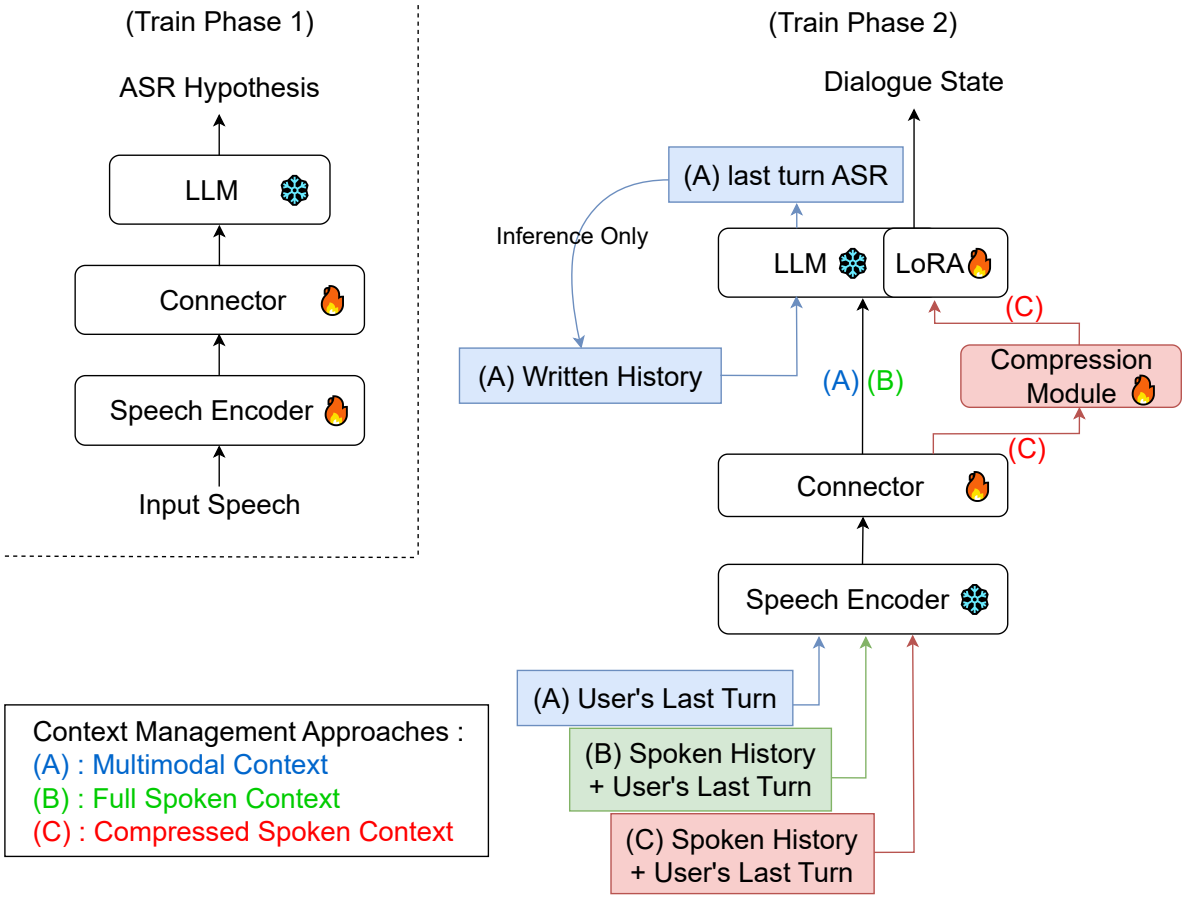


Figure 1: An overview of our system. to the left, the ASR pretraining stage. To the right finetuning for dialog state tracking

pooling using a TransformerDecoder architecture:

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

$$Speech\_Emb = (z_1||z_2||\ldots||h_n)$$

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 dialog modeling.

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) (Parcollet et al., 2025), the Fisher corpus (1,960 hours) (Cieri et al., 2004), and the train split from SpokenWOZ dataset (200 hours) (Si et al., 2023). 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-v3[1] (Radford et al., 2023) 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 (Druart et al., 2023; Sedláček et al., 2025) we remove the nine corrupted dialogs from the SpokenWOZ test set[2], and report the Joint Goal Accuracy (JGA) (Zhong et al., 2018) on both the dev and test sets.

3.2. Implementation details

For our component selection, we use W2v-BERT [3] (Barrault et al., 2023) 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 1B[4] (OLMo et al., 2025). We apply a LoRA adapter with a rank of 16 and an alpha value of 1, as determined by grid search. Note that for the specific case of scaling experiments, we follow the protocol of (Sedláček et al., 2025) and we therefore train new variants of our systems using W2V-BERT-2 as the speech encoder and Gemma2-9B-Instruct[5]as the LLM, while keeping the same connector. The training ASR datasets remained unchanged, but we used the augmented Speech Aware MultiWOZ (Soltau et al., 2023) with SpokenWOZ for model training, followed by a final fine-tuning step on SpokenWOZ alone.

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 experiments[6] were performed using SpeechBrain toolkit[7] (Ravanelli et al., 2024)

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 (Sedláček et al., 2025).

Our approach substantially outperforms other systems of comparable size. Note that we distinguish two different setups. (a) For the upper part of Table 1, we focus on approaches using LLMs with known and identified training data, guaranteeing the absence of test data contamination. And (b) for the bottom part, we report performance for sake of comparison with (Sedláček et al., 2025), which is using Gemma2-9B LLM. As this model training set is not known and according to (Sedláček et al., 2025), we can't be sure that there is no overlap with the test data of SpokenWOZ. Moreover, the number of parameters is larger than for upper lines methods, allowing to draw comparison between approaches for larger scale models. Note that the training data of these larger models in the specific DST step are following the protocol of (Sedláček et al., 2025), provided in the implementation details subsection 3.2.

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

[1] https://huggingface.co/openai/whisper-large-v3

[2] https://github.com/AlibabaResearch/DAMO-ConvAI/issues/87

[3] https://huggingface.co/facebook/w2v-bert-2.0

[4] https://huggingface.co/allenai/OLMo-2-0425-1B-Instruct

[5] https://huggingface.co/google/gemma-2-9b-it

[6] https://github.com/nizar139/SpokenDST

[7] https://github.com/speechbrain/speechbrain

Model	SpokenWOZ test
SPACE+WavLMalign (Si et al., 2023)	25.65%
E2E (Whisper+T5) (Druart et al., 2023)	24.10%
UBAR + GenWOZ (Gulzar et al., 2025)	25.90%
WavLM + conn. + OLMo-1B (Sedláček et al., 2025)	34.66%
Compressed Spoken Context (Ours)	36.49%
Full Spoken Context (Ours)	**39.32%**
WavLM + conn. + Gemma2-9B-Instruct (Sedláček et al., 2025)	42.17%
Compressed Spoken Context + Gemma2-9B-Instruct (Ours)	43.16%
Full Spoken Context + Gemma2-9B-Instruct (Ours)	**45.52%**

Table 1: Comparison of our approaches with prior works in two different setups: using open data models (upper part) and using potentially data-contaminated models such as Gemma2-9B-Instruct (bottom). JGA are reported with post processing as in (Sedláček et al., 2025)

.

it attempts them. In contrast, profile-related slots (e.g., `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.

3.4. Context Management Methods Comparison

All subsequent analyses use JGA without post-processing. Table 2 shows the JGA score on SpokenWOZ dev and test splits for each 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 dialog 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 per-

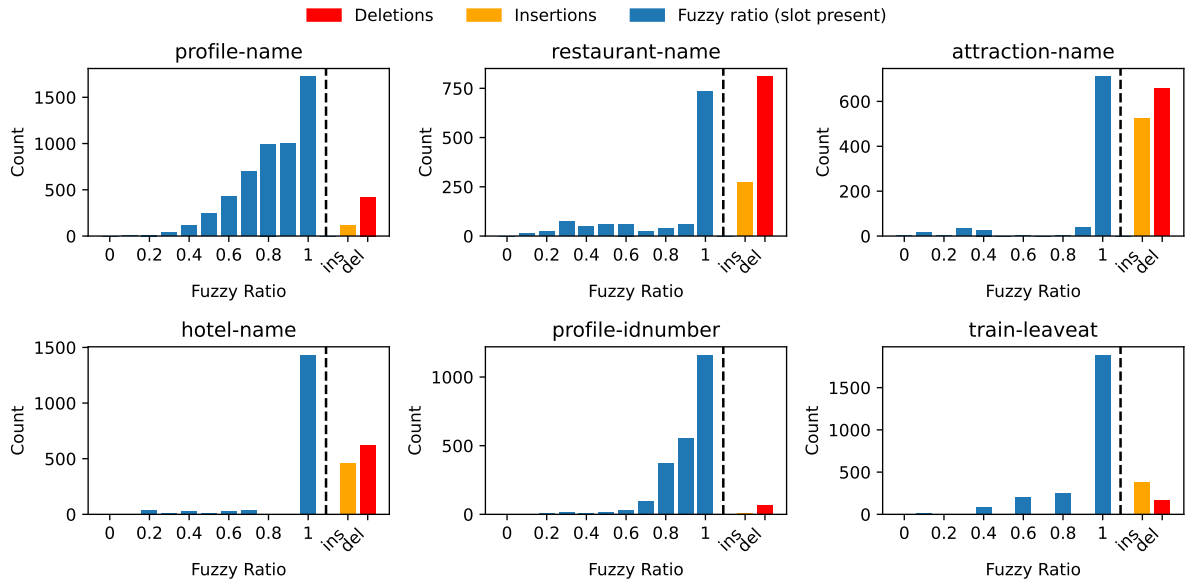
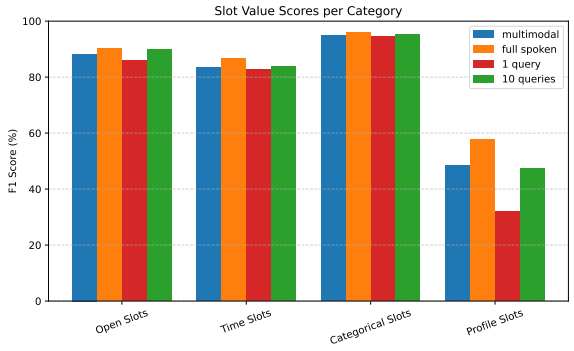


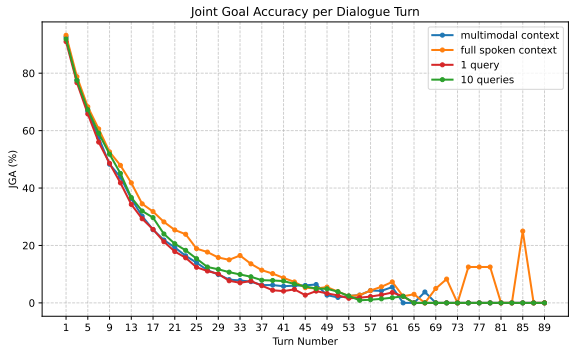
Figure 2: Distribution of Levenshtein (fuzzy) ratios for the six most error-prone slots, with counts of insertions (orange) and deletions (red). High fuzzy ratios indicate near-correct predictions.

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

Table 2: JGA Evaluation of different context management approaches on SpokenWOZ. Contrary to Table 1, no post-processing is applied.



(a) Slot type analysis



(b) JGA per turn

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

sonal 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.

Dialog Turn Analysis Figure 3b displays the evolution of Joint Goal Accuracy (JGA) across dialog 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 dialog 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

Scaling model and data Table 3 shows the scaling results. We note that for the full spoken context on Gemma2-9B, we decided to introduce context truncation, limiting the total speech-embedding tokens passed to the LLM to 1500 at most, mainly in order to run the experiments with the available resources. We observe that scaling the amount of train data is beneficial in all our setups (except the full spoken context with OLMo2-1B). However, the Full Spoken Context method benefits less compared to the Compressed Spoken Context. One possible reason for this is that the Transformer-Decoder introduced with this method is only initialized at the DST-finetuning stage, and SpokenWOZ alone is not able to fully train it to compress the context efficiently, which means that more data are beneficial for it, the same way as with the connector, which greatly benefits from more data during the ASR-pretraining. We also observe that the bigger LLM gets more benefit from scaling the training data, which is expected since a bigger model needs more training data. Finally, our Speech-LLM model with compressed spoken context is slightly ahead compared to the current SOTA while using a much smaller context size, which means less resources (especially since the attention scales quadratically with the context size). Finally, our Speech-LLM setup with the (truncated) full spoken context significantly outperforms the Gemma2-9B variant from (Sedláček et al., 2025), establishing the new SOTA for Spoken DST on SpokenWOZ.

Impact of the number of queries Figure 4 shows the impact of the number of queries on performance. We observe that the JGA scores increase significantly when increasing the number of queries from 1 to 3, but remain pretty similar for higher number of queries. we also 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

	LLM	Train Data	SpokenWOZ Test	
			JGA	JGA+PP
Full Spoken Context	OLMo2-1B	SpokenWOZ	36.29%	39.32%
		Combined + SW ft	36.23%	39.14%
	Gemma2-9B	SpokenWOZ	39.62%	43.12%
		Combined	40.06%	43.74%
		Combined + SW ft	**41.99%**	**45.52%**
10 queries Compressed Spoken Context	OLMo2-1B	SpokenWOZ	33.51%	36.49%
		Combined + SW ft	35.16%	38.59%
	Gemma2-9B	SpokenWOZ	35.94%	39.13%
		Combined	37.01%	40.15%
		Combined + SW ft	39.53%	43.16%
WavLM + conn. + Gemma2-9B-Instruct			38.76%	42.17%

Table 3: JGA performance of our scaled models, JGA+PP uses the post-processing.

data, as the compression module is only initialized at this stage.

Other ablations To better 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 (Panayotov et al., 2015) 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.

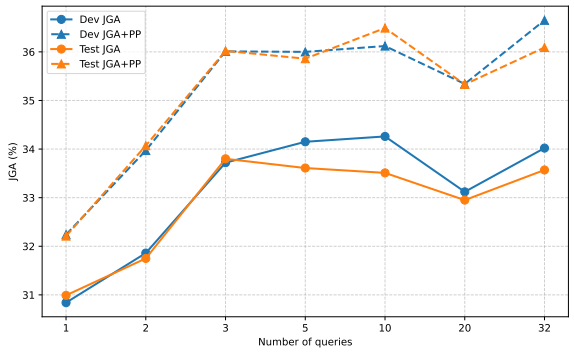


Figure 4: Impact of the number of queries on the JGA scores of the compressed spoken context systems.

Finally, for the multimodal context method, we normalized Whisper transcripts using NeMo Inverse Text Normalization (ITN) (Zhang et al., 2021), along with additional processing for time expressions. This preprocessing yielded a 1% absolute gain in JGA.

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 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.

At the same time, using the full spoken history entails practical trade-offs. Most notably, the increased memory usage and latency motivate the design of more efficient context representations. 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. We further plan to investigate streaming inference and integration within production pipelines. We also are interested in enlarging the scope of the evaluation beyond SpokenWOZ, as additional human-spoken DST benchmarks may become available. Finally, we aim to release code and trained models to facilitate reproducibility.

5. Limitations

While our experiments show that providing the full spoken history is beneficial for spoken DST, some limitations need to be mentioned. First, the memory and compute footprint of the full-spoken setting grows with dialog length. Indeed, each additional spoken turn expands the input sequence to process. It then increases attention cost (quadratically) and thus end-to-end latency. This problem is especially noticeable on long dialogs or when having production-level constraints on hardware (GPU memory can become a bottleneck even with relatively small context windows). We can note that our proposed compression variant helps tackle this issue, yet it may imply the loss of some fine-grained cues, such as named entities, numbers, or accurate time slots).

Then our models are effective enough for an offline research environment. Yet, they remain far from a delivery-level. Real-world TOD systems often require to enable streaming (e.g. partial hypotheses, incremental state updates) with small latency. The scope of the paper was not including the end-to-end latency and the failure modes under streaming constraints, which are all prerequisites for reliable deployment and may be included in future work.

Finally, our empirical analysis is focused on SpokenWOZ. To the best of our knowledge, it is the main benchmark built from human conversational speech for DST. We may extend the study to other benchmarks such as Spoken MultiWOZ, but it would require care in interpretation. This kind of corpora is mainly composed of synthesized voice for training and validation. Human speech is only in evaluation splits. So a large part of the dataset contains acoustic characteristics that are very different from fully human-spoken data. Thus, performance measured on such datasets may reflect robustness to TTS artifacts as much as to real conversational variability.

Future works may help to address these three limitations. We may explore more memory-efficient context representations (such as hierarchical or selective retention), dig on streaming compatible inference, and enlarge the evaluation scope to synthesized corpora with great care for acoustic quality.

6. References

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. 2023. Seamless: Multilingual expressive and streaming speech translation.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. 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)*.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: A resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71.

Lucas Druart, Valentin Vielzeuf, and Yannick Estève. 2023. Is one brick enough to break the wall of spoken dialogue state tracking? *arXiv preprint arXiv:2311.04923*.

Haris Gulzar, Monikka Roslianna Busto, Akiko Masaki, Takeharu Eda, and Ryo Masumura. 2025. Leveraging llms for written to spoken style data transformation to enhance spoken dialog state tracking. In *Proc. Interspeech 2025*, pages 1743–1747.

Léo Jacqmin, Lucas Druart, Valentin Vielzeuf, Lina Maria Rojas-Barahona, Yannick Estève, and Benoît Favre. 2023. OLISIA: a Cascade System for Spoken Dialogue State Tracking. In *Proceedings of The Eleventh Dialog System Technology Challenge*. Association for Computational Linguistics.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, et al. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Haitian Lu, Gaofeng Cheng, Liuping Luo, Leying Zhang, Yanmin Qian, and Pengyuan Zhang. 2025. 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)*, pages 1–5. IEEE.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2025. 2 olmo 2 furious.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Titouan Parcollet, Yuan Tseng, Shucong Zhang, and Rogier van Dalen. 2025. Loquacious set: 25,000 hours of transcribed and diverse english speech recognition data for research and commercial use.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Mirco Ravanelli, Titouan Parcollet, Adel Moumen, Sylvain de Langen, Cem Subakan, Peter Plantinga, et al. 2024. Open-source conversational ai with speechbrain 1.0. *Journal of Machine Learning Research*, 25(333).

Šimon Sedláček, Bolaji Yusuf, Ján Švec, Pradyoth Hegde, Santosh Kesiraju, Oldřich Plchot, and Jan Černockỳ. 2025. Approaching dialogue state tracking via aligning speech encoders and llms. *arXiv preprint arXiv:2506.08633*.

Shuzheng Si, Wentao Ma, Haoyu Gao, Yuchuan Wu, Ting-En Lin, Yinpei Dai, Hangyu Li, Rui Yan, Fei Huang, and Yongbin Li. 2023. SpokenWOZ: A Large-Scale Speech-Text Benchmark for Spoken Task-Oriented Dialogue Agents. In *NeurIPS Datasets and Benchmarks Track*.

Hagen Soltau, Izhak Shafran, Mingqiu Wang, Abhinav Rastogi, Wei Han, and Yuan Cao. 2023. DSTC-11: Speech aware task-oriented dialog modeling track. In *Proceedings of The Eleventh Dialog System Technology Challenge*. Association for Computational Linguistics.

David Suendermann and Roberto Pieraccini. 2011. Slu in commercial and research spoken dialogue systems. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pages 171–194.

Deyuan Wang, Tiantian Zhang, Caixia Yuan, and Xiaojie Wang. 2023. Joint modeling for asr correction and dialog state tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse*.

Yang Zhang, Evelina Bakhturina, Kyle Gorman, and Boris Ginsburg. 2021. Nemo inverse text normalization: From development to production. *arXiv preprint arXiv:2104.05055*.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.