Simultaneous Speech-to-Text Translation Web Application for Estonian

Bohdan Podziubanchuk and **Aivo Olev** and **Jiaming Kong** and **Tanel Alumäe**
Department of Software Science
Tallinn University of Technology
Estonia
{bohdan.podziubanchuk,aivo.olev,tanel.alumae}@taltech.ee

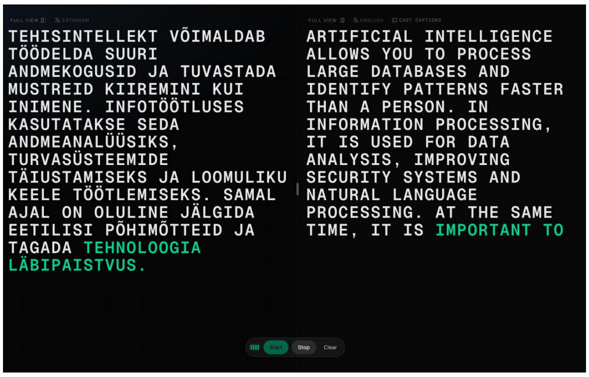
Abstract

This paper presents a new open-source web application for simultaneous speech-to-text translation. The system translates live Estonian speech into English, Russian, and Ukrainian text, and also supports English-to-Estonian translation. Our solution uses a cascaded architecture that combines streaming speech recognition with a recently proposed LLM-based simultaneous translation model. The LLM treats translation as a conversation, processing input in small five-word chunks. Our streaming speech recognition achieves a word error rate of 10.2% and a BLEU score of 26.1 for Estonian-to-English, significantly outperforming existing streaming solutions. The application is designed for real-world use, featuring a latency of only 3–6 seconds. The application is available at https://est2eng.cs.taltech.ee.

1 Introduction

Simultaneous speech-to-text translation (SimulST) systems produce real-time text-based translations from streaming speech. The latency of target language words is kept low enough for the listener to follow the speaker without major delay. This task is important in practical settings, for example when supporting talks at conferences with multilingual audience or generating live subtitles that must meet certain latency constraints.

Estonian is a Uralic language spoken by around one million native speakers. Estonia's growing ethnic and professional diversity, combined with the small size and complexity of the Estonian language, makes many newcomers reluctant to learn it. As a result, English is increasingly used in domains like higher education and technology, raising concerns about potential domain loss, where Estonian could gradually lose its functions in areas like higher education and technology.



TEHISINTELLEKT VÕIMALDAB
TÖÖDELDA SUURI
ANDMEKOGUSID JA TUVASTADA
MUSTREID KIIREMINI KUI
INIMENE. INFOTÖÖTLUSES
KASUTATAKSE SEDA
ANDMEANALÜÜSIKS,
TURVASÜSTEEMIDE
TÄIUSTAMISEKS JA LOOMULIKU
KEELE TÖÖTLEMISEKS. SAMAL
AJAL ON OLULINE JÄLGIDA
EETILISI PÕHIMÕTTEID JA
TAGADA TEHNOLOOGIA
LÄBIPAISTVUS.

ARTIFICIAL INTELLIGENCE
ALLOWS YOU TO PROCESS
LARGE DATABASES AND
IDENTIFY PATTERNS FASTER
THAN A PERSON. IN
INFORMATION PROCESSING,
IT IS USED FOR DATA
ANALYSIS, IMPROVING
SECURITY SYSTEMS AND
NATURAL LANGUAGE
PROCESSING. AT THE SAME
TIME, IT IS IMPORTANT TO

Figure 1: User interface with a split view (ASR and translation) optimized for low-latency incremental updates. Partial ASR and newly emitted translation words are visually marked to support stable live reading.

This paper describes a simultaneous speech-to-text translation system for Estonian, developed by Tallinn University of Technology and the University of Tartu within a project supported by the Estonian language technology programme. One of the goals of the project was to create a practical, easy-to-use, and accurate system that could be deployed in a variety of real-world settings, including conferences and seminars. The resulting system operates as a web application (see Figure 1) and currently supports translating Estonian speech into English, Ukrainian, and Russian text, as well as English speech into Estonian. The system follows a cascaded architecture: speech in the source language is first transcribed by a streaming automatic speech recognition (ASR) model and then translated by a machine translation (MT) model. Close-to-simultaneous translation is achieved by translating five-word chunks using a large language model (LLM) together with a multi-turn dialogue–based decoding strategy, in which source and target chunks appear interleaved in the translation history (Wang et al., 2025). The LLM is finetuned on parallel training data segmented into small, monotonic chunks according to word alignments, ensuring that no target word appears in an earlier chunk than its aligned source word.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 575–582
March 24-29, 2026 ©2026 Association for Computational Linguistics

The system freely available and open source, including the training scripts to generate supervision data for finetuning a LLM for multi-turn translation, and can thus be relatively easily adapted to other language pairs. The system comes with several addons that increase its usability: functionality to share live translation results to users' mobile devices and to stream translation to OBS Studio for overlaying over a video.

A demo video of the application is available at `https://youtu.be/F5bx3Wqyc4Q`.

2 Related work

According to our knowledge, the only publicly available simultaneous speech translation model that supports Estonian speech input and Estonian text output is the SeamlessStreaming model (Seamless Communication et al., 2023). SeamlessStreaming is an end-to-end simultaneous multilingual and multimodal translation framework built on the offline speech translation model SeamlessM4T-v2. It performs real-time speech-to-text and speech-to-speech translation for more than 100 input and nearly 100 output languages. Low-latency generation is achieved through Efficient Monotonic Multihead Attention (EMMA) and additional fine-tuning of the SeamlessM4T-v2 architecture for streaming inference. Its simultaneous text decoder follows a learned policy that decides whether to emit the next token or delay generation in order to read more input context.

Support for streaming Estonian speech translation has also recently been introduced by a few commercial providers, including Microsoft and Google. We were only able to test Microsoft's system.

3 Models

Our SimulST system is based on the cascaded approach, with independent streaming speech recognition and MT models.

3.1 Speech recognition

The Estonian streaming ASR system is based on the Zipformer neural transducer architecture (Yao et al., 2024) and was trained using the Icefall toolkit[1]. The model has about 150 million parameters and was trained on roughly 1334 hours of manually transcribed Estonian speech from the TalTech Estonian Speech Dataset 1.0[2] (Alumäe et al., 2023). In addition, training relied on around 4000 hours of automatically transcribed Estonian public TV (ETV) data, consisting of news and talk shows, and a 500-hour subset of the Gigaspeech dataset (Chen et al., 2021), which includes YouTube videos and podcasts. For automatically transcribing the Estonian data, we used Whisper *large-v3-turbo* (Radford et al., 2022), finetuned on the TalTech Estonian Speech Dataset 1.0. The ASR model produces properly capitalized and punctuated text. A subset of Gigaspeech was intermixed with Estonian data to improve the model's ability to transcribe English terms and expressions that are often embedded into Estonian sentences, especially in technological domains. Since the original transcripts of Gigaspeech are uppercase and not punctuated, we retranscribed the 500 hour subset using Whisper *large-v3-turbo*. The ETV audio remains the broadcaster's property; licensing details for the derived transcriptions are documented alongside the TalTech Estonian Speech Dataset 1.0.[3]

3.2 Machine translation

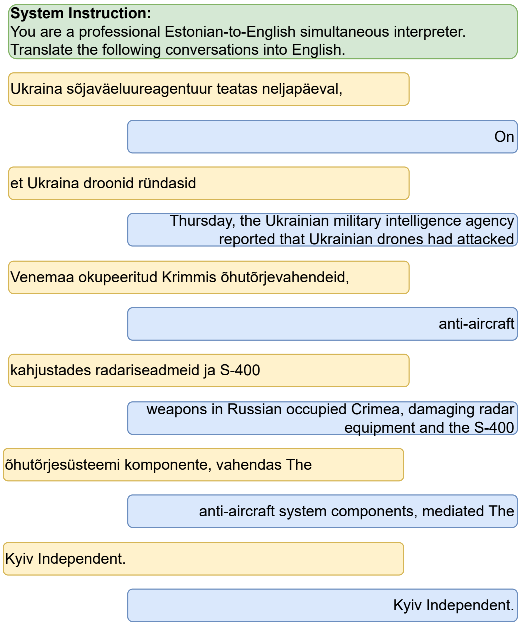
Our MT component is based on a recently proposed simultaneous MT approach that treats translation as a multi-turn dialogue between the source (as user turns) and the LLM (as assistant turns) (Wang et al., 2025), as illustrated in Figure 2. Instead of injecting new source tokens into the end of a growing prompt – a common workaround when adapting offline MT models for online use – each incoming source chunk is added as a new turn in the conversation. This setup allows the LLM to reuse its key–value cache efficiently, reducing both computational cost and latency. It also enables the use of existing, highly optimized LLM inference tools, since decoding follows the standard multi-turn dialogue pattern. Unlike translation models that integrate a policy network to decide whether to emit more tokens or wait for more input, the MT LLM used here simply finishes each chunk translation with an end-of-text token, after which control returns to the "user" to provide the next chunk.

In order to train LLM to perform such partial translations, the LLM has to be finetuned using specialized supervised fine-tuning data that mim-

[1] `https://github.com/k2-fsa/icefall`

[2] `https://cs.taltech.ee/staff/tanel.alumae/data/est-pub-asr-data/`

[3] `https://cs.taltech.ee/staff/tanel.alumae/data/est-pub-asr-data/`



System Instruction:
You are a professional Estonian-to-English simultaneous interpreter.
Translate the following conversations into English.

Ukraina sõjaväeluureagentuur teatas neljapäeval,

On

et Ukraina droonid ründasid

Thursday, the Ukrainian military intelligence agency reported that Ukrainian drones had attacked

Venemaa okupeeritud Krimmis õhutõrjevahendeid,

anti-aircraft

kahjustades radariseadmeid ja S-400

weapons in Russian occupied Crimea, damaging radar equipment and the S-400

õhutõrjesüsteemi komponente, vahendas The

anti-aircraft system components, mediated The

Kyiv Independent.

Kyiv Independent.

Figure 2: The translation LLM produces a translation for each fixed-word-length input chunk, taking into account the already translated chunks. The LLM sometimes correctly avoids producing the translation too early, if it not sure about the following context.

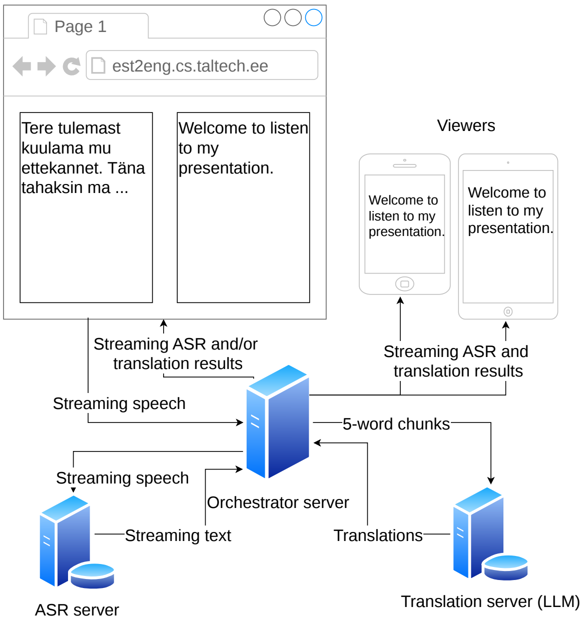

Page 1
est2eng.cs.taltech.ee
Tere tulemast kuulama mu ettekannet. Täna tahaksin ma ...
Welcome to listen to my presentation.
Viewers
Welcome to listen to my presentation.
Welcome to listen to my presentation.
Streaming ASR and/or translation results
Streaming ASR and translation results
Streaming speech
5-word chunks
Streaming speech
Orchestrator server
Streaming text
Translations
ASR server
Translation server (LLM)

Figure 3: Architecture of the application.

ics conversational chunked translation. Such data is generated by segmenting parallel sentences using word alignments and converting them into sequences of source and target texts, ensuring that a target word does not occur in a response earlier than the corresponding source word. To make the model robust to different latency settings, the segmented trajectories are further augmented with operations that merge or shift chunks. After training, the LLM can translate incoming partial source text chunk-by-chunk while maintaining coherence using previous conversational turns as context. Experiments by Wang et al. (2025) showed that conversational prompting approaches offline LLM-based translation in quality while substantially reducing latency and is a good alternative to specialized simultaneous MT systems in efficiency.

Our multi-turn simultaneous MT model was fine-tuned from the existing LLM-based offline MT model Hunyuan-7B-MT[4] (Zheng et al., 2025). Experiments showed that using Hunyuan-7B-MT as the base model yields better results than using more general LLMs, such as Llama 3.1 of similar size.

As training data, we sampled 500K sentence pairs for all supported translation directions (Estonian ↔ English, Estonian ↔ Ukrainian, and Estonian ↔ Russian) from the SynEst corpus (Korotkova et al., 2024). SynEst contains synthetic translations of texts from the Estonian National Corpus (Koppel and Kallas, 2022) into 11 languages, as well as translations from these 11 languages into Estonian, drawing on various web-based sources such as NewsCrawl[5] .

The sampled parallel data was processed as follows[6]. First, we generated word-alignment information using the pretrained multilingual encoder model XLM-R (Conneau et al., 2020). These alignments were then converted into dependency graphs indicating, for each target word, the minimal relevant source-side position. Finally, we transformed the dependency graphs into read–write trajectories, represented as multi-turn chat messages.

We trained Hunyuan-7B-MT using full finetuning for one epoch over the generated dataset. Training took 61 hours on an HPC node equipped with eight AMD MI250X GPUs.

4 Architecture

The system is deployed as a web application with a client-server architecture, as shown in Figure 3. The frontend runs in the user's browser and captures audio from the microphone, while the backend per-

[4] https://huggingface.co/tencent/Hunyuan-MT-7B

[5] https://data.statmt.org/news-crawl/

[6] Source code: https://github.com/jiamingkong/LLM_based_simulMT

forms ASR inference and streams transcriptions back to the client in real time.

4.1 System Overview

The frontend is built using Next.js and React, communicating with the backend via a secure WebSocket connection (WSS). Audio is captured using the Web Audio API with an AudioWorklet processor, resampled to 16 kHz, and transmitted as base64-encoded 16-bit PCM chunks at 100 ms intervals. The backend is implemented in Rust using the Axum web framework with Tokio for asynchronous I/O, and is deployed behind an nginx reverse proxy that handles TLS termination and WebSocket upgrades.

4.2 ASR Backend

The ASR backend[7] uses sherpa-onnx[8], a lightweight inference engine for speech recognition models in ONNX format. We interfaced with sherpa-onnx through Rust FFI bindings provided by the sherpa-rs library. The server maintains a pool of recognizer instances to handle concurrent sessions, with each session mapped to a recognizer using hash-based distribution.

For NeMo-based models such as FastConformer, we exported the models to ONNX format using NeMo's built-in export functionality. The export process involved configuring the decoder type (CTC or transducer), extracting streaming parameters such as chunk size and cache dimensions, and optionally applying INT8 quantization.

Endpointing in the live pipeline is client-driven: the client signals utterance end via an `utterance_end/stop` event, which triggers `finalize_session` on the server. The current client does not run VAD; silence-based endpointing can be enabled server-side or added client-side in future work.

The system supports multiple translation directions, each requiring a different ASR model. For Estonian input, we use the custom Zipformer model described in Section 3.1, which operates in true streaming mode using neural transducer architecture and produces properly punctuated and capitalized text. For English input (English→Estonian translation), we use NVIDIA's FastConformer Hybrid model[9], exported as an RNNT transducer. We use NeMo models for English because they provide ready-to-export streaming models with ONNX support, while comparable open streaming models for Estonian were not available. Whisper is used for Estonian only as an offline teacher and baseline; it does not provide a streaming ONNX model suitable for our deployment.

4.3 Translation server

Since the translation model is a standard finetuned LLM, it is hosted on a standard GPU-based inference endpoint that uses llama.cpp [10] for model serving. Translation requests are done for every five source words, using a history of 10 previous inputs and outputs. Once a chunk is sent to the LLM, its translation is treated as final and is not revised if upstream ASR hypotheses later change.

4.4 Web Application

The web application[11] captures audio with echo cancellation, noise suppression, and automatic gain control enabled. The application buffers five source words before sending them to the MT model to balance latency against translation quality. Caption sharing across devices is implemented via Firebase Firestore, while OBS Studio integration uses the obs-websocket protocol.

5 User interface

The system targets three practical roles: (i) a moderator/speaker who runs the application during a talk, (ii) a technical operator who configures caption casting for OBS Studio, and (iii) audience members who read live captions on mobile devices. The UI is optimized for readability and stable incremental updates under low-latency constraints.

5.1 Main interface for moderators and speakers

The main application page provides simple session controls (Start, Stop, Clear) and language-direction selection. On desktop, the interface uses a two-panel layout with a resizable split (ASR on the left, translation on the right). On mobile, it collapses to a single panel with a left/right toggle.

Operational state is exposed via lightweight cues. A floating ServerStatus widget reports backend availability (e.g., ready/waking up/unreach-

[7] Source: https://github.com/aivo0/rust-asr-server

[8] https://github.com/k2-fsa/sherpa-onnx

[9] https://huggingface.co/nvidia/stt_en_fastconformer_hybrid_medium_streaming_80ms_pc

[10] https://github.com/ggml-org/llama.cpp

[11] Source: https://github.com/Danbog32/Estonian-to-English-translation

able). A green "live" indicator shows when microphone/streaming is active. Connection or microphone failures are surfaced as toast notifications, making disruptions immediately visible.

To make recording status immediately recognizable without adding visual noise, we added a minimalistic audio level indicator matching the app's dark/emerald theme. It consists of four vertical animated bars, is responsive on both desktop and mobile, and is shown only while recording, providing an at-a-glance confirmation that audio capture is active.

5.2 Incremental display

To reduce distraction from streaming revisions, the ASR panel combines finalized text with the current partial hypothesis and highlights the most recent words, while earlier lines remain visually stable. The translation panel renders words incrementally, briefly highlighting newly emitted words and then fading the highlight to avoid flicker. Autoscroll follows the stream only while the user stays at the bottom; if the user scrolls up, a "Scroll to bottom" control appears.

Translation is performed in small source-text windows (five-word chunks), providing near-simultaneous output while maintaining local context. The UI does not expose latency–quality controls, keeping the interaction surface minimal.

5.3 Sharing and OBS overlay

For audience-scale use, the moderator can enable caption sharing via a "Cast captions" modal. When enabled, the system generates a unique session identifier and provides a shareable link together with a QR code and one-click copy. Viewers open a dedicated reader page that renders incoming captions.This separation of roles keeps the live session controllable: the host can start/stop broadcasting, while viewers remain read-only.

The same modal supports OBS Studio integration for live streams. During operation, captions are published in a broadcast-friendly format by selecting only the most recent words, wrapping them into two lines, and rate-limiting updates to avoid excessive refreshes. This produces compact, stable overlays suitable for conference recordings and live streams.

6 Evaluation

In order to assess the performance of the system, we evaluated both ASR and SimulST quality on dedicated test sets containing real-world broadcast and conversation data.

	Estonian	English
Non-streaming (Whisper)	10.0	16.3
Streaming (Icefall/Nvidia)	10.2	25.2

Table 1: WER results for Estonian and English.

6.1 Data

The evaluation data consists of long broadcast news and conversational recordings with different levels of spontaneity, including press conferences, TV talk shows, YouTube videos, and news programmes featuring many interviews. The total duration of the evaluation dataset is 4 hours for Estonian and 3 hours for English. This material has previously been used for offline speech-translation evaluation (Sildam et al., 2024) and is publicly available[12].

Reference translations for the evaluation set were created by professional translators in Estonia, using both the audio transcriptions and the audio recordings themselves as inputs.

6.2 Speech recognition

Table 1 compares word error rates (WERs) of offline and streaming models on Estonian and English test data. For English, we used Whisper *large-v3-turbo* with voice activity detection, the *–hallucination_silence_threshold* parameter set to 2.0, and word-level timestamps enabled, as these settings help reduce hallucinations. For Estonian, we used a version of *large-v3-turbo*[13] finetuned on 1334 hours of Estonian speech with verbatim transcripts.

WERs were computed from Whisper's long-form decoding output. Because this decoding strategy does not align hypotheses with reference sentences, we computed WERs after removing punctuation, lowercasing both hypotheses and references, and aligning words using minimum WER segmentation (mwerSegmenter) (Matusov et al., 2005) through the SLTev toolkit (Ansari et al., 2021).

Results show that, for Estonian, the streaming ASR quality is very close to offline finetuned Whisper performance. This is expected, as the streaming model is also trained on synthetic transcripts

[12] https://github.com/alumae/k6net6lke-benchmark

[13] https://huggingface.co/TalTechNLP/whisper-large-v3-turbo-et-verbatim

produced by Whisper and therefore learns to imitate Whisper's output. Whisper was also finetuned mostly on the same high quality data as the streaming model. For English, the comparison is between different models: the offline baseline is Whisper *large-v3-turbo*, while the streaming model is NVIDIA's FastConformer. The gap is therefore expected. The streaming setup is also intentionally minimal (greedy search without language-model rescoring), and endpointing is disabled during streaming, so long segments are decoded continuously, which can increase drift and substitution errors over time.

6.3 Speech translation

The streaming translation pipeline uses word-level windowing rather than traditional utterance-based translation. The frontend buffers transcribed words and triggers translation when six words have accumulated since the last emission, extracting five-word chunks for translation. Each chunk is sent to the LLM together with the previous ten source–target pairs as conversational context, enabling consistent terminology and better handling of anaphora across segments. This architecture achieves an end-to-end latency of approximately 3–6 seconds, competitive with professional human interpreters who typically maintain 3–10 second delays. In preliminary ablations, 5–6 word chunks yielded nearly identical translation quality, 4-word chunks degraded quality, and chunk sizes above 6 increased latency beyond acceptable live use; we therefore fixed chunk size to 5. We selected a 10-turn history window to bound latency while preserving local discourse context; larger windows increased response time without clear qualitative gains in pilot use.

SimulST evaluation was based on two metrics: BLEU (Papineni et al., 2002) and BLEURT (Sellam et al., 2020). BLEURT is a learned metric trained on human evaluation scores of translation references and corresponding MT outputs. We used the multilingual BLEURT-20D12 model (Pu et al., 2021). BLEU and BLEURT scores were computed after aligning words in the candidate translations with the references using mwerSegmenter via the SLTev toolkit.

As baselines, we used three systems. The offline cascaded system combines Whisper *large-v3-turbo* for ASR (using the finetuned version for Estonian) with the *Neutotõlge* MT system (Tättar et al., 2022) developed by the NLP research group at the University of Tartu. We accessed the system via its API, although the corresponding models are also publicly available. We also tested Microsoft Azure's streaming speech translation API. It should be noted that Azure provides only pseudo-streaming output: although translated words are emitted with low latency as response to streaming speech input, they remain unstable until the end of an utterance-like segment is detected, and both the wording and word order often change when new words are emitted, especially when the current segment is finalized. As the third baseline, we used the Seamless Streaming (Seamless Communication et al., 2023) models to generate translations.

Table 2 presents the Estonian–English, Estonian–Russian, and English–Estonian translation results in terms of BLEU and BLEURT. Among the baseline systems, Azure's pseudo-streaming translation performs on par with our best offline cascaded setup, while Seamless Streaming shows clearly lower quality.

We evaluated two MT models within our proposed streaming translation architecture: Hunyuan-7B-MT and Llama-3.1-8B-EstLLM[14] (a finetuned Llama-3.1 model with continued pretraining and instruction tuning on mostly Estonian data). The results show that finetuning a dedicated translation-oriented LLM gives better translation quality than using a language-specific but task-agnostic LLM. Our best cascaded system outperforms Seamless Streaming by a large margin and is approximately 5 BLEU points lower than the best open cascaded offline system in all translation directions.

Evaluation results suggest that translations from Estonian achieve higher quality than translations into Estonian. This is mainly due to the relatively weak English ASR model used in our current setup. For our main translation direction – Estonian to English – the quality is already sufficient for use at real live events.

6.4 Limitations

We did not include traditional text-only SimulMT baselines such as fixed wait-k policies. A fair comparison would require an end-to-end setup that maps streaming ASR timestamps to a text-level SimulMT policy and evaluates both latency and stability under identical real-time constraints. Implementing and validating such a baseline is left for future work.

[14] https://huggingface.co/tartuNLP/llama-estllm-prototype-0825

	$et \rightarrow en$		$et \rightarrow ru$		$en \rightarrow et$	
	BLEU	BLEURT	BLEU	BLEURT	BLEU	BLEURT
Baselines						
Offline cascaded (Whisper + Neurotõlge)	31.9	0.60	26.6	0.61	16.7	0.47
Azure (pseudo-streaming)	31.4	0.56	17.4	0.53	20.4	0.53
Seamless Streaming	13.8	0.36	9.5	0.29	8.8	0.27
Our cascaded streaming systems						
ASR + Llama-3.1-8B-EstLLM ft.	24.8	0.54	21.3	0.54	9.8	0.34
ASR + Hunyuan-7B-MT ft.	26.1	0.56	22.3	0.56	11.1	0.37

Table 2: Simultaneous translation evaluation results of baseline models and our cascaded systems. The system corresponding to the last row is used in the live system.

7 Conclusion

This work demonstrates that high-quality, low-latency simultaneous speech-to-text translation for Estonian is now technically feasible using an open, deployable, and reproducible system. By combining a strong streaming ASR model with a conversationally prompted, chunk-based LLM translator, we show that a cascaded architecture can approach offline translation quality while keeping latency within 3–6 seconds, suitable for real-world conference and seminar scenarios. Our work provides an example for building similar systems for other languages, including methods for generating multi-turn supervision data and an openly available web application ready for real-world use.

The most urgent future work is replacing the current English ASR model with a stronger one, thereby improving the English-to-Estonian translation direction.

Acknowledgments

This research was supported by the Estonian Centre of Excellence in AI (EXAI), National Program for Estonian Language Technology Program (projects EKTB77 and EKTB117), both funded by the Estonian Ministry of Education and Research, and by the Estonian Language Data Research Infrastructure (KeTA).

References

Tanel Alumäe, Joonas Kalda, Külliki Bode, and Martin Kaitsa. 2023. Automatic closed captioning for Estonian live broadcasts. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 492–499, Tórshavn, Faroe Islands. University of Tartu Library.

Ebrahim Ansari, Ondřej Bojar, Barry Haddow, and Mohammad Mahmoudi. 2021. SLTEV: Comprehensive evaluation of spoken language translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 71–79, Online. Association for Computational Linguistics.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. In *Interspeech 2021*. ISCA.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Kristina Koppel and Jelena Kallas. 2022. Eesti keele ühendkorpuste sari 2013–2021: mahukaim eestikeelsete digitekstide kogu. *Eesti Rakenduslingvistika Ühingu aastaraamat = Estonian Papers in Applied Linguistics*, 18:207–228.

Elizaveta Korotkova, Taido Purason, Agnes Luhtaru, and Mark Fishel. 2024. Multilinguality or back-translation? A case study with Estonian. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11838–11848, Torino, Italia. ELRA and ICCL.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of the Second International Workshop on*

Spoken Language Translation, Pittsburgh, Pennsylvania, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *Preprint*, arXiv:2212.04356.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, and 46 others. 2023. Seamless: Multilingual expressive and streaming speech translation. *Preprint*, arXiv:2312.05187.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Tiia Sildam, Andra Velve, and Tanel Alumäe. 2024. Finetuning end-to-end models for Estonian conversational spoken language translation. In *Proceedings of the Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)*, pages 166–174, Bangkok, Thailand. Association for Computational Linguistics.

Andre Tättar, Taido Purason, Hele-Andra Kuulmets, Agnes Luhtaru, Liisa Rätsep, Maali Tars, Marcis Pinnis, Toms Bergmanis, and Mark Fishel. 2022. Open and competitive multilingual neural machine translation in production. *Proceedings of Baltic HLT*.

Minghan Wang, Thuy-Trang Vu, Yuxia Wang, Ehsan Shareghi, and Gholamreza Haffari. 2025. Conversational SimulMT: Efficient simultaneous translation with large language models. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 93–105, Vienna, Austria (in-person and online). Association for Computational Linguistics.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. *Preprint*, arXiv:2310.11230.

Mao Zheng, Zheng Li, Bingxin Qu, Mingyang Song, Yang Du, Mingrui Sun, and Di Wang. 2025. Hunyuan-MT technical report. *Preprint*, arXiv:2509.05209.