NeMo@IWSLT 2026: Cascaded System for Simultaneous Speech Translation

Lilit Grigoryan[1,*], **Vladimir Bataev**[1,*], **Andrei Andrusenko**[1,*], **Oleksii Hrinchuk**[2], **Davit Karamyan**[1,3], **Enas Albasiri**[2], **Vitaly Lavrukhin**[2], **Nikolay Karpov**[2], **Boris Ginsburg**[2],
[1]NVIDIA, Armenia, [2]NVIDIA, USA, [3]ML Lab at YSU, Armenia
[*]Equal Contribution
Correspondence: lgrigoryan, vbataev, aandrusenko@nvidia.com

Abstract

This paper describes the NVIDIA NeMo team's submission to the IWSLT 2026 Simultaneous Speech Translation (SimulST) tracks. We use a cascaded architecture combining a dual-mode Unified ASR Transducer model with a multilingual Large Language Model (LLM). The ASR is trained to deliver stable transcriptions across wide range of latencies, providing a reliable foundation for high-quality LLM translation. Our submission participates in the English–German, English–Italian, and English–Chinese tasks, in both standard and contextualized settings, as well as the Czech–English standard track, covering both low- and high-latency scenarios. We further analyze how ASR and LLM design choices affect the system's overall latency and translation quality.

1 Introduction

This paper describes the NVIDIA NeMo team's submission to the IWSLT 2026 (Adelani et al., 2026) Simultaneous Speech Translation (SimulST) track. Our system utilizes a cascaded architecture, integrating specialized speech recognition (ASR) and machine translation (MT) components within a modular pipeline.

One of the advantages of this cascaded design is the decoupling of the recognition and translation tasks. This separation makes it possible to use a pretrained multilingual Large Language Model (LLM) for the translation stage. Specifically, our system utilizes pretrained Qwen-family models with demonstrated strong multilingual performance (Yang et al., 2025; Qwen Team, 2026).

However, the effectiveness of the downstream MT component in SimulST depends heavily on the upstream ASR system. High-quality simultaneous translation requires ASR models that produce accurate and stable hypotheses from partial audio. While standard offline models can be adapted for streaming through chunked or buffered decoding, these approaches often introduce significant degradation, especially in the low-latency regimes required by the competition.

In contrast, streaming-native architectures, such as cache-aware (Noroozi et al., 2024) or dual-mode Conformer systems (Gulati et al., 2020; Rekesh et al., 2023), are naturally suited for this setting, as they are designed to produce stable hypotheses from partial inputs with limited context. At the core of our submission is a Unified ASR Transducer (Andrusenko et al., 2026) model trained in dual-mode, which ensures reliable transcriptions during streaming decoding and maintains performance levels comparable to offline processing.

Our submission covers the English–German, English–Italian, English–Chinese, and Czech–English language pairs, participating in both the low- and high-latency tracks. For the English-source directions, we further explore the impact of contextualization by incorporating a boosting-tree-based word biasing method into the ASR component. We evaluate our system's performance on the latency–quality trade-offs defined by the IWSLT 2026 guidelines.

2 System Description

2.1 Methodology

We evaluate three streaming ASR strategies, all based on a FastConformer encoder paired with a Transducer-style decoder - either a standard Transducer (Graves, 2012) or a Token-and-Duration Transducer (TDT) (Xu et al., 2023).

We utilize the streaming inference engine available in the NVIDIA NeMo framework (Kuchaiev et al., 2019). It performs incremental decoding over incoming audio chunks, generating partial transcripts at each step. These partial hypotheses are then passed to an LLM to produce translations and may be revised as additional audio is

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 204–211
July 3-4, 2026 ©2026 Association for Computational Linguistics

	Latency	ASR Model	Settings	WER ↓	COMET ↑	BLEU ↑	LongYaaL(CU)↓
En→De	Low	nemotron-speech-streaming-en	[70, 13]	7.89	89.70	27.16	1611.61
		parakeet-tdt-v2	[70, 18, 18]	8.59	90.69	**29.37**	1831.36
		parakeet-unified-en	[70, 18, 18]	6.51	**91.36**	28.56	1853.54
	High	parakeet-tdt-v2	[70, 36, 36]	6.93	**91.96**	29.82	2946.72
		parakeet-unified-en	[70, 36, 36]	6.35	91.82	**30.31**	3185.07
En→It	Low	nemotron-speech-streaming-en	[70, 13]	7.89	83.24	36.37	1538.89
		parakeet-tdt-v2	[70, 18, 18]	8.59	85.22	38.82	1643.23
		parakeet-unified-en	[70, 18, 18]	6.51	**86.08**	**39.70**	1747.94
	High	parakeet-tdt-v2	[70, 36, 36]	6.93	**86.89**	**39.87**	2677.37
		parakeet-unified-en	[70, 36, 36]	6.35	86.61	39.57	2857.22
En→Zh	Low	nemotron-speech-streaming-en	[70, 13]	7.89	81.45	40.00	2200.42
		parakeet-tdt-v2	[70, 12, 12]	17.52	77.65	37.72	1923.97
		parakeet-unified-en	[70, 12, 12]	6.63	**82.15**	**42.04**	1879.00
	High	parakeet-tdt-v2	[70, 36, 36]	6.93	83.94	40.86	3108.38
		parakeet-unified-en	[70, 36, 36]	6.35	**84.22**	**42.21**	3476.45
Cs→En	Low	parakeet-tdt-v3	[70, 18, 18]	13.99	82.09	29.88	1506.67
		parakeet-unified-cs	[70, 18, 18]	11.33	**83.57**	**31.26**	1714.64
	High	parakeet-tdt-v3	[70, 36, 36]	13.98	83.11	31.51	3359.35
		parakeet-unified-cs	[70, 36, 36]	11.48	**84.12**	**32.10**	3143.27

Table 1: Translation results for four directions (En→De/It/Zh and Cs→En), under two latency regimes (low and high), and across three ASR architectures. The translation LLM is `Qwen/Qwen3-4B-Instruct-2507`. For cache-aware models, settings denote left and right context frames; for Parakeet and Unified Parakeet, they denote left, middle, and right context frames. Frame size is 80 ms. The EoU threshold is set to 800 ms for En→Zh and 1600 ms for all other directions.

processed. The outputs are finalized only after End-of-Utterance (EoU) detection, defined by a threshold of consecutive Transducer blank tokens, after which both transcripts and translations for the processed audio segment remain unchanged, and decoding proceeds with the next audio input.

The LLM employs a prompted prefix-matching strategy for consistency across updates. The model is provided with the current partial transcription, the prior translation prefix, and the history of finalized outputs as additional context. To maintain output stability, we utilize the Longest Common Prefix (LCP) strategy for prefix calculation following local agreement policy (Liu et al., 2020). Under this approach, the prefix for the current step is restricted to the portion of the translation that remains unchanged across successive updates, which prevents unstable ASR hypotheses from propagating errors. Following EoU detection, the transcription and translation contexts are updated and the prefixes are reset. Further technical details regarding the implementation are available in the NeMo framework[1].

[1] https://github.com/NVIDIA-NeMo/NeMo/blob/main/tutorials/asr/Streaming_ASR_Pipelines.ipynb

2.2 Standard Track: Speech-to-Text

ASR Model: The evaluated streaming ASR strategies differ in how they incorporate and process audio context.

Buffered Models: These models process audio using a sliding buffer. At each step, the input is divided into three segments: (1) left context providing past audio, (2) a central chunk, and (3) right context for look-ahead. For each new step, the entire buffer is reprocessed to produce transcripts for the central (fixed) and right (temporary transcription) segments, which are then passed to the translation LLM. The decoder state is passed between chunks, and the temporary part (corresponding to the right context) is discarded once it is produced as a chunk. This allows maintaining low latency (since the temporary transcription is produced immediately as frames become available) while preserving recognition quality (since right-context transcription is low-quality, it is replaced in the following step when it is produced with the available look-ahead context). The EoU detection is performed by counting frames on which only non-speech tokens are emitted, using both fixed and temporary transcription. Once EoU is detected, the transcription and the result-

LLM	En→De		En→It		En→Zh		Cs→En	
	COMET↑	BLEU↑	COMET↑	BLEU↑	COMET↑	BLEU↑	COMET↑	BLEU↑
`EuroLLM-9B-Instruct`	91.64	36.35	87.19	49.43	–	–	82.26	32.18
`Qwen3-4B-Instruct-2507`	91.36	28.56	86.08	39.70	**82.15**	42.04	83.57	31.26
`Qwen3-8B`	89.97	29.87	84.10	42.39	81.39	**45.57**	81.22	30.01
`Qwen3.5-4B`	90.12	33.67	84.51	40.56	79.17	38.75	81.66	29.25
`Qwen3.5-9B`	91.25	35.34	85.63	47.48	80.46	38.85	83.68	33.29
`Qwen3.5-27B`	92.66	**37.81**	87.72	**50.71**	81.97	42.14	**86.01**	**36.01**

Table 2: LLM translation quality across directions under the low-latency setting; see Table 1 for ASR decoding configurations.

ing translation hypothesis are considered fixed. This approach enables an offline ASR model such as `nvidia/parakeet-tdt-0.6b-v2` (NVIDIA, May 2025) or `nvidia/parakeet-tdt-0.6b-v3` (NVIDIA, Aug. 2025) to operate in a streaming setting. However, it has a key limitation: transcription quality degrades under low-latency constraints.

Cache-Aware Models: Cache-aware models (Noroozi et al., 2024) optimize computational efficiency by processing each audio frame exactly once. Instead of re-encoding a sliding window, these architectures maintain a bounded cache of self-attention and convolution layers, appending only new audio features to the existing state to eliminate redundant computation. Compared to the original Conformer-encoder (Gulati et al., 2020), the architecture is modified to prevent training-inference mismatch: Batch Normalization (Ioffe and Szegedy, 2015) layers are replaced with Layer Normalization (Ba et al., 2016), and all convolutions are made causal. We experiment with `nvidia/nemotron-speech-streaming-en-0.6b` (NVIDIA, Jan. 2026) cache-aware model. The model provides 70 frame context size, and limited number of latency regimes where the lookahead ranges from 0 to 13 frames, resulting in theoretical latency between 80 ms and 1.12 s.

Unified Transducer Models: Our primary system utilizes the Unified ASR Transducer `nvidia/parakeet-unified-en-0.6b` (NVIDIA, Apr. 2026), a single model trained to support both offline and streaming decoding modes. The model employs consistency regularization to ensure that streaming hypotheses remain stable and closely align with offline quality. The decoding process is similar to the buffered approach, where latency is regulated by adjusting the chunk size, and right context parameters. Due to its flexible design, the same model can be used for both low- and high-latency settings by selecting appropriate parameter configurations.

Translation LLM: We evaluate models from three families:

- EuroLLM (Martins et al., 2024): `EuroLLM-9B-Instruct`
- Qwen3: `Qwen3-4B-Instruct-2507`, `Qwen3-8B`
- Qwen3.5: `Qwen3.5-4B`, `Qwen3.5-9B`, `Qwen3.5-27B`

The system utilizes the user prompt shown below, which specifies the source and target languages. Following our local agreement strategy, the translation prefix (e.g., `"tgt_prefix":`) is **explicitly enforced** during decoding. This constraint ensures that the model's generation remains consistent with the partial output from the previous step.

User Prompt

```
Translate the following {src_lang} source text
to {tgt_lang}:
    {src_lang}: {src_text}
    {tgt_lang}: {tgt_prefix}
```

For Qwen models, we additionally apply a system prompt shown below, while EuroLLM models are used without a system prompt.

System Prompt

```
You are a professional machine translation
assistant.
Translate the input text into the target
language.
  - Output only the translation.
  - Do not complete or extend the text.
  - The input may be incomplete; preserve
    incompleteness.
  - Do not infer missing content.
  - Stop immediately after translating.
  - Preserve named entities, numbers,
    punctuation, and formatting.
```

Setup	Settings	COMET↑	BLEU↑	LongYaaL(CU)↓
GT text off. MT	–	95.11	33.99	–
Str. ASR off. MT	[70,8,8]	91.58	29.13	–
	[70,12,12]	91.94	29.19	–
	[70,18,18]	92.14	29.57	–
	[70,36,36]	92.57	29.51	–
Str. ASR str. MT	[70,8,8]	88.44	25.02	1142.40
	[70,12,12]	91.41	30.35	1397.37
	[70,18,18]	91.36	28.56	1854.54
	[70,36,36]	91.82	30.31	3185.07

Table 3: Translation quality under different ASR and translation setups. Translation LLM: `Qwen/Qwen3-4B-Instruct-2507`. Direction: En→De.

2.3 Speech-to-Text with Extra Context

To customize the cascade pipeline using the provided scientific papers, we used per-stream graph-based phrase boosting (Andrusenko et al., 2025; Bataev et al., 2025) with greedy decoding. We chose to tune hyperparameters on a pure ASR task (with both chunk and right-context lengths of 0.96 s) due to the simplicity of this approach and independence from the quality of the translation model.

The baseline pipeline (Team, Apr. 2026) extracted named entities only from the introductory part of each paper (title, authors, abstract), by prompting LLM 16 times for each prompt and keeping entities that appeared at least 8 times during this sampling process. This pipeline produced a short, low-coverage entity list, with a median of 25 entities per paper, dominated mostly by terms already well covered by the ASR system. This explains only a marginal improvement of 7.90% to 7.83% in WER after tuning the biasing weight on the dev set.

We updated the pipeline to cover full paper texts, parsing the abstract, numbered body sections, and reference list. We stripped out all URLs, emails, figure placeholders, and trailing appendices. The body text is grouped into chunks of $\sim$ 4000 words, split along section boundaries. Each chunk is sent to `Qwen3-30B-A4B-Instruct` in the same way as in the baseline pipeline. The body prompt explicitly enumerates 10 target categories (models, methods, datasets, metrics, etc.) and instructs against selecting common English words and general terms.

Since each chunk is much more informative than a single abstract, we reduced sampling to 8 times with a majority-vote threshold of 4. This keeps the extraction cost relatively low despite the larger input volume. After the extraction, we apply post-processing that merges chunk-level outputs, removes pure numbers, applies a length filter, and removes common stop-words. It also performs case-insensitive deduplication and preserves only the most frequent form of each entity.

The resulting phrase set is substantially longer and has higher coverage than the baseline (median length is 114 entities), which allowed us to increase the boosting weight from 0.3 to 0.4 (re-tuned on the dev set). These changes result in an improvement in ASR WER from 7.90% to 7.67%.

3 Experimental Setup

The English ASR models used in our experiments are publicly available on Hugging Face. For Czech, we train a Unified ASR Transducer model with 0.6B parameters on the MOSEL v2 Czech subset (Koluguri et al., 2025), comprising ∼15k hours of Czech speech. The encoder is initialized from `parakeet-tdt-0.6b-v3`. We refer to this model as `parakeet-unified-cs` in the result tables below.

Metrics: Following the competition requirements, we use OmniSTEval Toolkit (Polák et al., 2025) for evaluating both translation quality and latency (computation-aware and computation-unaware). Translation quality is primarily measured using COMET (Guerreiro et al., 2024), specifically by `Unbabel/XCOMET-XL` model. We additionally report BLEU scores. Latency is evaluated using the LongYAAL metric (Polák et al., 2026). Systems with computation-unaware LongYAAL less than 2 s are considered low-latency, while those with LongYAAL up to 4 s fall into the high-latency track.

4 Results

4.1 Ablations

ASR Model: Tab. 1 compares the three ASR models. For En→X, the Unified ASR Transducer consistently performs best in the low-latency setting. In the high-latency regime, the offline Parakeet models slightly outperform the Unified architecture (for En→It and En→De), likely due to the larger chunk size (2.88 s), which provides sufficient context for accurate transcription. However, when the chunk size is reduced to 1.44 s, the performance of the offline Parakeet models degrades noticeably. The cache-aware model, even with the

Direction	Our pipeline	Latency	System	COMET↑	BLEU↑	LongYaaL(CU)↓
En→De	`parakeet-unified-en` `Qwen3.5-27B`	Low	Baseline	74.80	22.35	1809.9
			Ours	92.74	**37.91**	1641.6
			Baseline + custom.	76.51	22.59	1747.2
			Ours + custom.	**92.84**	37.87	1648.7
		High	Baseline	83.51	27.44	3431.1
			Ours	**93.38**	37.41	2736.7
			Baseline + custom.	84.28	27.66	3353.5
			Ours + custom.	93.30	**37.88**	2729.8
En→It	`parakeet-unified-en` `Qwen3.5-27B`	Low	Baseline	68.25	30.63	1763.5
			Ours	**87.70**	50.62	1548.8
			Baseline + custom.	69.60	31.45	1734.8
			Ours + custom.	87.66	**51.01**	1556.4
		High	Baseline	78.06	37.28	3300.0
			Ours	88.38	50.58	2593.8
			Baseline + custom.	78.77	37.76	3231.1
			Ours + custom.	**88.67**	**51.14**	2549.7
En→Zh	`parakeet-unified-en` `Qwen3-4B-Instruct-2507`	Low	Baseline	74.96	40.85	1908.8
			Ours	82.15	42.04	1879.0
			Baseline + custom.	74.91	38.32	1875.8
			Ours + custom.	**82.43**	**42.12**	1891.1
	`parakeet-unified-en` `Qw3-4B-Instruct-2507/Qw3.5-27B`	High	Baseline	79.51	43.85	3479.1
			Ours	84.22	42.21	3476.4
			Baseline + custom.	80.04	43.84	3354.0
			Ours + custom.	**84.50**	**43.92**	3170.6
Cs→En	`parakeet-unified-cs` `Qwen3.5-27B`	Low	Baseline	81.45	31.84	1984.11
			Ours	**86.01**	**35.91**	1441.9
		High	Baseline	82.86	32.05	2547.22
			Ours	**87.48**	**38.68**	2787.7

Table 4: Comparison of our systems against the corresponding baselines. The pipeline column lists the ASR model on the first line and the translation LLM on the second line. Best BLEU and COMET values per direction and latency are highlighted in bold. Note that for high latency En→Zh our system uses `Qwen3-4B-Instruct-2507` for standard track and `Qwen3.5-27B` for customized.

largest right-context setting, remains within the low-latency constraints and achieves comparable, though slightly lower, COMET scores.

For En→Zh, we observe higher latency overall. This is caused by frequent reordering in the Chinese translations produced by the LLM, where the word order changes repeatedly as new context becomes available. This reduces the length of stable common prefixes, leading delayed stabilization of the output. To mitigate this, we use smaller chunk and right-context sizes (960 ms, or 12 frames) and reduce the EoU threshold to 800 ms, compared to 1600 ms for other directions. The Parakeet model shows substantial degradation relative to the Unified Parakeet model, driven by the use of smaller chunk sizes.

Translation LLM: The comparison of different multilingual LLM models is shown in the Tab. 2. `Qwen/Qwen3.5-27B` model outperforms other models for directions En→De, En→It and Cs→En. However, for the En→Zh direction `Qwen/Qwen3-4B-Instruct-2507` shows the highest COMET and BLEU scores.

Low-latency performance: Table 3 compares three pipelines: (1) ground-truth text with offline translation (upper bound), (2) streaming ASR with offline translation, and (3) streaming ASR with streaming translation.

Moving from (1) to (2) shows the effect of using streaming ASR, which leads to a small but consistent drop in quality. As the chunk size increases, this gap becomes smaller, suggesting that having more context helps recover most of the loss.

Comparing (2) and (3) shows the effect of

streaming translation. The differences are largest at the smallest chunk sizes, while for moderate chunk sizes the gap remains limited. Overall, the results suggest that even at extremely low latency, our system's performance is close to the oracle.

4.2 Final submission

Final results and comparison with the baselines are presented in Table 4. The official IWSLT2026 baseline is a cascaded system consisting of `Qwen/Qwen3-ASR-1.7B` (Shi et al., 2026) for speech recognition and `Qwen/Qwen3-4B-Instruct-2507` for translation. Official baseline supports En→De, En→It, and En→Zh translation directions.

For the Cs→En direction baseline, we use the Windowed Canary approach from the `simulstream` toolkit (Gaido et al., 2025). We run Canary-1b-v2 (NVIDIA, Aug 2025) in a streaming setting using this method. The method applies a fixed-length sliding-window retranslation strategy with deduplication to process unsegmented audio streams. It detects the longest common subsequence between consecutive windows to prevent repeated tokens caused by overlapping audio segments (Sen et al., 2022). We used grid search across parameters to determine system with highest COMET score for a given latency constraint.

Our final system components are listed in column 'Our Pipeline' of Table 4. For specific ASR decoding setups for each latency mode see 'Settings' Column in Table 1. Our systems consistently outperform the baselines across all directions and latency settings. We observe substantial gains in COMET and BLEU, particularly for En→De and En→It, while maintaining or reducing latency. Improvements for En→Zh are smaller but consistent. We also observe consistent gains in COMET from customization, indicating improved handling of named entities in translation.

5 Conclusion

We presented a cascaded speech translation system combining ASR and LLM-based translation, with a focus on low-latency settings. Across all evaluated directions, our approach consistently improves translation quality over the official IWSLT2026 baseline, yielding substantial gains in both COMET and BLEU while maintaining or reducing latency.

The results show that replacing the baseline ASR with a Unified Parakeet model provides strong improvements, and that LLM-based translation is robust across latency regimes. Additional gains are obtained from word-boosting based customization through improved handling of named entities.

Overall, the system demonstrates that a relatively simple cascaded setup, with careful tuning and targeted customization, can achieve strong performance in simultaneous speech translation without requiring complex architectural changes.

References

David Ifeoluwa Adelani, Victor Agostinelli, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Sebastien Bratières, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, Marcello Federico, Marco Gaido, Mahendra Gupta, HyoJung Han, Ali Hatami, David Javorský, Yejin Jeon, Marek Kasztelnik, Antoine Laurent, and 33 others. 2026. Speech translation and metrics in 2026: Findings of the iwslt campaign. In *Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026)*, San Diego, California, US. Association for Computational Linguistics.

Andrei Andrusenko, Vladimir Bataev, Lilit Grigoryan, Vitaly Lavrukhin, and Boris Ginsburg. 2025. Turbobias: Universal asr context-biasing powered by gpu-accelerated phrase-boosting tree. In *2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–7.

Andrei Andrusenko, Vladimir Bataev, Lilit Grigoryan, Nune Tadevosyan, Vitaly Lavrukhin, and Boris Ginsburg. 2026. Reducing the offline-streaming gap for unified asr transducer with consistency regularization. *Preprint*, arXiv:2604.19079.

J-L Ba, J-R. Kiros, and G-E. Hinton. 2016. Layer normalization. *arXiv:1607.06450*.

Vladimir Bataev, Andrei Andrusenko, Lilit Grigoryan, Aleksandr Laptev, Vitaly Lavrukhin, and Boris Ginsburg. 2025. NGPU-LM: GPU-Accelerated N-Gram Language Model for context-biasing in greedy ASR decoding. *Interspeech*.

Marco Gaido, Sara Papi, Mauro Cettolo, Matteo Negri, and Luisa Bentivogli. 2025. simulstream: Open-Source Toolkit for Evaluation and Demonstration of Streaming Speech-to-Text Translation Systems. *arXiv*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. In *ICML*.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xCOMET: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Anmol Gulati, James Qin, Chung-Cheng Chiu, and 1 others. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Proc. Interspeech 2020*, pages 5036–5040.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Nithin Rao Koluguri, Monica Sekoyan, George Zelenfroynd, Sasha Meister, Shuoyang Ding, Sofia Kostandian, He Huang, Nikolay Karpov, Jagadeesh Balam, Vitaly Lavrukhin, Yifan Peng, Sara Papi, Marco Gaido, Alessio Brutti, and Boris Ginsburg. 2025. Granary: Speech recognition and translation dataset in 25 european languages. *Interspeech*, abs/2505.13404.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, and 1 others. 2019. Nemo: a toolkit for building ai applications using neural modules. In *NeurIPS Workshop on Systems for ML*.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Interspeech 2020*, pages 3620–3624.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe. *Preprint*, arXiv:2409.16235.

Vahid Noroozi, Somshubra Majumdar, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2024. Stateful conformer with cache-based inference for streaming automatic speech recognition. *Preprint*, arXiv:2312.17279.

NVIDIA. Apr. 2026. Parakeet-unified-en-0.6b: Unified asr model for offline and streaming inference.

NVIDIA. Aug 2025. nvidia/canary-1b-v2.

NVIDIA. Aug. 2025. parakeet-tdt-0.6b-v3: Multilingual speech-to-text model.

NVIDIA. Jan. 2026. Nemotron-speech-streaming-en-0.6b: Nemotron asr streaming.

NVIDIA. May 2025. Parakeet tdt 0.6b v2 (en).

Peter Polák, Sara Papi, Luisa Bentivogli, and Ondřej Bojar. 2025. Better late than never: Evaluation of latency metrics for simultaneous speech-to-text translation. *arXiv preprint arXiv:2509.17349*.

Peter Polák, Sara Papi, Luisa Bentivogli, and Ondřej Bojar. 2026. Better late than never: Meta-evaluation of latency metrics for simultaneous speech-to-text translation. *Preprint*, arXiv:2509.17349.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, and Boris Ginsburg. 2023. Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8.

Sukanta Sen, Ondřej Bojar, and Barry Haddow. 2022. Simultaneous translation for unsegmented input: A sliding window approach. *Preprint*, arXiv:2210.09754.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*.

IWSLT Organization Team. Apr. 2026. Iwslt 2026 simultaneous translation baseline.

Hainan Xu, Fei Jia, Somshubra Majumdar, He Huang, Shinji Watanabe, and Boris Ginsburg. 2023. Efficient sequence transduction by jointly predicting tokens and durations. *Preprint*, arXiv:2304.06795.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.