SENS-ASR: Semantic Embedding injection in Neural-transducer for Streaming Automatic Speech Recognition

Youness Dkhissi[1,2], **Valentin Vielzeuf**[1], **Elys Allesiardo**[1], **Anthony Larcher**[2]
1. Orange Innovation, 4 Rue du Clos Courtel, 35510 Cesson-Sévigné, France
2. LIUM, Le Mans Université Avenue Olivier Messiaen,72085 Le Mans CEDEX 9, France
{youness.dkhissi, valentin.vielzeuf, elys.allesiardo}@orange.com, anthony.larcher@univ-lemans.fr

Abstract

Many Automatic Speech Recognition (ASR) applications require streaming processing of the audio data. In streaming mode, ASR systems need to start transcribing the input stream before it is complete, i.e., the systems have to process a stream of inputs with a limited (or no) future context. Compared to offline mode, this reduction of the future context degrades the performance of Streaming-ASR systems, especially while working with low-latency constraint. In this work, we present **SENS-ASR**, an approach to enhance the transcription quality of Streaming-ASR by reinforcing the acoustic information with semantic information. This semantic information is extracted from the available past *frame-embeddings* by a context module. This module is trained using knowledge distillation from a sentence embedding Language Model fine-tuned on the training dataset transcriptions. Experiments on standard datasets show that SENS-ASR significantly improves the Word Error Rate on small-chunk streaming scenarios.

Keywords: streaming automatic speech recognition, semantic representation, neural transducer, large language model.

1. Introduction

In recent years, End-to-End (E2E) ASR approaches such as Connectionist Temporal Classification (CTC) (Graves et al., 2006), Attention Encoder-Decoder (AED) (Chan et al., 2016; Radford et al., 2023) and Recurrent Neural Network Transducer (RNN-T) (Graves, 2012) have gained a lot of popularity compared to hybrid models (Bourlard and Morgan, 2012), especially due to the emergence of transformer-based architectures (Vaswani, 2017). These approaches give great results in terms of transcription quality when having access to the full speech audio as input. However, in Streaming-ASR (Variani et al., 2022) where models should begin transcribing without having the full speech context, the deployment of E2E models presents challenges. Most of the proposed approaches suffer from severe performance degradation as they use causal masking to learn transcribing without relying on the future context (Moritz et al., 2021; Wang and Xu, 2024).

Among the works that try to tackle the problem of transcription quality degradation in Streaming-ASR, (Yu et al., 2021b) proposed a streaming-offline process to teach a unique model to transcribe with and without having the full speech context. When operating in streaming mode, this approach enhances the frame representations generated by the encoder with information gathered from the model's offline training, which utilizes the full context.

Other works change the causal attention mask into a chunk-wise attention mask that enables the model to benefit from a full attention operation on well-defined chunks of frames, rather than being limited to causal attention (Gulzar et al., 2023; Wang and Xu, 2024). This masking technique improves the predictions of the model, especially for the frames located at the beginning of each chunk. Nevertheless, this approach does not compensate for the lack of future context when processing the last frames of each chunk. This drawback motivated the work of (Zeineldeen et al., 2024) in which a chunk-wise attention mechanism with lookahead enables the model to take into account some frames of the future chunk (i.e., part of the future context) when encoding the frames from the current chunk. However, this adaptation increases the latency induced by waiting for extra frames as well as the computational cost due to the redundancy of encoding frames used in adjacent chunks. These drawbacks encourage other works to generate a simulated lookahead based on past frames rather than using a true one (An et al., 2022; Zhao et al., 2024).

All of these previous works try to improve Streaming-ASR models based on acoustic features only. In fact, (Choi et al., 2024) and (Sanabria et al., 2023) show that the embeddings generated for the audio frames mostly embed acoustic information rather than semantic information. In addition, (Kim et al., 2021) indicates that RNN-T, in particular, has poor results in modeling long-range linguistic information. For this reason, the existing approaches use rescoring methods based on External Language Models during inference to address the lack of semantic information in the embeddings generated by the encoder and thus improve the

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10233–10241
11–16 May 2026.

transcription quality.

Other recent works try to exploit Large Language Models for the ASR task (Chen et al., 2024) and also in streaming context (Tsunoo et al., 2024). However, using some of these LLMs in the core of ASR architecture brings many doubts about their real effectiveness, as they are evaluated on public test datasets whose transcriptions could potentially be used in the training of these LLMs. These leaks are found and confirmed on many Natural Language Processing datasets (Balloccu et al., 2024; Xu et al., 2024), which increases the possibility that transcriptions of public ASR datasets have been used in the training of some LLMs. A recent work (Tseng et al., 2025) shows that a substantial amount of the LibriSpeech (Panayotov et al., 2015) and Common Voice (Ardila et al., 2020) evaluation sets, which are amongst the most used datasets for speech recognition, appear in public LLM pretraining corpora. This specifically questions the results and the improvements obtained by LLM based models on these datasets.

We propose **SENS-ASR**, a novel framework that directly addresses the semantic deficiency of Streaming-ASR by injecting semantic information into the encoder's *frame-embeddings*. Unlike prior approaches that treat acoustic and linguistic modeling as separate or loosely coupled components, SENS-ASR introduces a dedicated context module that operates in real time, generating semantic embeddings from the history of past acoustic frames. This module is trained via knowledge distillation from a sentence embedding language model, which is itself fine-tuned on the target ASR domain to ensure relevance and robustness.

The core idea behind SENS-ASR is to bridge the gap between local acoustic features and global semantic context, enabling the encoder to produce frame representations that are both acoustically and semantically informative. By enriching each frame with semantic context derived from the preceding audio, SENS-ASR empowers the decoder to make more accurate and coherent predictions, even under strict streaming constraints.

Overall, our paper comes with the following contributions: **(a)** a transducer model equipped with an additional context module, designed to inject semantic information into the representation of the frames and **(b)** a finetuning protocol of the sentence embedding allowing us to then train the context module.

2. Proposed method

Figure 1 shows the training process and the new components applied to an RNN-T model. **SENS-ASR** operates in two stages: at training stage, a *context module* learns to model the semantic context by distillation from a teacher Sentence Embedding Model; at inference stage, this module extracts semantic information to enrich the acoustic *frame-embeddings*.

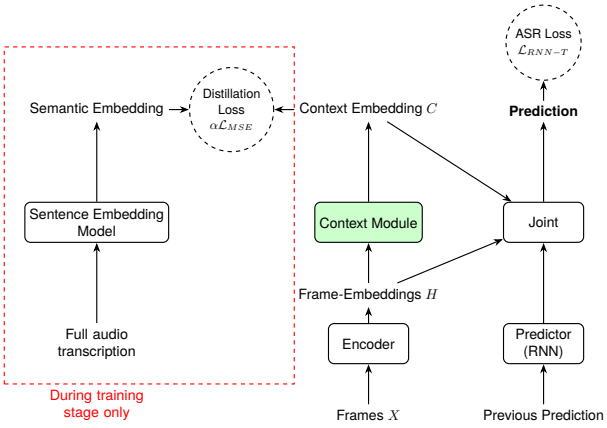

Figure 1: Architecture of the SENS-ASR system using an RNN-T model and a context module. Components in the red dashed-rectangle are only used during training. Components in dashed-circles are the parts of the system global loss.

2.1. Semantic Context

Let $X = \{x_i^{(\gamma)}\}$ be the sequence of frames we input per chunk into the ASR encoder, with $x_i^{(\gamma)}$ being the frame i in the chunk γ. All chunks have a fixed length s. The ASR encoder generates a *frame-embedding* for each frame, $x_i^{(\gamma)}$, using a chunk-wise attention. In the decoding step, **SENS-ASR** injects, in each *frame-embedding*, a semantic information extracted from the past-context of this frame. Each *frame-embedding*, $h_i^{(\gamma)}$, is computed as a function of all frames from past chunks 0 to $\gamma - 1$ plus all frames from the current chunk γ.

$$\begin{cases} H = Encoder(X) \\ h_i^{(\gamma)} = f(x_0^{(0)}, \ldots, x_{s-1}^{(\gamma)}) \end{cases} \quad (1)$$

To inject semantic information into the decoding process, past *frame-embeddings* are processed by the context-module to generate one context-embedding, $C_i^{(\gamma)}$, for each $h_i^{(\gamma)}$. To reduce the computational complexity, a unique context embedding $C^{(\gamma)}$ is computed for each chunk, γ, using the past P chunks.

$$C^{(\gamma)} = Context\_Module(x_0^{(\gamma-P)}, \ldots, x_{s-1}^{(\gamma-1)}) \quad (2)$$

This module is trained to have an output similar to that of the Sentence Embedding Model. This model takes the transcription of the full audio and generates a sentence embedding of this transcription.

This context-embedding can be computed by techniques such as max, average or attention pooling. Based on (Chen et al., 2018), we use attention pooling in our semantic context module. More precisely, we apply successive cross-attention operations on *frame-embeddings* to produce a single vector.

After computing the semantic context embedding for a chunk, γ, we concatenate it with each *frame-embedding*, $h_i^{(\gamma)}$, of this chunk before passing them to the joint network together with predictor network output to generate the prediction as described in (Graves, 2012).

We train our model using the following loss:

$$\mathcal{L}_{SENS-ASR} = \mathcal{L}_{RNN-T} + \alpha \mathcal{L}_{MSE} \quad (3)$$

where $\mathcal{L}_{RNN-T}$ is the standard transducer loss for the ASR task, α is a scalar and $\mathcal{L}_{MSE}$ is the Mean Square Error loss used to train the context module to mimic the semantic embedding generated by the Teacher Sentence Embedding Model.

2.2. Teacher Sentence Embedding Model fine-tuning

To improve the quality of the semantic information extracted in our application domain, it is necessary to fine-tune the teacher Sentence Embedding Model that will guide the training of our context module. This involves generating pairs of sentences by paraphrasing the transcriptions from the training dataset, which will be used to fine-tune the Language Model with the objective of reinforcing the similarity between embeddings of these pairs. To avoid neural collapse (Papyan et al., 2020), we design a dataset with positive and negative pairs of sentences.

2.2.1. Paraphrasing protocol

To perform fine-tuning, we create a set of text pairs (sentence A, sentence B) where sentence A is the transcription of an audio segment, while sentence B is a paraphrase artificially generated. Paraphrases can be generated by lexical replacement, back-translation (Liu et al., 2020) or by using Language Models (small ones fine-tuned for paraphrase generation or even LLMs). We choose to use LLM paraphrasing as other techniques generate text which is not far enough from the original in terms of information order and vocabulary.

However, LLMs could hallucinate, i.e., rewrite the original text in addition to the proposed paraphrase or respond to the content of the original text if it is truncated or in a question form. To avoid these hallucinations, as described in Figure 2, we generate multiple paraphrases and filter them according to two criteria: i) we discard sentences whose BERTScore with the original text is less than 0.5 (Zhang* et al., 2020); ii) paraphrases that are too long with respect to the original text (2x longer or more) are also discarded. In the end, the paraphrase is randomly chosen among the remaining sentences.

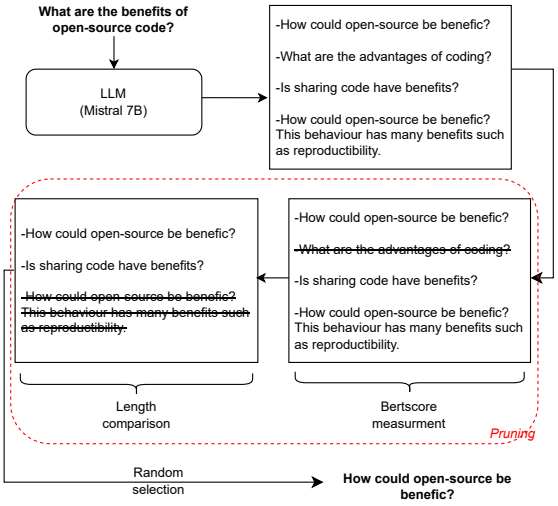

Figure 2: Example of the proposed LLM paraphrasing process. The text in **bold** refers to the input and the output, while the strikethrough text represents the rejected paraphrases in each pruning step.

We could not control these hallucinations while paraphrasing truncated sentences. That is why we choose to apply the fine-tuning of the Teacher Sentence Embedding Model using full audio transcription, which is more challenging for the context module to mimic. Fortunately, this choice is not an obstacle to have an efficient training as shown in Section 3.3.

2.2.2. Avoiding neural collapse during fine-tuning

In the previous section, we built positive pairs of sentences that share similar semantic content. We now create negative pairs by associating an original text transcribed from an audio utterance with the transcription of another audio utterance or its paraphrase.

We create triplets (sentence A, sentence B, *label*) where *label* is a Cosine Similarity objective between sentence A and sentence B which is randomly chosen as follows:

- $label \sim \mathcal{U}(0.8, 1)$ if (sentence A, sentence B) is a positive pair.
- $label \sim \mathcal{U}(-0.2, 0.2)$ if (sentence A, sentence B) is a negative pair.

In our training corpus, each speaker tends to talk about a limited number of topics. Thus, associating transcriptions from a single speaker in a negative

pair would induce a bias. We constrain our negative pairs to come from different speakers.

Positive pairs represent $\frac{2}{3}$ of the fine-tuning dataset, while the remaining $\frac{1}{3}$ represents the portion of negative pairs. These proportions are chosen to optimize the model training while avoiding neural collapse.

2.3. Dynamic Chunk Training

Dynamic Chunk Training (DCT) (Zhang et al., 2020) is a technique designed to enable an Automatic Speech Recognition (ASR) model to operate effectively in both streaming and offline modes by exposing it to a variety of context lengths during training. This approach relies on constructing a mask matrix $M \in \{0,1\}^{T\times T}$, where T is the total number of frames in the input sequence. Each element $m_{t,u}$ of the mask indicates whether, at time step t, the model is permitted to attend to the input at time u.

During training, a chunk size S is sampled randomly from the interval $[1, T]$. To control the amount of past context the model can attend to, the left context is limited to P chunks, with P sampled from the set $\{0, \lceil T/S \rceil\}$. The mask matrix M is then constructed such that:

$$m_{t,u} = \begin{cases} 1, & \text{if } \left\lfloor \frac{t}{S} \right\rfloor - P \leq \left\lfloor \frac{u}{S} \right\rfloor \leq \left\lfloor \frac{t}{S} \right\rfloor, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Where $\lfloor \frac{t}{S} \rfloor$ is the chunk index of t and $\lfloor \frac{u}{S} \rfloor$ is the chunk index of u.

This formulation ensures that, at each time step t, the model's attention is confined to the current chunk and a limited number of previous chunks, effectively controlling the computational complexity of the attention mechanism. When $S = T$, the mask reduces to all ones, allowing the model to attend to the entire sequence, which corresponds to offline processing. Conversely, smaller values of S simulate streaming conditions with limited context, enabling the model to generalize across both streaming and non-streaming scenarios.

In practice, this dynamic chunking strategy is integrated into the training process by randomly selecting S and P for each batch, thereby exposing the model to a diverse range of context lengths. This approach facilitates the development of models capable of flexible inference, maintaining high accuracy while reducing computational costs during decoding.

3. Experiments

3.1. Evaluation protocol

We conduct our experiments on two widely used public datasets: LibriSpeech (Panayotov et al., 2015) that contains 960 hours of read English speech and TEDLIUM-2 (Zhou et al., 2020) that contains 207 hours of TED Talks in order to show the effectiveness of our approach on more spontaneous speech.

To evaluate our approach on the Streaming-ASR task, we use the Word Error Rate metric. Also, to show the significance of the results, each result is presented together with its confidence interval[1]. These intervals are calculated using the bootstrapping method with 1,000 bootstrap sets. They were also calculated between 2.5 and 97.5 percentiles to exclude outliers.

3.2. Model configuration

All experiments[2] were conducted using the SpeechBrain toolkit (Ravanelli et al., 2021). The baseline used for Streaming-ASR task and our proposed system are Recurrent Neural Network Transducer (RNN-T (Graves, 2012)) which share the following components:

- **2 Convolutional layers** with kernel size of 2 and stride of 2, which downsamples the frame rate by 4.
- **12-layer Conformer encoder** (Gulati et al., 2020) having 512-dimensional input where each layer is composed of: feed-forward network of size 2048, convolution bloc having kernel size of 31 with stride of 1 and a self-attention bloc with 8 attention heads.
- **Predictor network** of 1-layer LSTM (Graves and Graves, 2012) with hidden size of 512.
- **Joint network** where a projection of dimension 640 is applied to the encoder output and another to the predictor output. Then we sum the 2 resulting vectors.

Our proposed SENS-ASR architecture includes an additional **Context module** composed of an attention pooling of a 3-layer transformer decoder followed by a linear projection of dimension 768 to match the output of the teacher Sentence Embedding Model.

During the training stage, the RNN-T loss with Fastemit regularization (Yu et al., 2021a) ($\lambda = 0.006$) is used to optimize the model latency. The distillation loss has a weight of $\alpha = 0.2$. This value

[1] https://github.com/luferrer/ConfidenceIntervals
[2] https://github.com/Orange-OpenSource/sens-asr

Table 1: the Word Error Rate (WER) measurements on the LibriSpeech test-clean, test-other and TEDLIUM-2 datasets with different inference chunk sizes using a unique model trained on Dynamic Chunk Training (DCT) interval of [160ms;1,280ms].The values in brackets correspond to the difference compared to the baseline and the values in square brackets represent the confidence interval of the result

Dataset	Model	Chunk size				
		160ms	320ms	640ms	1,280ms	Full-context
LibriSpeech test-clean	Baseline	7.55 [7.24;7.87]	4.82 [4.57;5.06]	3.90 [3.66;4.12]	3.49 [3.27;3.69]	2.90 [2.71;3.11]
	SENS-ASR	7.21**(-0.34)** [6.89;7.53]	4.73**(-0.09)** [4.48;4.99]	3.83**(-0.07)** [3.61;4.03]	3.44**(-0.05)** [3.24;3.65]	2.93(+0.03) [2.73;3.14]
LibriSpeech test-other	Baseline	18.34 [17.80;18.88]	12.41 [11.98;12.87]	9.70 [9.32;10.10]	8.39 [8.03;8.78]	6.76 [6.44;7.07]
	SENS-ASR	17.89**(-0.45)** [17.38;18.47]	12.11**(-0.30)** [11.69;12.60]	9.66**(-0.04)** [9.30;10.07]	8.55(+0.16) [8.21;8.94]	6.90(+0.14) [6.56;7.22]
TEDLIUM-2	Baseline	16.52 [15.83;17.14]	11.94 [11.39;12.52]	10.04 [9.50;10.55]	9.00 [8.52;9.51]	8.33 [7.88;8.84]
	SENS-ASR	15.60**(-0.92)** [14.98;16.23]	11.82**(-0.12)** [11.25;12.35]	9.79**(-0.25)** [9.28;10.30]	8.96**(-0.04)** [8.49;9.43]	8.33 [7.85;8.85]

was chosen empirically to ensure the convergence of the distillation loss without affecting the convergence speed of $\mathcal{L}_{RNN-T}$. In addition, an Adam optimizer is set with a learning rate of 0.0008 and a weight decay of 0.01.

Both baseline RNN-T and SENS-ASR systems are trained only once using Dynamic Chunk Training (DCT) (Zhang et al., 2020). During training, 60% of the batches use a chunk size randomly chosen from the interval [160ms;1,280ms] while other batches contain the full speech context.

The context module is trained with MPnet model (Song et al., 2020) which has previously been pretrained and fine-tuned on 1 billion pairs of sentences[3]. We perform a second stage of fine-tuning, as explained in section 2.2, using Mistral 7B Large Language Model to generate the paraphrases.

In the inference stage, we use **greedy search without an External Language Model rescoring** in all experiments to highlight the benefit of our contributions.

3.3. Results and discussions

Table 1 shows the performance of our SENS-ASR system compared to the baseline. Both systems are trained only once with DCT but used with a fixed chunk size at inference.The results indicate that SENS-ASR significantly reduces WER when inference is conducted with small chunk sizes (160ms and 320 ms). Then, insignificant improvements are observed with larger chunk sizes (640ms and 1,280ms).Finally, there is no improvement using the full-context audio. This trend can be attributed to the fact that larger chunks often contain sufficient acoustic information for a complete word(s) pronunciation. In particular, the WER for the SENS-ASR model is 7.21% for a chunk size of 160ms, which is an absolute reduction of 0.34% compared to the baseline, and 3.44% for 1,280ms, showing an absolute decrease of 0.05%. Also, for test-other, SENS-ASR reduces the WER by an absolute value of 0.45% compared to the baseline for 160ms while it gets a slight increase of 0.16% for 1,280ms. Similarly, for the TEDLIUM-2 dataset, SENS-ASR achieves a WER of 15.60% at a chunk size of 160ms, which is an absolute reduction of 0.92% compared to the baseline, and 8.96% for 1,280ms, indicating a decrease of 0.04%. These results demonstrate consistent improvements across datasets for small chunk sizes.

We compare in Table 2 our model with the State Of The Art models in Streaming-ASR, tested on LibriSpeech test-clean. Overall, the proposed model achieves competitive ASR performance and sometimes even better than models that use larger chunk size. It is difficult to draw a definitive conclusion on which model is better due to the differences on architectures used and the configurations that are discussed or not, in the papers. Yet, our SENS-ASR model, which has been trained only once with DCT is competitive on all reported chunk sizes compared with models trained specifically for this size.

To further push the comparison, we train two SENS-ASR models, each on a unique chunk size[4] , 320 ms and 640 ms, later used in inference and compare them to the model we train using Dynamic Chunk Training (DCT). While testing on Librispeech test-clean, the model improves in terms of WER when training on a targeted chunk size. The model trained on 160 ms chunk size has a WER of 4.58%,

[3]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[4]thus training 40% of batches with full context and 60% of unique chunk size

Table 2: Comparison between State Of The Art models and SENS-ASR in terms of WER on LibriSpeech test-clean. For SENS-ASR, the chunk sizes with (*) were not the only targeted sizes during training stage.

Model	Additional characteristics	Chunk size(ms)	WER (%)
trimtail (Song et al., 2023a)	-	640	4.68
ZeroPrompt (Song et al., 2023b)	-	640	4.41
CA transformer (Li et al., 2023)	Beam size=10	1,280	3.8
Delay penalized transducer (Kang et al., 2023)	$\lambda = 0.006$	640	3.74
Streamable decoder-only (Tsunoo et al., 2024)	Beam size=10 w LM	1,600	3.2
SENS-ASR(ours)	$\lambda = 0.006$	640*	3.83
	$\lambda = 0.006$	1,280*	3.44

i.e., an absolute WER reduction of **2.63%**. Also, the model trained on 320 ms chunk size has an absolute WER reduction of **0.55%** compared to the model trained with DCT.

Although the targeted chunk size training gives a big improvement in terms of transcription quality, this technique is **more computationally costly** (one training for each chunk size). Moreover, it **reduces the model robustness** while testing it with unseen chunk sizes during training.

3.4. Error analysis

To complete our analysis, we propose an error analysis. We compare the baseline and SENS-ASR with the 160 ms chunk size setup. We aim to better identify the potential benefits and drawbacks of the proposed method compared to a basic approach.

Table 3: Error comparison between SENS-ASR and the baseline using the chunk size of 160 ms by using WER and by each type of error. The values in brackets written in **bold** correspond to the relative difference compared to the baseline.

	Baseline	**SENS-ASR**
WER (%)	7.55	7.21
Number of Insertions	507	403 **(-20.51%)**
Number of Deletions	374	370 **(-1.07%)**
Number of Substitutions	3091	3020 **(-2.30%)**

We compare the errors of models by type of edition, as we show in the Table 3. We may consider that our approach has succeeded in reducing a significant number of insertions (compared to the baseline). This leads us to the idea that adding a semantic embedding helps reduce the tendency for overly verbose transcriptions generated by the baseline.

4. Conclusion

In this paper, we present SENS-ASR, a method that enhances streaming Automatic Speech Recognition (ASR) by integrating semantic information into *frame-embeddings*. Our approach addresses the limitations of traditional models, resulting in noticeable improvements in Word Error Rate (WER) on Librispeech and TEDLIUM-2. These positive results indicate that incorporating semantic context can effectively improve transcription quality in streaming scenarios while keeping great performances using full context audio due the Dynamic Chunk Training. Further research will aim to evaluate the proposed method with languages that have different linguistic structures and adapt the chunk size during inference depending on the linguistic and acoustic features of the input audio. Moreover, we aim to improve the training of the context module and the fine-tuning of the Sentence Embedding Model by using truncated text instead of the full audio transcription.

5. Acknowledgements

This work was also granted access to the HPC resources of IDRIS under the allocation 2025-A0191014876 made by GENCI.

6. Ethical considerations and limitations

We used Mistral-7B to generate paraphrases for fine-tuning the sentence-embedding teacher model. Because the pretraining corpora for Mistral-7B are not fully disclosed, we cannot exclude the possibility that public ASR corpora (e.g., LibriSpeech or TEDLIUM-2) overlap with its training data; this creates a theoretical risk of test-set contamination or memorized content being reintroduced via paraphrases. To mitigate this risk in our experiments, we (i) restrict the LLM use to an offline paraphrase-generation stage that is only used to fine-tune the teacher (the LLM and the teacher are never used at inference time) and (ii) limit the teacher's fine-tuning to training-set transcripts only.

7. Bibliographical References

Keyu An, Huahuan Zheng, Zhijian Ou, Hongyu Xiang, Ke Ding, and Guanglu Wan. 2022. Cuside: Chunking, simulating future context and decoding for streaming asr. In *Interspeech 2022*, pages 2103–2107.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93.

Herve A Bourlard and Nelson Morgan. 2012. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4960–4964. IEEE.

Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1815–1826, Santa Fe, New Mexico, USA.

Zhehuai Chen, He Huang, Andrei Andrusenko, Oleksii Hrinchuk, Krishna C Puvvada, Jason Li, Subhankar Ghosh, Jagadeesh Balam, and Boris Ginsburg. 2024. Salm: Speech-augmented language model with in-context learning for speech recognition and translation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13521–13525. IEEE.

Kwanghee Choi, Ankita Pasad, Tomohiko Nakamura, Satoru Fukayama, Karen Livescu, and Shinji Watanabe. 2024. Self-supervised speech representations are more phonetic than semantic. In *Interspeech 2024*, pages 4578–4582.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pages 5036–5040.

Haris Gulzar, Monikka Roslianna Busto, Takeharu Eda, Katsutoshi Itoyama, and Kazuhiro Nakadai. 2023. ministreamer: Enhancing small conformer with chunked-context masking for streaming asr applications on the edge. In *Interspeech*, pages 3277–3281.

Wei Kang, Zengwei Yao, Fangjun Kuang, Liyong Guo, Xiaoyu Yang, Long Lin, Piotr Żelasko, and Daniel Povey. 2023. Delay-penalized transducer for low-latency streaming asr. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Suyoun Kim, Yuan Shangguan, Jay Mahadeokar, Antoine Bruguier, Christian Fuegen, Michael L Seltzer, and Duc Le. 2021. Improved neural language model fusion for streaming recurrent neural network transducer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7333–7337. IEEE.

Mohan Li, Cong-Thanh Do, and Rama Doddipatla. 2023. Cumulative attention based streaming transformer asr with internal language model joint training and rescoring. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Pei Liu, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A survey of text data augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*, pages 191–195. IEEE.

Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2021. Dual causal/non-causal self-attention for streaming end-to-end speech recognition. In *Interspeech 2021*, pages 1822–1826.

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.

Vardan Papyan, XY Han, and David L Donoho. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663.

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.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. 2021. Speechbrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*.

Ramon Sanabria, Wei-Ning Hsu, Alexei Baevski, and Michael Auli. 2023. Measuring the impact of domain factors in self-supervised pre-training. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5. IEEE.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Xingchen Song, Di Wu, Zhiyong Wu, Binbin Zhang, Yuekai Zhang, Zhendong Peng, Wenpeng Li, Fuping Pan, and Changbao Zhu. 2023a. Trimtail: Low-latency streaming asr with simple but effective spectrogram-level length penalty. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xingchen Song, Di Wu, Binbin Zhang, Zhendong Peng, Bo Dang, Fuping Pan, and Zhiyong Wu. 2023b. Zeroprompt: Streaming acoustic encoders are zero-shot masked lms. In *Interspeech 2023*, pages 1648–1652.

Yuan Tseng, Titouan Parcollet, Rogier van Dalen, Shucong Zhang, and Sourav Bhattacharya. 2025. Evaluation of llms in speech is often flawed: Test set contamination in large language models for speech recognition. *arXiv preprint arXiv:2505.22251*.

Emiru Tsunoo, Hayato Futami, Yosuke Kashiwagi, Siddhant Arora, and Shinji Watanabe. 2024. Decoder-only architecture for streaming end-to-end speech recognition. In *Interspeech 2024*, pages 4463–4467.

Ehsan Variani, Ke Wu, Michael D Riley, David Rybach, Matt Shannon, and Cyril Allauzen. 2022. Global normalization for streaming speech recognition in a modular framework. *Advances in Neural Information Processing Systems*, 35:4257–4269.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Fangyuan Wang and Bo Xu. 2024. Sscformer: Push the limit of chunk-wise conformer for streaming asr using sequentially sampled chunks and chunked causal convolution. *IEEE Signal Processing Letters*.

Cheng Xu, Shuhao Guan, Derek Greene, and M Tahar Kechadi. 2024. Benchmark data contamination of large language models: A survey. *CoRR*.

Jiahui Yu, Chung-Cheng Chiu, Bo Li, Shuo-yiin Chang, Tara N Sainath, Yanzhang He, Arun Narayanan, Wei Han, Anmol Gulati, Yonghui Wu, et al. 2021a. Fastemit: Low-latency streaming asr with sequence-level emission regularization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6004–6008. IEEE.

Jiahui Yu, Wei Han, Anmol Gulati, Chung-Cheng Chiu, Bo Li, Tara N Sainath, Yonghui Wu, and Ruoming Pang. 2021b. Dual-mode asr: Unify and improve streaming asr with full-context modeling. In *International Conference on Learning Representations*.

Mohammad Zeineldeen, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2024. Chunked attention-based encoder-decoder model for streaming speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11331–11335. IEEE.

Binbin Zhang, Di Wu, Zhuoyuan Yao, Xiong Wang, Fan Yu, Chao Yang, Liyong Guo, Yaguang Hu, Lei Xie, and Xin Lei. 2020. Unified streaming and non-streaming two-pass end-to-end model for speech recognition. *arXiv preprint arXiv:2012.05481*.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wenbo Zhao, Ziwei Li, Chuan Yu, and Zhijian Ou. 2024. Cuside-t: Chunking, simulating future and decoding for transducer based streaming asr. In

2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP), pages 11–15. IEEE.

8. Language Resource References

Ardila, Rosana and Branson, Megan and Davis, Kelly and Kohler, Michael and Meyer, Josh and Henretty, Michael and Morais, Reuben and Saunders, Lindsay and Tyers, Francis and Weber, Gregor. 2020. *Common Voice: A Massively-Multilingual Speech Corpus*. European Language Resources Association.

Panayotov, Vassil and Chen, Guoguo and Povey, Daniel and Khudanpur, Sanjeev. 2015. *Librispeech: an asr corpus based on public domain audio books*. IEEE.

Zhou, Wei and Michel, Wilfried and Irie, Kazuki and Kitza, Markus and Schlüter, Ralf and Ney, Hermann. 2020. *The RWTH ASR system for TED-LIUM release 2: Improving hybrid HMM with specaugment*. IEEE.