UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Henrik Lepson

Estonian Simultaneous Speech-to-Text Machine Translation

Master's Thesis (30 ECTS)

Supervisor:
Mark Fišel, PhD

Tartu 2025

Estonian Simultaneous Speech-to-Text Machine Translation

Abstract:

Simultaneous machine translation is a task, where the translation system is expected to start translating before having access to the entire input sequence. This makes it a challenging and error-prone task. This thesis explored the feasibility of using pre-trained open models for simultaneous speech-to-text translation on the Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian directions. Two types of systems were evaluated: cascaded and end-to-end. The cascaded system relied on Whisper large-v3-turbo and NLLB-200 distilled 1.3B. The end-to-end system was based on Seamless M4Tv2 large. In addition, both systems used Voice Activity Detection (VAD) and LocalAgreement. The systems were compared with and without fine-tuning. For fine-tuning, a synthetic dataset with more than 4 million samples was created from various publicly available datasets. The dataset contained a 1:1 mix of full and partial sequences. The evaluation results showed that both systems are strongest on the Estonian-English direction followed by English-Estonian. Estonian-English direction can be translated without additional fine-tuning. Both systems struggled on the Estonian-Russian and Russian-Estonian directions. The translation quality and latency improved for both directions after fine-tuning.

Keywords: simultaneous machine translation, VAD, Whisper, NLLB-200, Seamless M4T

CERCS: P176, Artifical Intelligence

Kõnesignaali sünkroonne masintõlge eesti-inglise ja eesti-vene keelepaaride vahel

Lühikokkuvõte:

Sünkroonne masintõlge on valdkond, kus tõlkesüsteem peab hakkama tõlkima osalise sisendi põhjal, millest tulenevalt on tegu keerulise ja veaohtliku ülesandega, eriti kui varasema väljundi parandamine ei ole lubatud. Selles töös uuriti levinud vabade kaaludega masinõppe mudelite võimekust tõlkida sünkroonselt eesti-inglise, eesti-vene, inglise-eesti ja vene-eesti suundade vahel. Tõlkimiseks loodi kaks erinevat süsteemi. Esimene oli kaskaadsüsteem, mis kasutas Whisper large-v3-turbo mudelit transkriptsioonide loomiseks ja NLLB-200 distilled 1.3B mudelit tõlkimiseks. Teine süsteem põhines Seamless M4Tv2 large mudelil. Lisaks kasutasid mõlemad süsteemid VAD-i ja LocalAgreement strateegiat. Süsteeme võrreldi ilma ja koos peenhäälestamisega. Mudelite peenhäälestamiseks loodi avalike andmestike põhjal enam kui 4 miljonist näitest koosnev sünteetiline andmestik, mis sisaldas võrdsetes osades täielikke ja osalisi lauseid. Mõlemad süsteemid olid ilma peenhäälestamiseta tugevad eesti-inglise suunal, kuid ka inglise-eesti tulemused olid rahuldavad. Eesti-vene ja vene-eesti suundadel tõlkimiseks oli peenhäälestamine vajalik, mille tulemusena paranes tõlke kvaliteet ning vähenes sisendi ja väljundi vaheline viivitus.

Võtmesõnad: sünkroonne masintõlge, VAD, Whisper, NLLB-200, Seamless M4T

CERCS: P176, Tehisintellekt

Contents

1. Introduction

Simultaneous machine translation (SiMT) is a subset of machine translation which starts to translate the input sequence into the output sequence without waiting for the input sequence to end. It is a difficult and error-prone process because the translation system does not initially have access to the full context and changing the previously generated output may not be allowed, thus creating a situation where it might be difficult for the model to produce correct output going forward due to the mistakes that it has already made. According to M. Ma et al. (2019), another challenging aspect is the differences in word order between languages, such as English and German.

In terms of designing simultaneous translation systems, the main question is the balance between quality and latency, which is not an issue for non-simultaneous translation systems (Cho and Esipova, 2016). It is obvious, that the best translation quality can be achieved by waiting for the input sequence to end, but at this point the task turns into regular machine translation task. However, starting to translate prematurely will degrade translation quality. There is no right answer to how quality and latency should be balanced because it depends on the use case.

Although many translation systems, regular and simultaneous, work exclusively with text input and output, there also exist speech-to-text, text-to-speech and speech-to-speech translation tasks. These tasks are more difficult to solve because there is less available speech data compared to text data. This is problematic for low-resource languages, which do not have enough text data to start with. Although Estonian can not be considered a low resource language in the context of speech-to-text translation because it is included in well-known benchmark datasets, such as CoVoST2 (Wang, Wu, and Pino, 2020) and FLEURS (Conneau et al., 2022), it is still a relatively low resource language compared to English, Spanish, Russian, German, and other widely spoken languages.

This thesis explores the possibility of using open translation models for Estonian simultaneous speech-to-text translation. Although, many well-known open translation models are pre-trained on the Estonian language, such as NLLB-200 (N. Team et al., 2022), MADLAD-400 (Kudugunta et al., 2023) and Seamless M4T (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023), the Estonian translation quality is poorer compared to translation directions between high resource languages. In addition, these models are usually not trained for simultaneous translation, thus the initial translation performance becomes even more important because the model has access to limited context. This work focuses on Estonian-English, Estonian-Russian, English-

Estonian and Russian-Estonian directions. English-Russian and Russian-English directions are included in the evaluations in order to provide context for the translation quality between high-resource languages.

Two types of speech-to-text translation systems are evaluated, cascaded system and end-to-end system. Cascaded systems use a speech recognition model to create transcriptions, which are then translated using a text-to-text translation model. End-to-end systems do not use a separate speech recognition model and work directly with the input sequence. The performance of these systems is evaluated with and without fine-tuning. There exist approaches that try to adapt existing non-simultaneously trained models for the simultaneous translation task, so it is possible that fine-tuning may not be necessary for all translation directions. After that, the translation models are fine-tuned and evaluated again.

The models are fine-tuned on the Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian directions. There does not exist a large enough parallel dataset with required translation directions, so the dataset is created by combining publicly available Estonian, English and Russian speech datasets. These datasets are originally speech datasets with transcriptions, so translations were generated using machine translation. The translation of the transcriptions is not part of this thesis, but has been previously done by the TalTech Laboratory of Language Technology and TartuNLP group. The evaluation of the systems is done on the FLEURS dataset (Conneau et al., 2022).

The main research questions of this thesis are:

- Can existing open models be used for simultaneous speech-to-text translation between Estonian, English and Russian languages without any fine-tuning?
- Is it possible to significantly improve translation quality by fine-tuning the pre-trained models on a synthetic parallel Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian dataset?

All data processing, training runs and evaluations are conducted on the High Performance Computing Center of the University of Tartu (UT HPC), also known as Rocket (University of Tartu, 2018). The available GPUs are Nvidia V100 32GB, A100 40GB and A100 80GB.

The thesis has 6 chapters. Chapter 2 gives a short overview of the machine translation task with a focus on speech-to-text translation and simultaneous translation. Although simultaneous speech-to-text translation is in principle a sequence-to-sequence task similar to the regular

text-to-text machine translation, there are some differences in regards to input processing and generating partial sequences. The criteria for selecting models, simultaneous translation specific adaptions, fine-tuning process and evaluation setup are described in Chapter 3. Chapter 4 describes the process of creating a combined Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian speech-to-text parallel dataset with 4.5 million samples. The results are presented in Chapter 5 and further analysed in Chapter 6.

2. Background

This chapter gives a brief overview of the machine translation task before proceeding to the details of speech-to-text and simultaneous speech-to-text translation. In addition, some of the most commonly used translation quality and latency metrics are described. Latency metrics are used to evaluate simultaneous translation systems.

2.1 Machine Translation

Machine translation is a sequence-to-sequence task that transforms a sequence in one language to a sequence in another language while trying to keep the meaning of the initial sequence. Translated sequences are usually in text or audio format. Depending on the input and output sequences, the machine translation has four variants: text-to-text translation, speech-to-text translation, text-to-speech translation and speech-to-speech translation. Interestingly, it is possible to think of automatic speech recognition (ASR) as a case of speech-to-text translation, where the source and target languages are the same.

The best machine translation approaches are based on neural networks, which in 2014 started to surpass traditional phrase-based systems that used statistical machine learning (Bahdanau, Cho, and Bengio, 2016). Initially, recurrent neural networks (RNN) were used that encoded the entire input sequence into a single fixed-length vector before using it to generate an output, but this limitation was eliminated by the attention mechanism that allowed the decoder to focus only on the relevant parts of the input sequence at each decoding step (Bahdanau, Cho, and Bengio, 2016). Currently, RNNs are not widely used anymore, and the leading translation models use the Transformer architecture instead.

The Transformer architecture was proposed in the paper ”Attention is All You need” (Vaswani et al., 2017). Vaswani et al. (2017) showed that a neural network, which uses only the attention mechanism and leaves out recurrence and convolutions, can achieve state-of-the-art results in English-to-German and English-to-French translation. Although the Transformer was initially used for machine translation, it is now used for a variety of tasks, not limited to natural language processing, such as text classification, object classification, image segmentation, text generation, and more. The original Transformer was an encoder-decoder architecture, but decoder-only generative models, such as GPT-4, can offer good translation quality.

”Attention is All You Need” trained two separate models for the English-to-German and English-to-French benchmarks. However, the translation models do not have to be limited to two

or a small set of languages, but can support hundreds of high- and low-resource languages simultaneously, as demonstrated by M2M-100, NLLB-200 and MADLAD-400 (Fan et al., 2020; N. Team et al., 2022; Kudugunta et al., 2023). Pre-training models on high-resource languages is also beneficial for low-resource languages, because it has been shown that some of the performance, measured using BLEU, gets transferred to low-resource translation pairs (Zoph et al., 2016).

M2M-100, NLLB-200 and MADLAD-400 are text-to-text models, but multilingual models also exist for speech-to-text, text-to-speech and speech-to-speech translation. For example, Seamless M4T is an end-to-end model, which supports text-to-text translation in almost 100 languages, but also adds speech-to-text, text-to-speech and speech-to-speech translation support for a subset of the languages (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023). The main difficulty in supporting speech translations is the lack of high-quality parallel speech-to-text and speech-to-speech data (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023).

2.2 Speech-to-Text Translation

Speech-to-text translation is a multimodal variant of machine translation, where the input is an audio sequence and the output is a text sequence. The following subchapter describes the adjustments that are made to better process audio input and some possible approaches for speech-to-text translation.

2.2.1 Audio Data

Processing audio data is more complicated than text data because it has a very high dimensionality, which is dependent on the sampling rate. Audio encoding models usually expect 16 kHz sampling rate, so each second of audio consists of 16,000 numbers. For example, 30 second audio sample is represented by 480,000 values, which makes processing very long audio sequences computationally expensive.

In theory, a neural network can deal with any number of input features as long as the network and hardware are capable of it, but in practice this is infeasible. In order to reduce the dimensionality of the audio array, one possible approach is to convert the input data to the Log Mel Spectrogram, which in essence transforms the audio data into an image. This is done by Whisper (Radford et al., 2022) and for 16 kHz audio the number of input features is halved. An example of an audio file processed by Whisper is shown in the Figure 1.

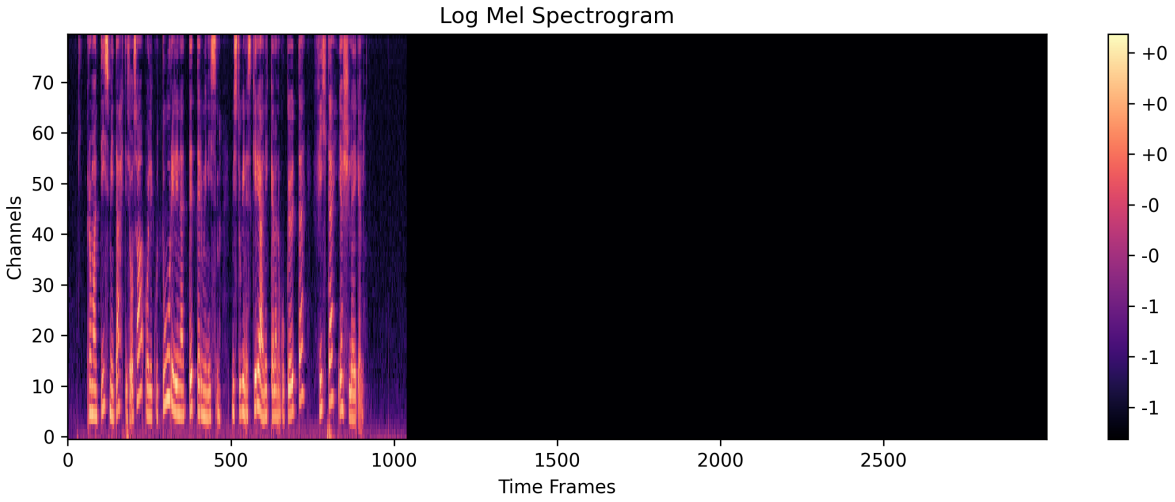

Figure 1. Log Mel Spectrogram created by the Whisper's log_mel_spectrogram method. The audio is a 10s long FLEURS sample that contains Estonian speech. The right hand side values, where usually DBs are shown, are small because it is a log scale spectrogram. The last time frames, equalling to 20 seconds, are empty due to padding.

The audio data is continuous, which presents another problem. It is hard to tell when a word or a sentence stops and a new one begins. Naively chunking the audio data can result in information loss because the chunks may not respect word and sentence boundaries (Macháček, Dabre, and Bojar, 2023). This problem can be alleviated by using pre-trained Voice Activity Detection (VAD) models, which try to segment the audio according to the speech and eliminate silence. One popular VAD model is Silero (S. Team, 2024).

2.2.2 Convolutions and Conformer Architecture

Although the Transformer architecture is still a popular choice for working with audio data, it was originally designed to work with text sequences. In order to enhance its audio processing capabilities, the features of Convolutional Neural Network (CNN) have been introduced. For example, the Whisper model, after converting the raw audio data to the Log Mel Spectrogram, applies two convolutional layers, the GeLU activation function and sinusoidal positional embeddings before forwarding the data to the Transformer blocks (Radford et al., 2022). The audio is processed as an image because a spectrogram is an image of audio and CNNs are widely used for solving computer vision tasks such as object detection and image classification.

There also exist specialized architectures for working with audio data such as Conformer. Conformer, which stands for Convolution-augmented Transformer, is a modified Transformer architecture that is designed to work better with high-dimensional audio data (Gulati et al.,

2020). The Conformer is different from the Transformer architecture in two aspects. Conformer uses convolutional subsampling to reduce the dimensionality of the input data and it replaces the Transformer blocks with Conformer blocks, which include convolutional layers. Gulati et al. (2020) states that the Confromer retains the Transformer's ability to model long-term dependencies and gives it the ability to model local dependencies, which is the strength of CNN models.

The Conformer architecture is used by well-known speech recognition models such as Google's Universal Speech Model (Zhang et al., 2023) and w2v-BERT (Chung et al., 2021). w2v-BERT is also used as a building block in the Seamless M4T (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023).

2.2.3 Systems

Speech-to-text task is generally solved by two different types of translation systems: cascaded and end-to-end. The cascaded system is usually a combination of an automatic speech recognition (ASR) model, which converts the speech into text format, and a text-to-text model, which translates the text to the target language. End-to-end systems use a single model to encode speech and decode text, merging the functionalities of the ASR and text-to-text models into one. End-to-end solutions are becoming more popular, but it is not clear at the moment which of the two approaches is the best (Ahmad et al., 2024b).

An example of a capable cascaded system is Whisper and NLLB, which supports speech-to-text translation in around 100 languages (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023). Whisper might be the most well-known ASR model, but other models such as Universal Speech Model (USM) and AudioPaLM-2-8B-AST have shown comparable results (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023). However, AudioPaLM has not been made publicly available and USM can only be accessed through an API, making Whisper a more accessible option for ASR. Theoretically, it is possible to extend the source-side language support for more than 100 languages as demonstrated by the "Massively Multilingual Speech" project, which pre-trained wav2vec 2.0 models for over 1000 languages (Pratap et al., 2023). The choice of text-to-text translation models is not limited to traditional machine translation models like NLLB, but also LLMs are used (Ahmad et al., 2024a).

The cascaded systems have advantages and disadvantages. Because the cascaded system consists of different models, it is possible to improve or swap out each of them independently of each other, depending on how the field progresses, and also the translation performance of the text-

to-text models is very strong (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023). However, cascaded systems can struggle with poor translation quality due to the error propagation caused by poor transcription quality, which affects low-resource languages more (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023).

Although end-to-end systems are newer than cascaded systems, they have now achieved similar performance (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023). The benefit of the end-to-end system is their simplicity, there is no separate speech recognition model, which also solves the error propagation problem. Examples of well-known end-to-end models are SeamlessM4T, AudioPaLM-2-8B-AST and XLS-R-2B-S2T (Communication, Barrault, Chung, Mariano Cora Meglioli, et al., 2023). If the target language is English, it is possible to use Whisper as an end-to-end speech-to-text translation model (Radford et al., 2022).

2.3 Simultaneous Machine Translation

Simultaneous Machine Translation (SiMT) is similar to the standard machine translation task. The goal remains the same, transforming an input sequence in one language to a sequence in some other language, but now additional requirements are introduced. The simultaneous translation system is expected to start translating before the input sequence has finished, which makes the task more difficult because the model has access only to limited context until the last generation step. Usually, it is not allowed to change previously generated output, so mistakes can force the model to generate wrong output. In addition, there are language specific difficulties, such as differences in word order (M. Ma et al., 2019).

The most important aspect for simultaneous translation systems is finding the optimal balance between latency and translation quality (Cho and Esipova, 2016). Focusing on translation quality will increase latency because the model has to wait for more context to translate correctly. It is possible for the model to wait until the entire input sequence is available, but this turns the simultaneous translation into regular speech-to-text task. Generating output immediately reduces latency, but will most likely result in suboptimal translation quality. This balance between quality and latency is dictated by a policy, which guides the consumption of the input sequence and the generation of the output (Zheng et al., 2020).

In general, policies can be divided into fixed and adaptive policies. Fixed policies are rule-based and tend to be simpler compared to adaptive policies (Zheng et al., 2020). A fixed policy, such as wait-k initially waits for k steps before starting to translate and always translates every sequence

using exactly the same strategy. Wait-k does not take into account the available context, which can cause the model to translate too quickly or too slowly because it does not have the ability to adapt (Zheng et al., 2020). Adaptive policies are more flexible because they take into account the available context at each step before translating, but as a consequence they can be more complex to implement (Zheng et al., 2020). There are approaches that claim to be policy-free because there is no explicitly defined policy function (Koshkin, Sudoh, and Nakamura, 2024), but they can be considered to be part of adaptive policies because the model has implicitly learned the policy during training.

There are different approaches when it comes to model selection. Some fine-tune a model specifically for simultaneous translation such as transLLaMa (Koshkin, Sudoh, and Nakamura, 2024), but it is not strictly necessary because it is possible to use decoding algorithms that support simultaneous translation together with regular translation models (Cho and Esipova, 2016). For example, the ISWLT 2024 submissions relied on Whisper, Seamless M4T and also combinations of different model layers (Ahmad et al., 2024b).

2.4 Evaluation metrics

Evaluating simultaneous speech-to-text systems requires two types of metrics. Translation quality is measured with standard translation quality metrics, and latency, how much of the input is consumed before translating, is measured with simultaneous translation specific metrics.

2.4.1 Quality

Common choices for automatic machine translation quality evaluation are BLEU and chrF++. BLEU is an easy-to-calculate and language-independent word n-gram matching metric (Papineni et al., 2002). It is designed around the idea, that the best translation is done by a professional human translator, so it rates highly translations that are as close to the references as possible. BLEU uses a modified n-gram precision, where the frequency of an n-gram is clipped by its frequency in the reference sentence, penalizing repeating n-grams and unnecessarily long translations, and it uses a reference-length-based brevity score to penalize translations that are too short (Papineni et al., 2002). The matching of the n-grams is position independent and the recommended maximum n-gram length value is 4. It should be kept in mind, that even though the n-gram matches are calculated sentence-by-sentence, the BLEU score applies to the whole test set. BLEU score is also influenced by the number of provided references for each sentence; more references leads to a higher score, even when the generated translations remain the same (Papineni et al., 2002).

Although calculating BLEU is not complicated, it is not always possible to compare BLEU scores from different experiments. The main challenges for comparing BLEU scores are its parameterization options, which allow the use of multiple translation references and changing of the maximum n-gram length, and its dependence on tokenization and processing of references, which can vary for different authors (Post, 2018). These problems can make comparisons misleading, if not meaningless. In order to make the scores comparable, sacreBLEU was proposed by (Post, 2018). It is a Python script that helps with the evaluation of references and allows to export shareable evaluation settings, but it can also be thought of as a rule stating that the references should not be processed by the researchers. However, some issues remain, such as the use of different sentence casing options and the varying sizes of the test sets for the common evaluation datasets (Post, 2018).

chrF++, sometimes also referred to as chrF2++, is a metric that is derived from chrF. chrF is a language and tokenization-independent character n-gram matching metric (Popović, 2015). It uses character n-gram precision and recall values to calculate the translation score. The importance of precision and recall in relation to each other can be adjusted with a parameter β and the ideal maximum n-gram length is 6 (Popović, 2015). chrF++ uses chrF as a base, but adds word unigrams and bigrams, thus making it both a character and word n-gram matching metric (Popović, 2017). chrf++ sets $\beta = 2$; reason why it is sometimes referred to as chrF2++. chrF++ is claimed to correlate better with human assessment of the translation quality compared to the original chrF (Popović, 2017).

2.4.2 Latency

In order to optimize the simultaneous translation system for latency, two commonly used metrics are Average Lagging (AL) and Length-Adaptive Average Lagging (LAAL). These metrics are not concerned with the quality of the translation, but only with the degree of synchronization between the input and output.

Average Lagging (AL) uses the input as a proxy to measure how out-of-sync the translation system is on average when producing the output (M. Ma et al., 2019). AL was proposed to replace two previously used metrics, Consecutive Wait (CW), which measures local latency between consecutively translated words and not overall latency, and Average Proportion (AP), which is disproportionately influenced by the input sequence length, even when the input and output sequences have the same length. M. Ma et al., 2019 defined AL:

$$AL_g(x, y) = \frac{1}{\tau(|x|)} \sum_{t=1}^{\tau_g(|x|)} g(t) - \frac{t-1}{r} \tag{1}$$

where x is the source sequence, y is the generated sequence, $\tau_g(|x|)$ is the step when the entire source sequence is consumed, $g(t)$ is the number of consumed source words for generating y_t and $r = \frac{|y|}{|x|}$. In case of speech-to-text translation, the input sequence x usually consists of fixed-length audio segments.

As an example, assume that x and y are text sequences, $|x| = |y|$ and a simple wait-k policy is used, where $k = 3$, which means that 3 input words are consumed initially before consuming a new word and producing an output word at every following step, then according to Eq. 1 $AL = 3$.

Average Lagging, as defined above, is not suitable for speech-to-text use cases because it is possible for the model to stop generating the output before the input sequence has been consumed, which can happen when the audio ends with a long pause, and in general it favors generated translations that are shorter than reference translations (X. Ma et al., 2020). It is possible for the latency to be negative in these use cases. X. Ma et al. (2020) modified the original AL when creating SimulEval framework in order to account for shorter generations by using the length of the input sequence as a reference instead of the expected translation, which creates the following AL definition:

$$AL_{speech} = \frac{1}{\tau'(|X|)} \sum_{i=1}^{\tau'(|X|)} d_i - d_i^* \tag{2}$$

where X is a sequence of audio segments with duration T_j, $d_i = min\{i|d_i = \sum_{j=1}^{|X|} T_j\}$, $d_i^* = (i-1) \cdot \sum_{j=1}^{|X|} T_j / |Y^*|$ and Y^* is the translation reference sequence.

The problem with the AL, as defined in SimulEval, is that it rewards over-generation meaning that if the model generates longer sentences compared to the reference values, the AL value will be artificially low, which does not correspond to the reality (Papi, Gaido, et al., 2022). Papi, Gaido, et al. (2022) proposed LAAL, which is AL with one change. While AL compares the reference sentence to the input length, LAAL chooses the longest sequence between the reference and the generated sequence and compares that to the input. This change helps with both under-generation and over-generation issues; however, it is important to note, that this still does not solve the issue that speech may contains silences and is in general varying so the

latency scores might not reflect the real users experience (Papi, Gaido, et al., 2022). LAAL uses the same formula as Eq. 2, but redefines $d_i^* = (i-1) \cdot \frac{sum_{j=1}^{|X|} T_j}{\max\{|Y|,|Y*|\}}$, where Y is the generated translation.

3. Methods

There exist different approaches to simultaneous speech translation, which differ in terms of models, training methods, and translation strategies. This chapter describes the chosen models, methods and evaluation setup.

3.1 Translation systems

The following subchapters describe the creation of two types of simultaneous translation systems, that were evaluated in this work. This work tries to adapt open pre-trained models for the simultaneous translation task. The first subchapter focuses on the restrictions regarding openness of the underlying models and hardware limitations because they narrowed down the list of potential solutions.

3.1.1 Restrictions

The models have to be open. It is not important for the entire project to be open source, including publicly available training code and datasets, because the goal of this work is not to validate previous results or reproduce the performance of these models from scratch, but it is important for the models to have open weights so that they can be used without external APIs and permissive licensing, which allows the use of models for research and non-commercial purposes. Most licenses used in ”open” projects support aforementioned use cases.

In addition, there are two hardware constraints. First, it should be possible to run the translation systems in near real time on the Nvidia A100 40GB or V100 32GB GPUs. Near real-time in this context demands that the generation of translations is fast enough that it does not introduce significant extra computational latency in addition to the latency allowed by the translation strategy, although it is possible that some inference optimization is required. For example, the wait-k strategy waits for k steps before starting to translate, thus there is inherently some latency. Due to GPU limitations, it is not possible to select the largest open models, which may offer better translation quality, and it would not make sense to select models that use the entire available VRAM because the inference may become slow. Second, the fine-tuning of the models must be conducted on the GPUs available on the High Performance Computing Center of the University of Tartu (UTHPC), also known as Rocket (University of Tartu, 2018). Rocket provides Nvidia V100 32GB and A100 40GB/80GB GPUs.

3.1.2 Models

Simultaneous speech-to-text translation systems generally use two approaches. Cascaded systems use separate models for speech recognition and translation. End-to-end systems are based on models that can transcribe and translate speech, such as Seamless M4T. ISWLT 2024 results show that cascaded systems can still compete with end-to-end systems, although end-to-end systems tend to be more popular (Ahmad et al., 2024a). Because cascaded approaches are still used, both approaches are evaluated in this work.

Based on the restrictions listed in Chapter 3.1.1, the cascaded system uses **Whisper large-v3-turbo** (809M parameters) for speech recognition and **NLLB-200 distilled 1.3B** for text-to-text translation. Although there exist Whisper models fine-tuned for Estonian language such as **Whisper large-v3-turbo-et-subs** (Fedorchenko and Alumäe, 2025)[1], in this work a system is expected to handle all directions including English-Estonian and Russian-Estonian, thus this model is not the best option. The end-to-end system relies on **Seamless M4Tv2 large** (2.31B parameters), which some teams used for the ISWLT 2024 submissions (Ahmad et al., 2024a). Seamless is a multimodal and multilingual model that supports translating between text and speech modalities. Whisper, NLLB-200 and Seamless models are pre-trained on Estonian, English and Russian, so adding a new language is not needed. The total number of parameters for both systems is comparable, **2.109B** and **2.31B**.

3.1.3 Voice Activity Detection

Preliminary experiments with Whisper large-v3-turbo revealed that it struggles with silences in the input audio, especially when the audio was in Estonian, which has poorer performance compared to the high-resource English and Russian languages. Silences in the input audio can cause Whisper to produce hallucinations and get stuck in producing the same sequence of tokens indefinitely.

It is possible to remove silences using a Voice Activity Detection (VAD) model. It has been shown that if the audio has silences, VAD can help with simultaneous interpretation quality (Macháček, Dabre, and Bojar, 2023). This work used Silero VAD (S. Team, 2024), which is a small and fast model[2], so the effect on inference speed is minimal. Silero VAD detects speech

[1] https://huggingface.co/TalTechNLP/whisper-large-v3-turbo-et-subs
[2] https://github.com/snakers4/silero-vad

in the audio and segments it accordingly. The segmentation eagerness depends on the duration of allowed silence between two consecutive speech segments. If the allowed silence duration is low, then one continuous speech may be broken into two or more segments, and if the allowed silence is too long, then one segment may contain more than one act of speech.

In addition to improving the quality of the input audio, one beneficial side-effect of VAD in the context of this work is that the audio is broken down to smaller chunks, which can be processed quicker. Also, Whisper and Seamless support only chunks with duration up to 30 seconds, but with VAD it is no longer a problem to handle longer audio streams because the speech segments will be shorter, unless the allowed silence duration is too long. In principle, these systems could be then used to translate audio sequences with arbitrary length.

3.1.4 LocalAgreement

Another issue with simultaneous translation systems is the possibility of generating wrong tokens due to the limited input context. In the context of this work, modifying the previous output is not allowed, thus wrong early predictions can cause the system to continue generating wrong output, if the model is forced to use already generated tokens as a prefix.

In order to make the system more stable and reduce the possibility of outputting wrong tokens, one possible solution is to use LocalAgreement. LocalAgreement compares the outputs generated at two consecutive timesteps and only outputs the matching parts, starting from the beginning (Liu, Spanakis, and Niehues, 2020). If there is no match, the last generated tokens are stored in place of previous output and the comparison is done again with new tokens during the next timestep. LocalAgreement has been shown to work better compared to the standard wait-k and hold-k methods (Liu, Spanakis, and Niehues, 2020).

LocalAgreement is a good option in this work because it does not require fine-tuning or changing the underlying models, it can be implemented by changing the inference process. It is possible to use it with any translation model. One drawback when using LocalAgreement with other strategies such as wait-k is that it increases latency. First, in order to generate any output, two consecutive steps must have a match, so it is not possible to output anything after k steps because there is no previous output, which can be used in comparison. In this situation, the minimum wait before the first generation step is actually k+1. It is possible that the model keeps changing output every step while nothing has been commited, so in the worst case scenario the model might not produce anything before the input audio has been consumed, thus translating the

sentence in non-simultaenous manner. However, the benefits of LocalAgreement outweigh its disadvantages.

In addition to proposing LocalAgreement, Liu, Spanakis, and Niehues (2020) showed that it is possible to improve the simultaneous translation performance of pre-trained translation models by fine-tuning on a 1:1 mix of full and partial sequences. Partial sequences were generated by taking from the beginning of each sample's source audio and target text $10 - 40\%$ chunks (Liu, Spanakis, and Niehues, 2020). The same approach was adopted in this work. This also doubles the number of samples in the dataset.

3.2 Finetuning

The NLLB-200 and Seamless models were fine-tuned on the Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian translation directions. A single A100 80GB GPU was used for each fine-tuning run and during one run the model was trained on all 4 directions. The training dataset is described in Chapter 4.5.

The Seamless M4Tv2 large model was fine-tuned using the Seamless m4t_finetune script[3]. The script uses AdamW optimizer and scaled Noam learning rate scheduler [4]. The parameters for fine-tuning were following: maximum epochs 15, learning rate 1e-6, batch size 64, warmup steps 2000, eval steps 10000 and patience 10. The last two parameters are important for early stopping the training. It is important to note, that the fine-tuned Seamless model created by the m4t_finetune script and the model that is expected by the transformer library SeamlessM4Tv2ForSpeechToText class[5], which was used for inference, use different layer names; there are over 1000 mismatches. In order to use the fine-tuned model with the transformer library, the layer names were mapped to the expected names so that the weights could be correctly loaded.

It is possible to train NLLB-200 models using the fairseq2 library, but in this case custom python and pytorch script was used instead. The script also incorporated mixed precision training using torch.autocast and torch.GradScaler. GradScaler helps with vanishing gradients, which might occur when training the models with torch.float16 datatypes[6]. The training script used a linear

[3]https://github.com/facebookresearch/seamless_communication/tree/main/src/seamless_communication/cli/m4t/finetune
[4]https://github.com/facebookresearch/fairseq2/blob/main/src/fairseq2/optim/lr_scheduler/_myle.py
[5]https://huggingface.co/docs/transformers/model_doc/seamless_m4t_v2
[6]https://docs.pytorch.org/docs/stable/amp.html

learning rate scheduler instead of scaled Noam scheduler and AdamW optimizer. The learning hyperparameters are the same as above except for the batch size, which was set at 16.

For fine-tuning, the maximum allowed training time was 8 days. It is the default wall-time limit enforced by the Rocket cluster.

3.3 Evaluation

All evaluation runs are were conducted using SimulEval framework. In order to understand how the evaluation works, this subchapter briefly describes SimulEval before proceeding to the evaluation setup.

3.3.1 SimulEval

SimulEval is a Python framework for evaluating simultaneous translation and speech recognition systems (X. Ma et al., 2020)[7]. In order to evaluate a system with SimulEval, the system has to implement an agent class. There are four different types of agent classes: SpeechToText, SpeechToSpeech, TextToSpeech and TextToText. In short, if the system implements SpeechToText agent, it will receive an audio stream as input and it is expected to return text output. Agents can be combined using the AgentPipeline. For example, a cascaded system can use SpeechToText and TextToText agents sequentially instead of a single SpeechToText agent. The process, which guides the consumption of the input and the output generation, is dictated by the policy function, which must be implemented by the user.

Although SimulEval is developed to evaluate simultaneous translation systems, it is very easy to evaluate offline translation systems. This can be done by defining a policy function that waits for the source sequence to finish before producing the entire translation at once. This approach is used in this work to evaluate the offline baseline systems instead of writing a separate evaluation script.

SimulEval uses BLEU as the default translation quality metric. It is possible to add new quality metrics by implementing the QualityScorer class, which has __call__ method that has to return a quality score. By default SimulEval does not support chrF++. In order to evaluate the sentences using chrF++ it was implemented using the evaluate library, which depends on the sacrebleu library. An additional benefit of the SimulEval library is that when the output folder is preserved,

[7]https://github.com/facebookresearch/SimulEval

it is possible to recalculate the scores using a different metric without generating the translations again. Thus BLEU and chrF++ scores can be calculated by one actual evaluation run.

In addition to the translation quality metrics, SimulEval measures latency metrics. The output includes Length-Adaptive Average Lagging (LAAL), Average Lagging (AL), Average Proportion (AP), Differentiable Average Lagging (DAL) and Average Translation Delay (ATD). LAAL and AL are explained in Chapter 2.4.2.

3.3.2 Evaluation setup

The cascaded system was created using the built-in VADAgent (SpeechToSpeech), included in the seamless repository and based on Silero VAD, WhisperAgent (SpeechToText) and NLLBAgent (TextToText). LocalAgreement strategy was implemented in the NLLBAgent. The WhisperAgent used beam size of 5 and the number of maximum tokens was set at 30. The beam size 5 was used in the WhisperAgent to make the output more stable because the Whisper model was not fine-tuned. The NLLBAgent used greedy decoding. At the end of each generation step, unless it was the final step, the last token from the NLLB output was discarded to make the generation more stable.

The end-to-end system was implemented using the same VADAgent and a SeamlessAgent (SpeechToText). The LocalAgreement strategy was implemented in the SeamlessAgent class. The generation used beam size of 5 and the number of maximum tokens was set to 30. During each non-final generation step the last token from the output was discarded.

The SimulEval source segment size parameter, which determines the size of the audio chunks that the system receives, was set at 1000ms and the silence limit for Silero VAD was set at 500ms. The impact of the size of the audio chunks on the translation quality together with Local Agreement has been studied by Polák et al. (2022), but there it was combined with a hold-k not the wait-k strategy. The default quality metric was BLEU and the chrF++ scores were calculated afterwards using the output folder. The output of the translation systems was not post-processed and the scores were calculated directly using the references. The evaluation dataset is described in Chapter 4.6.

The code for the agents and the scripts for running the SimulEval are available in the GitHub[8].

[8]https://github.com/Henrik895/est-simt-s2tt

3.3.3 Baselines

The problem with metrics such as BLEU and chrF++ is that they output a single number, that by itself is hard to understand. It is impossible to conclude whether a BLEU score of 25 for simultaneous Estonian-English translation is good or bad without taking into account additional context. In order to provide context for the evaluation scores, two baselines are established. Similar baseline setup was used by Liu, Spanakis, and Niehues (2020), but the lower baseline was defined differently.

The first baseline establishes the lowest acceptable limit for the BLEU and chrF++ scores. This baseline takes the models from the cascaded and end-to-end approaches and measures their performance with the wait-k 2 strategy, which waits 2 seconds before starting to translate, and limiting the number of tokens to 30 for one generation step, which prevents the system from getting stuck in the loop. Also, the last word was withhold at each step, increasing the quality further. The scores after implementing simultaneous translation-specific strategies and fine-tuning the models must be higher compared to the scores provided by this baseline. Lower scores would be an indication of a mistake in the training or evaluation process.

The second baseline sets a relatively high benchmark using the non-simultaenous translation scenario, which consumes the entire input sequence before generating the translated sentence at once. In short, this is regular speech-to-text translation and the resulting BLEU and chrF++ scores should be significantly higher compared to the first baseline. Considering that in the simultaneous translation scenario the systems do not have access to the entire input sequence and the word order in source and target sentences may differ, these scores should be hard to achieve.

Both baselines include BLEU and chrF++ scores for the English-Russian and Russian-English directions. English and Russian are high resource languages, which open models such as Whisper, NLLB-200 and Seamless M4T support, and even though these directions are not in the focus of this work, they can be used as proxies for high-resource translation performances, useful for comparisons.

Using these two baselines as guides, it is possible to evaluate the performance of the fine-tuned models. If the performance is similar to the first baseline, then the approach can be considered unsuccessful, but if they approach or even manage to exceed the second baseline, then the results can be considered successful.

4. Data

There are no large publicly available speech-to-text datasets that cover Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian translation directions. Similar datasets have been created, but not released (Sildam, Velve, and Alumäe, 2024). There are smaller datasets, such as FLEURS, which include all of these directions (Conneau et al., 2022). Due to size, these datasets are useful for experimentation and benchmarking. This chapter describes the process of creating a combined parallel speech-to-text Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian dataset, which was used for fine-tuning. English-Russian and Russian-English directions were not included in the fine-tuning because the focus of this work is on the Estonian directions. English and Russian are high-resource languages and there is naturally more interest towards translating between these languages.

The general preprocessing steps are described first because they are applicable to all the datasets mentioned in the subsequent chapters. Dataset specific preprocessing steps are described in their respective chapters.

4.1 Preprocessing

The datasets included in this work contain automatically or manually transcribed audio data, but do not include translations. In order to create a parallel speech-to-text dataset, the transcriptions have to be translated to the target languages. This work used translations generated beforehand by the TalTech Laboratory of Language Technology and the TartuNLP group.

Data preprocessing and filtering can have a great impact on fine-tuning results. ”Garbage in, garbage out” is a well-known statement in regards to data quality, which emphasizes the importance of removing noisy and low quality samples. However, minimal preprocessing was used in this work. This decision was made because the source datasets were already filtered and curated in some capacity. Due to the number of total samples, the average sample should have sufficient quality and the remaining outliers should not strongly affect fine-tuning results.

The following preprocessing steps were taken to remove the worst examples of low quality data and to make the audio files compatible with Whisper, not fine-tuned, and Seamless M4T. The minimum audio length for each source audio sample was set to 1 second to remove short samples without speech. This requirement is also important for generating partial sequences. Otherwise, there would have been extremely short samples in the training dataset. The maximum length was set at 30 seconds because Whisper and Seamless do not support processing of longer audio

chunks. In addition, Seamless only supports audio files with a 16000 Hz sampling rate, so every audio file with different sampling rate was resampled using Torchaudio Resample and saved as a wav file. The default method for Torchaudio Resample is sinc interpolation with Hann window[9]. The translated transcriptions were not modified, except for generating partial sequences.

Partial sequences were created from every remaining sample-translation pair. This was done according to the process described in Chapter 3.1.4. Because the length of the source audio samples was between 1 and 30 seconds, the length for partial sequences was between 0.1 and 12 seconds. The creation of partial sequences doubled the number of total samples.

4.2 Estonian-English and Estonian-Russian datasets

The Estonian-English and Estonian-Russian speech-to-text parallel datasets are based on the **TalTech Estonian Speech Dataset 1.0** train split [10]. The dataset contains 1334 hours of Estonian audio data, not exlusively speech, together with manually created transcriptions (Alumäe et al., 2023). The development and test splits, which had respectively 21 and 23 hours of audio data, were excluded.

The TalTech speech dataset is made up of three different data sources, which are broadcast data, lecture data and parliament speech data. The broadcast dataset is mostly made up of Estonian Public Broadcasting television shows, such as Aktuaalne Kaamera, Terevisoon, Ringvaade, and radio broadcasts, such as Välismääraja and Reporteritund. Examples of the lecture data are TEDxTartu speeches, HITSA web seminars and TTU development conferences. Parliament speech data is self-explanatory.

The train split has 581647 transcriptions with timestamps. The transcriptions are in TRS, STM and VTT formats. The duration of the shortest transcription is 0.003 seconds and the longest 723.191 seconds. There were a small number of transcriptions with negative length, but they were excluded together with transcriptions shorter than 1 second or longer than 30 seconds. In total, 14671 transcriptions were excluded. 11249 transcriptions were shorter than 1 second and 3422 were longer than 30 seconds. The audio files were cut into smaller chunks based on the remaining 566976 transcriptions, one chunk per transcription, and matched to their respective translations, thus creating parallel speech-to-text translation datasets. The translations were

[9]https://docs.pytorch.org/audio/2.7.0/generated/torchaudio.transforms.Resample.html
[10]https://cs.taltech.ee/staff/tanel.alumae/data/est-pub-asr-data/

generated previously by TalTech Laboratory of Language Technology[11]. Resampling was not necessary because all audio files already used a 16000 Hz sampling rate.

After generating partial sequences, both Estonian-English and Estonian-Russian datasets had 1133952 samples. The total audio duration for both datasets is around 1500 hours, which can be calculated by combining the training and validation splits shown in Table 3.

4.3 English-Estonian dataset

The English-Estonian dataset is based on 6 publicly available speech datasets: **LJSpeech-1.1**, **LibriSpeech**, **TED-LIUM Release 3**, **Common Voice 21.0 2025-03-14**, **Tatoeba** and **Voxpopuli**. None of the listed datasets came with Estonian translations. The speech transcriptions were translated into Estonian by the TartuNLP group using Neurotõlge API.

LJSpeech-1.1 dataset contains 13100 audio files with a total length of roughly 24 hours (Ito and Johnson, 2017). The audio files are short sentences from seven different books read by one speaker. All audio files have a duration between 1 and 30 seconds, so none were filtered out. The original audio files are in the mp3 format and use 22050 Hz sampling rate, so they were resampled to 16000 Hz and saved as wav files.

LibriSpeech is based on audio books from the **LibriVox** project (Panayotov et al., 2015). The dataset contains multiple splits: train-clean-100, train-clean-360, dev-clean, test-clean, train-other-500, dev-other and test-other. Splits with suffix -clean contain good quality speech and splits with -other also include speech with more errors (Panayotov et al., 2015). The numbers at the end of the split refer to the approximate hours of speech data in the split. This work selected the train-clean-360 split, which contained 95404 samples. All audio files had desired length, so none were filtered out. The audio files use the flac format and 16000 Hz sampling rate.

TED-LIUM Release 3 is based on the TED Conference speeches (Hernandez et al., 2018). It contains 452 hours of transcribed English speech distributed over 268263 samples. The number of previously cleaned and translated samples was 114813, so only these samples were used. 114801 samples remained after filtering out short and long samples. The audio files are wav files with a 16000 Hz sampling rate.

[11] https://cs.taltech.ee/staff/tanel.alumae/data/train-et2en-et2ru.zip

Common Voice 21.0 2025-03-14 was the largest included English speech dataset. Common Voice is a crowdsourced dataset created by volunteers reading prepared statements in their chosen languages (Ardila et al., 2020). The validated split with 1845370 samples was used, but one was removed during translation. It should be noted, that validated does not refer to the validation split, but to the samples that have passed quality control in contrast to the samples in the invalidated split (Ardila et al., 2020). 135 of the 1845369 translated samples were filtered out. For Common Voice resampling was necessary because the original mp3 files used 32000 Hz sampling rate. The resampled 16000 Hz files were saved in the wav format.

Similarly to the Common Voice dataset, **Tatoeba**[12] is also crowdsourced. It does not have any splits and contains 300077 samples of which 300076 remained after translating. 127 samples were filtered out for being short or long. Interestingly, there might have been some errors in the translation process because 523 transcriptions did not have corresponding audio files anymore, meaning the files were lost. These samples were removed as well. Nevertheless, 299427 samples remained after filtering. The original mp3 files used 32000 Hz sampling rate, thus they were resampled and converted to wav format.

VoxPopuli is based on European Parliament speeches (Wang, Riviere, et al., 2021). It contains unlabeled and transcribed speech data. The proportion of unlabeled data is much higher, but only transcribed data is useful in the context of this work. The train split of the transcribed English speech data was used, which originally contained 182483 and after translating 182466 samples remained. 2148 samples were filtered out. The audio files are in ogg format and use a 16000 Hz sampling rate.

In total, 2548283 samples remained after the creation of translations, not done as part of this work, and subsequent filtering, which removed 2945 samples. After generating partial sequences as described in Chapter 3.1.4, the English-Estonian dataset contained 5096566 samples. The total length in hours and the average duration of the samples are shown in Table 1.

4.4 Russian-Estonian dataset

The Russian-Estonian dataset is based on 7 publicly available Russian speech datasets: **Common Voice 20.0 2024-12-06**, **Golos (opus format)**, **private_buriy_audiobooks_2**,

[12]https://downloads.tatoeba.org/audio/

Source	Samples	Hours	Avg. duration (seconds)
LJSpeech-1.1	26.2K	29.9	4.11
LibriSpeech	190.8K	416.4	7.86
TED-LIUM Release 3	229.6K	210.2	3.30
CV 21.0 2025-03-14	3690.5K	3276.1	3.20
Tatoeba	598.9K	231.2	1.39
VoxPopuli	360.6K	634.8	6.34
Total	5096.6K	4798.5	3.39

Table 1. Overview of the English-Estonian dataset. The samples, duration and avg. hours include partial sequences, that were generated after filtering.

public_youtube1120, **radio_2**, **Russian LibriSpeech** and **TEDxRU**. Similarly, these datasets did not include Estonian translations. The transcriptions were translated to Estonian by TartuNLP group using Neurotõlge API.

Common Voice 20.0 2024-12-06 Russian speech corpus has been crowdsourced by volunteers reading prepared texts in Russian (Ardila et al., 2020). The validated split, not validation, contains 168634 samples. 168633 remained after translations and 14 more samples were removed after filtering. The original mp3 files used 32000 Hz sampling rate, so they were resampled to 16000 Hz and saved using the wav format.

Golos (opus format) dataset is a manually annotated Russian speech dataset, which has been mostly crowdsourced and also crowd validated (Karpov, Denisenko, and Minkin, 2021). The train split contains 1103799 samples, of which 1094017 were translated. Filtering removed 911 samples that were either too short or long. The audio files are in opus format and use 16000 Hz sampling rate.

private_buriy_audiobooks_2 is part of a larger Russian speech-to-text dataset called Open STT[13]. This dataset contains 1511 hours of automatically annotated audio book data distributed over 114904 samples, of which all remained after translating. 1145 samples were removed during filtering and 1 translated sample did not have a corresponding audio file. The audio files use 16000 Hz sampling rate and are in opus format.

[13]https://github.com/snakers4/open_stt

public_youtube1120 is part of the same Open STT dataset as private_buriy_audiobooks_2. It is based on Youtube subtitle data. It has 1410911 samples, totaling 1104 hours, and all of them remained after translating. Filtering for audio duration removed 6713 samples. The audio files are in opus format and use 16000 Hz sampling rate.

radio_2 is the last of the datasets included here that is part of the Open STT dataset. This dataset contains 1439 hours of radio data. All 651645 samples were successfully translated, but 34301 samples were removed during filtering, which is the most for any Russian or English speech dataset included in this work. The opus audio files use a 16000 Hz sampling rate.

Russian LibriSpeech[14], also refered to as **RuLS**, is composed of audiobook data based on the **LibriVox** dataset. It contains 98 hours of audio data and is divided into train, dev and test splits, of which only train split was used. The train split contains 54472 samples, of which 50260 were translated. Filtering did not remove any samples. The audio files are in wav format and use a 16000 Hz sampling rate.

TEDxRU dataset is part of the **Multilingual TEDx** dataset, which contains speech data in different languages (Salesky et al., 2021). The TEDxRU is divided into train, validation and test splits. The train split contains 29161 samples and all of them remained after translating. Further filtering removed 1770 samples. The original audio files are in wav format and use a 44100 Hz sampling rate, so they were resampled to 16000 Hz and stored in the same format.

In total, after combining all translated Russian-Estonian datasets and filtering out 44855 invalid samples, 4509176 full samples remained. The final number of samples after generating partial sequences, as described in section 4.1, rose to 9018352. The total length in hours and the average duration of the samples are shown in the Table 2.

4.5 Combined dataset

The final fine-tuning dataset was created by combining the previously created Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian datasets. The Estonian-Russian and Estonian-English datasets have fewer examples compared to the English-Estonian and Russian-Estonian datasets, so simply combining them all would have created an imbalanced dataset. In order to balance the datasets, undersampling was used, so every sample from the Estonian-

[14]https://openslr.org/96/

Source	Samples	Hours	Avg. duration (seconds)
CV 20.0 2024-12-06	337.2K	300.1	3.20
Golos (opus format)	2186.2K	1520.6	2.50
private_buriy_audiobooks_2	2296.5K	1881.2	2.95
public_youtube1120	2808.4K	1378.2	1.77
radio_2	1234.7K	1504.0	4.39
RuLS	100.5K	105.8	3.79
TEDxRU	54.8K	58.6	3.85
Total	9018.4K	6748.5	2.69

Table 2. Overview of the Russian-Estonian dataset. The samples, duration and avg. hours include partial sequences, that were generated after filtering.

English and Estonian-Russian datasets was included, but only an equal amount of data from the English-Estonian and Russian-Estonian datasets. In the end, each translation direction had 1133952 samples with a 1:1 mix of full and partial sequences. It would also have been possible to use oversampling to increase the amount of data in the Estonian-English and Estonian-Russian datasets, but there was more than enough data so it was not deemed necessary. It also important to note that partial sequences were created from the original examples, which may be viewed as a weak form of oversampling.

The English-Estonian and Russian-Estonian datasets were created by combining datasets, which had different sizes ranging from tens of thousands to millions of examples. Randomly sampling 1133952 examples from each of them would have meant that the largest datasets would have made up most of the random samples without taking into account data quality or domain. A fairer approach to sampling was used, which tried to take into account dataset sizes.

In order to make sure that there are as many samples from each dataset as possible, the required number of samples was divided by the number of source datasets, 6 for English-Estonian and 7 for Russian-Estonian. The result was used as the required number of samples for each source dataset. If the source dataset had fewer samples, the entire dataset was included, and if the dataset had more samples, the required number of samples was taken and the remaining samples were sent to the pool of unselected samples. In order to compensate for the size of the smaller datasets, the missing number of samples was randomly sampled from the pool of remaining samples.

This process resulted in a combined Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian dataset with 4535808 samples. The dataset was divided into train and validation splits using the 95-5 split. The training split had 4309017 and the validation split 226791 samples. The decision to use 95-5 split instead of more commonly used splits such as 80-20 or 90-10 was made on the basis that the validation split is not used to evaluate the performance of the model after the training has completed, but to allow the fine-tuning to terminate early if the performance stops improving. The train and validation split compositions are shown in Table 3.

	Train			Validation		
Source	Smpl	Hrs	Avg. (s)	Smpl	Hrs	Avg. (s)
Estonian-English	1077.6K	1438.9	4.81	56.3K	75.0	4.79
Estonian-Russian	1077.2K	1438.1	4.81	56.8K	75.8	4.81
LJSpeech-1.1	24.9K	28.4	4.11	1.3K	1.5	4.12
LibriSpeech	179.5K	391.8	7.86	9.5K	20.8	7.86
TED-LIUM Release 3	181.0K	165.7	3.30	9.7K	8.9	3.31
CV 21.0 2025-03-14	310.5K	275.7	3.20	16.5K	14.5	3.15
Tatoeba	195.0K	75.3	1.39	10.3K	4.0	1.39
VoxPopuli	186.0K	327.3	6.33	9.7K	17.3	6.38
English-Estonian	1076.9K	1264.1	4.23	57.1K	66.9	4.22
CV 20.0 2024-12-06	157.4K	140.0	3.20	8.3K	7.3	3.21
Golos (opus format)	194.2K	135.0	2.50	10.3K	7.2	2.54
private_buriy_audiobooks_2	196.2K	160.4	2.94	10.3K	8.6	2.99
public_youtube1120	206.7K	101.4	1.77	10.9K	5.4	1.79
radio_2	175.3K	213.6	4.39	9.1K	11.3	4.48
RuLS	95.6K	100.6	3.79	4.9K	5.2	3.81
TEDxRU	51.9K	55.6	3.85	2.8K	3.0	3.79
Russian-Estonian	1077.4K	906.6	3.03	56.6K	48.1	3.06
Total	4309.0K	5047.8	4.22	226.8K	265.8	4.22

Table 3. Overview of the train and validation splits used for fine-tuning.

The scripts that were used to create the dataset are available on GitHub[15].

4.6 Evaluation dataset

Simultaneous speech-to-text translation systems can be evaluated on the same datasets as regular speech-to-text translation models. Instead of giving the system access to the entire source sequence, streaming is used, which feeds the system chunks of audio with a specified length. SimulEval framework, described in Chapter 3.3.1, works this way. Because the fine-tuning dataset was entirely synthetic, the decision was made to look for a higher quality benchmark-focused dataset, which included all six translation directions.

One dataset that fits the criteria is FLEURS, which is a n-way parallel dataset in 102 languages (Conneau et al., 2022). FLEURS includes all 6 translation directions, that are evaluated in this work, and the translations have been created by human translators, which should result in higher translation quality compared to the synthetic machine translated datasets. FLEURS dataset is divided into train, dev and test splits. The splits contain 1509, 150 and 350 samples (Conneau et al., 2022).

This work only uses the test split. Training and dev splits were discarded. Initially, the test set contained 350 samples, but after filtering the English and Russian datasets for sentences that are present in the Estonian dataset, 339 samples remained. One additional benefit of FLEURS dataset is that the sentences with matching ids have the same meaning, so for all translation directions the sentences are from the same domain and roughly equal in length.

[15]https://github.com/Henrik895/est-simt-s2tt

5. Results

This chapter starts by describing the results of the baselines, which gives context for evaluating the cascaded and end-to-end systems. After baselines, the cascaded and end-to-end systems are compared with and without finetuning, in order to determine if the systems can be used without finetuning and how finetuning affected the end results. The translation quality metrics are BLEU and chrF++. The order of quality scores might be different for BLEU and chrF++ because BLEU compares words but chrF++ includes n-grams. Translation latency is measured using LAAL[16] and AL[17]. The target AL is 3 seconds or less. This is more lenient compared to the IWSLT competition, which uses a 2 second limit for AL (Ahmad et al., 2024a), but in this case the focus is more on the translation quality. Latency metrics are omitted for the non-simultaneous baseline. Average scores are shown separately for est-x and x-est directions because only these directions were fine-tuned.

5.1 Baselines

The non-simultaneous baseline results are shown in Table 4. The results are similar for both models. Whisper + NLLB-200 has a slightly better BLEU score, but Seamless scores higher in chrF++. The strongest translation direction for Whisper + NLLB-200 is Russian-English with 32.34 BLEU (57.06 chrF++), but Estonian-English is second with 29.77 BLEU (55.13 chrF++). For Seamless, the strongest direction was Estonian-English with 31.28 BLEU (57.08 chrF++), which is surprising because Russian-English is a high-resource language pair. Looking at the BLEU and chrF++ scores, it is clear that these models are good at translating into English, followed by Russian and then Estonian. The target language seems to be the most important factor.

The weakest translation directions for both models are Estonian-Russian and Russian-Estonian. On the Russian-Estonian dataset Whisper + NLLB-200 scores 17.33 BLEU (42.77 chrF++) and Seamless 11.54 BLEU (39.99 chrF++). Scores for Russian-Estonian are similar with 13.64 BLEU (42.61 chrF++) for Whisper + NLLB-200 and 11.17 BLEU (43.19 chrF++) for Seamless. It is clear that finetuning for these directions will be required because the base performance is not good enough compared to Estonian-English and English-Estonian directions.

[16]Length Adaptive Average Lagging
[17]Average Lagging

	Whisper + NLLB-200		Seamless M4Tv2	
Direction	BLEU	chrF++	BLEU	chrF++
Est-Eng	29.77	55.13	31.28	57.08
Est-Rus	17.33	42.77	11.54	39.99
Eng-Est	17.67	47.32	22.75	53.14
Eng-Rus	25.15	50.10	26.43	52.23
Rus-Est	13.64	42.61	11.17	43.19
Rus-Eng	32.34	57.06	30.33	56.88
Average	22.65	49.17	22.25	50.42
Average (Est)	19.60	46.96	19.18	48.35

Table 4. Non-simultaneous baseline evaluation results.

Although the BLEU scores for directions that include English are similar to previously reported BLEU scores for the FLEURS test split (Communication, Barrault, Chung, Mariano Coria Meglioli, et al., 2023), it should be noted that in this work 339 samples out of 350 were used for evaluation. The removal of 11 samples was described in Chapter 4.6. For this reason, the BLEU scores are slightly different.

The fixed wait-k 2 baseline results are shown in Table 5. As expected, the BLEU and chrF++ scores are significantly lower compared to the non-simultaneous baseline. The Whisper + NLLB-200 and Seamless differ in results depending on whether the Estonian is on the source or target side. On the Estonian-English and Estonian-Russian directions, Whisper + NLLB-200 is better, scoring 14.96 BLEU (39.15 chrF++) and 9.29 BLEU (31.28 chrF++), although Seamless has a better chrF++ on the Estonian-Russian despite the lower BLEU score. Seamless was stronger on the English-Estonian and Russian-Estonian directions, scoring 14.33 BLEU (45.57 chrF++) and 6.71 BLEU (36.44 chrF++). It is surprising that Seamless scored only 8.00 BLEU (27.97 chrF++) on the Estonian-English direction because it was its strongest direction in the non-simultaneous baseline. In general, both models struggled more with Estonian-Russian and Russian-Estonian directions, which could be expected after non-simultaneous baseline results.

The biggest difference between the two models is the latency. For Whisper + NLLB-200, the LAAL and AL scores are relatively in line with what can be expected from wait-k 2 policy, but the scores for Seamless are low or even negative. As described in Chapter 2.4.2, over-generation can cause the AL values to be artificially low or negative. This indicates that sometimes the

	Whisper + NLLB-200				Seamless M4Tv2			
Direction	BLEU	chrF++	LAAL	AL	BLEU	chrF++	LAAL	AL
Est-Eng	14.96	39.15	1718.09	819.55	8.00	27.97	-508.50	-2117.11
Est-Rus	9.29	31.28	2112.58	1427.85	6.73	32.06	499.59	-1279.21
Eng-Est	11.76	39.52	1848.53	1478.00	14.33	45.57	772.19	77.00
Eng-Rus	17.16	42.41	1797.69	1482.52	19.67	45.44	998.10	650.26
Rus-Est	4.75	31.37	2110.58	606.44	6.71	36.44	855.85	-653.19
Rus-Eng	9.27	38.39	1594.31	-3521.77	16.01	40.27	-1.21	-1121.63
Average	11.20	37.02	1863.63	382.10	11.91	37.96	436.00	-740.65
Average (Est)	10.19	35.33	1947.45	1082.96	8.94	35.51	404.78	-993.13

Table 5. Simultaneous wait-k 2 baseline evaluation results.

model got stuck in loops, which was confirmed by examining the evaluation output. The 30 token generation limit for each generation step has a big impact on the latency scores. Reducing the limit would make the latency scores higher because it would eliminate over-generation.

5.2 LocalAgreement + VAD without fine-tuning

The results of the evaluation without fine-tuning are shown in Table 6. Using the wait-k 2 policy together with Voice Activity Detection (VAD) and LocalAgreement significantly improved the translation quality compared to the wait-k 2 baseline. This was especially noticeable for the end-to-end system, which scored 25.28 BLEU (52.86 chrF++) on Estonian-English and 17.64 BLEU (46.77 chrF++) on English-Estonian directions. The Estonian-English result is close to the high-resource Russian-English, although the latency is higher, which indicates that the model is less confident in its output. The latency for Estonian-English is 2890.17 LAAL and 2216.07 AL, but for Russian-English the numbers are 2571.46 and 1873.12. The translation quality for Estonian-Russian and Russian-Estonian is close to the non-simultaenous baseline, but the starting point was not strong to begin with. In general, the end-to-end system managed to achieve on Estonian directions average BLEU of 16.01 (45.20 chrF++) compared to the non-simultaneous 19.18 BLEU (48.35 chrF++).

The results for the cascaded system improved also compared to the wait-k 2 baseline, but not as much as the end-to-end system. On the Estonian directions, the baseline scored on average 10.19 BLEU (35.33 chrF++) and the cascaded system 15.72 BLEU (43.03 chrF++). The strongest Estonian direction was Estonian-English with 23.63 BLEU (49.13 chrF++), which was lower

	Cascaded				End-to-end			
Direction	BLEU	chrF++	LAAL	AL	BLEU	chrF++	LAAL	AL
Est-Eng	23.63	49.13	3407.10	3012.14	25.28	52.86	2890.17	2216.07
Est-Rus	13.66	38.76	3687.77	3367.94	10.61	39.01	3127.96	2765.75
Eng-Est	14.16	42.88	3266.16	3084.35	17.64	46.77	2891.44	2557.51
Eng-Rus	18.63	43.79	3094.31	2885.07	20.37	45.69	2956.38	2748.28
Rus-Est	11.45	41.33	3398.06	3058.12	10.50	42.15	2968.23	2404.48
Rus-Eng	28.20	53.67	2972.31	2486.32	26.53	54.32	2571.46	1873.12
Average	18.29	44.93	3304.29	2982.32	18.49	46.80	2900.94	2427.54
Average (Est)	15.72	43.03	3439.77	3130.64	16.01	45.20	2969.45	2485.95

Table 6. Cascaded and end-to-end simultaneous evaluation results without finetuning.

than Russian-English 28.20 BLEU (53.67 chrF++). Similarly to the end-to-end system, the weakest directions were Estonian-Russian and Russian-Estonian, although for these directions the cascaded system achieves better quality. In general, the two systems are roughly equal in terms of translation quality.

The biggest drawback for both systems is the increase in latency. On the Estonian directions, the cascaded system measured 3439.77 LAAL and 3130.64 AL and end-to-end system 2969.45 LAAL and 2485.95 AL. For comparison, the wait-k 2 baseline measurements for the cascaded system were 1947.45 LAAL and 1082.96 AL and for the end-to-end system 404.78 LAAL and -993.13 AL. The change in latency scores is explained by using the LocalAgreement matching strategy, which allows the model to output only the translations that it is confident in. The AL score for the end-to-end system is not negative anymore, so the over-generation issue was fixed. These latency scores are roughly in line with what is allowed in this work, but relatively high compared to the limits in IWSLT competitions, so depending on the direction these systems may not be suitable for low-latency use cases.

5.3 LocalAgreement + VAD with fine-tuning

In addition to the first fine-tuning run, the Seamless was fine-tuned a second time with patience set to 20. During the first run, the fine-tuning terminated after 4 epochs with an eval loss of 4.0030. The inspection of training logs revealed that one evaluation run had a relatively low loss and even though the average loss was trending down, it was not able to match the best loss in the next 10 evaluation steps. In order to eliminate the possibility of ”bad luck”, a second run was

conducted with all parameters except patience remaining the same. The second run terminated after 10 epochs, but the eval loss improved marginally, finishing with 3.9754. The results in Table 7 are from the second run. NLLB was fine-tuned only once.

	Cascaded				End-to-end			
Direction	BLEU	chrF++	LAAL	AL	BLEU	chrF++	LAAL	AL
Est-Eng	23.10	51.75	2955.32	2333.30	26.12	55.10	2659.10	1942.26
Est-Rus	14.36	40.55	3220.01	2782.86	15.61	43.16	2897.39	2491.57
Eng-Est	16.43	46.20	2862.29	2625.65	17.43	47.36	2733.94	2367.10
Eng-Rus	18.51	44.40	2864.64	2658.96	20.07	45.84	2758.25	2493.06
Rus-Est	12.99	44.05	2963.74	2459.90	12.71	44.41	2861.92	2219.15
Rus-Eng	24.41	53.32	2587.21	1855.81	24.86	53.58	2466.22	1530.19
Average	18.30	46.71	2908.87	2452.75	19.46	48.24	2729.47	2173.89
Average (Est)	16.72	45.64	3000.34	2550.43	17.97	47.51	2788.09	2255.02

Table 7. Cascaded and end-to-end evaluation results after fine-tuning.

The results show that both systems benefitted from fine-tuning. The cascaded system achieved 16.72 BLEU (45.64 chrF++) on the Estonian directions and the end-to-end system scored 17.97 BLEU (47.51 chrF++). Overall, the end-to-end system improved more. The biggest improvement for the end-to-end system can be observed on the Estonian-Russian direction. Previously, the BLEU score was 10.61 (39.01 chrF++), but after fine-tuning 15.61 (43.16 chrF++). Russian-Estonian improved from 10.50 BLEU (42.15 chrF++) to 12.71 (44.41 chrF++). These Estonian-Russian and Russian-Estonian results surpass the non-simultaneous baseline, which confirms that the initial model was not strong on these directions because simultaneous model translates using limited context. The cascaded system improved the most on the English-Estonian direction, going from 14.16 BLEU (42.88 chrF++) to 16.43 BLEU (46.20 chrF++).

In addition to improving translation quality, the latency was reduced for all translation directions for the cascaded and end-to-end systems. In the Estonian directions, LAAL for the cascaded system dropped from 3439.77 to 3000.34 and AL from 3130.64 to 2550.43. For the end-to-end system, LAAL dropped from 2969.45 to 2788.09 and AL from 2485.95 to 2255.02. The latency for the end-to-end system is lower compared to the cascaded system on all directions and for every direction, except Russian-Estonian, the translation quality is also better.

Both systems also showed signs of catastrophic forgetting. The models were not fine-tuned on the English-Russian and Russian-English directions, causing the translation quality to decrease.

Despite the decrease in quality, the latency was still improved, showing the benefits of fine-tuning on partial sequences. Interestingly, the translation quality, measured in BLEU, dropped for the cascaded system also on Estonian-English direction, despite the fine-tuning. For the non-fine-tuned system the BLEU was 23.63, but it was reduced to 23.10. However, chrF++ increased from 49.13 to 51.75, which shows that these two metrics do not always agree on translation quality. Similar changes in BLEU and chrF++ can also be observed for the end-to-end system on English-Estonian direction.

6. Discussion

The results showed that using pre-trained translation models for simultaneous Estonian speech-to-text translations is feasible, but this is highly dependent on the translation direction, the selected system, and the latency requirements. It is important to note that in this work both translation systems were tasked with translating between all four Estonian directions. For benchmarking or production purposes, it might be possible to get better results by building and fine-tuning separate systems for each translation direction.

It is possible to roughly estimate the simultaneous translation capabilities of the underlying translation models before starting to build the translation systems. The non-simultaneous baseline provides a good indication of what the performance of the cascaded and end-to-end systems is going to be without fine-tuning. For example, on the non-simultaneous Estonian-English baseline the Seamless M4Tv2 scored 31.28 BLEU (57.08 chrF++). During the simultaneous evaluation, the non-fine-tuned system scored 25.28 BLEU (52.86 chrF++), which is a strong result considering that the model only has access to limited context. A good example of the translation quality limitations is the Russian-Estonian direction, which both models struggled with. For example, on the non-simultaneous Russian-Estonian direction Whisper + NLLB-200 scored 13.64 BLEU (42.61 chrF++) and during the simultaneous evaluation the cascaded system scored 11.45 BLEU (41.33 chrF++). The result is good in terms of being close to the baseline, but the translation quality is still relatively weak compared to the previously discussed Estonian-English direction. The same applies for the Seamless M4Tv2 and non-fine-tuned end-to-end simultaneous translation results. In general, the results showed that in order to translate between Estonian-Russian and Russian-Estonian directions, additional fine-tuning is required, at least for the systems evaluated in this work. The English-Estonian direction results were neither high or low, so fine-tuning necessity depends on the use case.

The simultaneous translation scores after fine-tuning demonstrated that it is possible to noticeably improve the translation quality on the Estonian-Russian and Russian-Estonian with a modest amount of fine-tuning. For the end-to-end system, the Estonian-Russian BLEU score improved from 10.61 (39.01 chrF++) to 15.61 (43.16 chrF++) and the Russian-Estonian BLEU score improved from 10.50 (42.15 chrF++) to 12.71 (44.41 chrF++). The translation quality also improved for the cascaded system, but not as much as the end-to-end system. Some translation directions responded to fine-tuning surprisingly. For example, the end-to-end system had a lower BLEU score on the English-Estonian after fine-tuning, dropping from 17.64 to 17.43.

However, the chrF++ rose from 46.77 to 47.36. This shows that the two metrics did not agree on translation quality. The same can be observed for the cascaded system on the Estonian-English direction. Overall, the average translation quality improved for Estonian directions with respect to BLEU and chrF++ for both systems. When examining fine-tuning results, it should be taken into account that the fine-tuning was done on a synthetic parallel dataset. Fine-tuning using real translation data with the same parameters and time constraints would probably improve the translation quality more.

The benefits of fine-tuning were not strictly limited to improvements in BLEU and chrF++ scores. Fine-tuning the NLLB-200 and Seamless M4Tv2 models on a 1:1 mix of full and partial sequences, which were generated by taking 10-40% prefixes from each sample, reduced LAAL and AL for all translation directions. This includes the English-Russian and Russian-English directions, which were left out of the fine-tuning dataset. It is clear that during the fine-tuning the models adapted to translating with access to limited context. Based on the latency results, fine-tuning is recommended even in cases, when the initial translation quality is good because it allows the models to translate faster with same or better translation quality. The cascaded system benefitted more from the reduction in latency, but the initial LAAL and AL values were higher compared to the end-to-end system. Even after fine-tuning, the end-to-end system still had lower latency values. If very low latency values are required, the wait-k 2 baseline showed that in some directions, such as Estonian-English with Whisper + NLLB-200 and English-Estonian with Seamless M4Tv2, it is possible to translate fast with decent quality. Combining the wait-k 2 baseline with the fine-tuned models could be a good option for these use cases.

In general, the end-to-end system is the better choice of the two. It responded better to fine-tuning in terms of translation quality and after fine-tuning it managed to exceed the cascaded system on all translation directions except Russian-Estonian, where the two systems were almost equal. The end-to-end system scored 12.71 BLEU (44.41 chrF++) compared to 12.99 BLEU (44.05 chrF++) of the cascaded system. In addition to better translation quality, the end-to-end system measured lower on the LAAL and AL metrics on all translation directions, showing that not only the translation quality is better but it can translate faster. The end-to-end system is also easier to implement and fine-tune, making further fine-tuning and implementing new methods more straightforward.

There are some limitations that should be considered when discussing the results. First, the entire training dataset is synthetic. The audio files are real, but the corresponding speech translations

were created from the machine translated transcriptions. In addition, some source datasets had automatically created transcriptions, thus their quality might not be comparable to the transcriptions created by human labelers. Because machine translations are not flawless, they will cause compounding mistakes. Using real parallel data would most likely result in better translation quality, but this type of data is much harder to obtain and it might not even exist in the required quantities. Second, only the NLLB-200 model was fine-tuned for the cascaded system, but fine-tuning Whisper would have helped with transcription quality, especially on the Estonian speech data, and thus improved the final translation quality. However, it is not clear if it would be enough to surpass the end-to-end system based on Seamless M4Tv2 because on two of the four directions Estonian is on the target side. Last, the choice of models was narrowed down by hardware constraints. Using larger and more capable models, such as NLLB-200 3.3B, would probably improve the results further.

The next step to improve the Estonian simultaneous speech-to-text performance, would be to evaluate the AlignAtt policy, which similarly to the LocalAgreement can be applied on the regular translation models (Papi, Turchi, and Negri, 2023). AlignAtt uses the information in the attention weights to guide the simultaneous translation process, and depending on the translation directions, it can offer slightly better BLEU scores at lower latency (Papi, Turchi, and Negri, 2023). Although, in the comparisons, the LocalAgreement was not trained with partial sequences as suggested by Liu, Spanakis, and Niehues (2020), so this may have an effect on the final results.

7. Conclusion

In this thesis, the feasibility of using open pre-trained machine translation models for simultaneous speech-to-text translation for Estonian-English, Estonian-Russian, English-Estonian and Russian-Estonian directions was evaluated. Two different types of translation sytems were compared, cascaded and end-to-end. The cascaded system was created using Voice Activity Detection (VAD), Whisper large-v3 turbo and NLLB-200 distilled 1.3B models. End-to-end system was based on VAD and Seamless M4Tv2 large. The choice of models was limited by hardware constraints. The system must be able to run on Nvidia's V100 32GB or A100 40GB GPU and training must be conducted on A100 80GB GPU. These GPUs are available on UT HPC.

The systems were not allowed to modify previously generated output, thus two strategies were used in order to improve the generation quality and stability. First, LocalAgreement strategy was implemented, which only outputs the matching prefixes from the two consecutive generations. The prefixes were compared on the word basis, as opposed to character based comparisons. In the cascaded system the LocalAgreement was added to the NLLB-200 model and in the end-to-end system to the Seamless model. Second, during each non-final generation step the last generated token was discarded. This prevents the model from outputting punctuation characters too early, although with LocalAgreement the impact is limited and mostly affects the wait-k baseline.

In order to fine-tune the NLLB-200 and Seamless models, a parallel synthetic dataset was created based on the publicly available Estonian, English and Russian speech datasets. The transcriptions were machine translated by the TalTech Laboratory of Language Technology and the TartuNLP group, thus creating a parallel speech-to-text translation datasets. The samples were filtered for duration and partial sequences were created from each remaining sample by taking 10-40% chunks from the beginning of audio and translations. The creation of partial sequences led to an initial dataset of 16 million samples. For each direction, 1133952 samples were selected with 1:1 mix of full and partial sequences. In the end, the finetuning dataset contained 4309K examples totaling 5047.8 hours and the validation dataset, for early stopping, contained 226.8K examples totaling 265.8 hours. Final evaluations were conducted using the test split of the FLEURS benchmarking dataset.

The final results showed that pre-trained translation models can be used for simultaneous Estonian speech-to-text translation without fine-tuning. However, this depends on the performance of

the pre-trained model on the selected translation direction and the translation strategy. The strongest direction was Estonian-English, where the non-finetuned end-to-end system achieved 25.28 BLEU (52.86 chrF++) with 2890.17 LAAL and 2216.07 AL. This is almost equal to the high-resource Russian-English direction, although the latency was higher. In the case of Estonian-Russian and Russian-Estonian, the initial translation quality metrics were poor, and thus it was not possible to achieve relatively good results without fine-tuning. The fine-tuning improved the end-to-end system's performance on the Estonian-Russian direction significantly, going from 10.61 BLEU (39.01 chrF++) to 15.61 (43.16 chrF++) and with lower latency. The Russian-Estonian direction improved as well, but not as much.

In addition to improving the translation quality, which was especially important for Estonian-Russian and Russian-Estonian directions, fine-tuning on a 1:1 mix of full and partial sequences also reduced the translation latency. This also worked on English-Russian and Russian-English directions, even though these directions were not included in the fine-tuning dataset. The latency reduction was more noticeable for the cascaded system, although the latency measurements for it were higher before and after fine-tuning compared to the end-to-end system. On the Estonian-Russian and Russian-Estonian directions, the end-to-end system managed to surpass the non-simultaneous Seamless M4T translation scores, which indicates that the starting point was relatively weak.

In general, when comparing the cascaded and end-to-end systems, the end-to-end system based on Seamless M4T showed more promise. It responded better to fine-tuning on the Estonian-Russian and Russian-Estonian directions, which were relatively weak for both systems in the beginning. Overall, it has a better translation quality and a lower latency. However, it is possible that fine-tuning Whisper in addition to the NLLB-200 distilled 1.3B model would have affected the results. Nevertheless, the gap in translation quality appeared already when comparing the systems without fine-tuning.

References

Ahmad, Ibrahim Said et al. (Aug. 2024a). "FINDINGS OF THE IWSLT 2024 EVALUATION CAMPAIGN". In: *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*. Ed. by Elizabeth Salesky, Marcello Federico, and Marine Carpuat. Bangkok, Thailand (in-person and online): Association for Computational Linguistics, pp. 1–11. DOI: 10.18653/v1/2024.iwslt-1.1. URL: https://aclanthology.org/2024.iwslt-1.1/.

Ahmad, Ibrahim Said et al. (2024b). *Findings of the IWSLT 2024 Evaluation Campaign*. arXiv: 2411.05088 [cs.CL]. URL: https://arxiv.org/abs/2411.05088.

Alumäe, Tanel et al. (May 2023). "Automatic Closed Captioning for Estonian Live Broadcasts". In: *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*. Tórshavn, Faroe Islands: University of Tartu Library, pp. 492–499. URL: https://aclanthology.org/2023.nodalida-1.49.

Ardila, R. et al. (2020). "Common Voice: A Massively-Multilingual Speech Corpus". In: *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pp. 4211–4215.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv: 1409.0473 [cs.CL]. URL: https://arxiv.org/abs/1409.0473.

Cho, Kyunghyun and Masha Esipova (2016). *Can neural machine translation do simultaneous translation?* arXiv: 1606.02012 [cs.CL]. URL: https://arxiv.org/abs/1606.02012.

Chung, Yu-An et al. (2021). *W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training*. arXiv: 2108.06209 [cs.LG]. URL: https://arxiv.org/abs/2108.06209.

Communication, Seamless, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, et al. (2023). *SeamlessM4T: Massively Multilingual & Multimodal Machine Translation*. arXiv: 2308.11596 [cs.CL]. URL: https://arxiv.org/abs/2308.11596.

Communication, Seamless, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, et al. (2023). *Seamless: Multilingual Expressive and Streaming Speech Translation*. arXiv: 2312.05187 [cs.CL]. URL: https://arxiv.org/abs/2312.05187.

Conneau, Alexis et al. (2022). "FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech". In: *arXiv preprint arXiv:2205.12446*. URL: https://arxiv.org/abs/2205.12446.

Fan, Angela et al. (2020). *Beyond English-Centric Multilingual Machine Translation*. arXiv: 2010.11125 [cs.CL]. URL: https://arxiv.org/abs/2010.11125.

Fedorchenko, Artem and Tanel Alumäe (2025). "Optimizing Estonian TV Subtitles with Semi-supervised Learning and LLMs". In: *Proceedings of the 25th Nordic Conference on Computational Linguistics (NoDaLiDa)*.

Gulati, Anmol et al. (2020). *Conformer: Convolution-augmented Transformer for Speech Recognition*. arXiv: 2005.08100 [eess.AS]. URL: https://arxiv.org/abs/2005.08100.

Hernandez, François et al. (2018). "TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation". In: *Speech and Computer*. Springer International Publishing, pp. 198–208. ISBN: 9783319995793. DOI: 10.1007/978-3-319-99579-3_21. URL: http://dx.doi.org/10.1007/978-3-319-99579-3_21.

Ito, Keith and Linda Johnson (2017). *The LJ Speech Dataset*. https://keithito.com/LJ-Speech-Dataset/.

Karpov, Nikolay, Alexander Denisenko, and Fedor Minkin (2021). "Golos: Russian Dataset for Speech Research". In: *arXiv preprint arXiv:2106.10161*.

Koshkin, Roman, Katsuhito Sudoh, and Satoshi Nakamura (Nov. 2024). "TransLLaMa: LLM-based Simultaneous Translation System". In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Ed. by Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen. Miami, Florida, USA: Association for Computational Linguistics, pp. 461–476. DOI: 10.18653/v1/2024.findings-emnlp.27. URL: https://aclanthology.org/2024.findings-emnlp.27/.

Kudugunta, Sneha et al. (2023). *MADLAD-400: A Multilingual And Document-Level Large Audited Dataset*. arXiv: 2309.04662 [cs.CL]. URL: https://arxiv.org/abs/2309.04662.

Liu, Danni, Gerasimos Spanakis, and Jan Niehues (2020). *Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection*. arXiv: 2005.11185 [cs.CL]. URL: https://arxiv.org/abs/2005.11185.

Ma, Mingbo et al. (2019). *STACL: Simultaneous Translation with Implicit Anticipation and Controllable Latency using Prefix-to-Prefix Framework*. arXiv: 1810.08398 [cs.CL]. URL: https://arxiv.org/abs/1810.08398.

Ma, Xutai et al. (2020). *SimulEval: An Evaluation Toolkit for Simultaneous Translation*. arXiv: 2007.16193 [cs.CL]. URL: https://arxiv.org/abs/2007.16193.

Macháček, Dominik, Raj Dabre, and Ondřej Bojar (2023). *Turning Whisper into Real-Time Transcription System*. arXiv: 2307.14743 [cs.CL]. URL: https://arxiv.org/abs/2307.14743.

Panayotov, Vassil et al. (2015). “Librispeech: an ASR corpus based on public domain audio books”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, pp. 5206–5210.

Papi, Sara, Marco Gaido, et al. (2022). “Over-Generation Cannot Be Rewarded: Length-Adaptive Average Lagging for Simultaneous Speech Translation”. In: *Proceedings of the Third Workshop on Automatic Simultaneous Translation*. Association for Computational Linguistics, pp. 12–17. DOI: 10.18653/v1/2022.autosimtrans-1.2. URL: http://dx.doi.org/10.18653/v1/2022.autosimtrans-1.2.

Papi, Sara, Marco Turchi, and Matteo Negri (Aug. 2023). “AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation”. In: *INTERSPEECH 2023*. interspeech$_2$023. ISCA, pp. 3974–3978. DOI: 10.21437/interspeech.2023-170. URL: http://dx.doi.org/10.21437/Interspeech.2023-170.

Papineni, Kishore et al. (July 2002). “Bleu: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040/.

Polák, Peter et al. (May 2022). “CUNI-KIT System for Simultaneous Speech Translation Task at IWSLT 2022”. In: *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Ed. by Elizabeth Salesky, Marcello Federico, and Marta Costa-jussà. Dublin, Ireland (in-person and online): Association for Computational Linguistics, pp. 277–285. DOI: 10.18653/v1/2022.iwslt-1.24. URL: https://aclanthology.org/2022.iwslt-1.24/.

Popović, Maja (Sept. 2015). “chrF: character n-gram F-score for automatic MT evaluation”. In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Ed. by Ondřej Bojar et al. Lisbon, Portugal: Association for Computational Linguistics, pp. 392–395. DOI: 10.18653/v1/W15-3049. URL: https://aclanthology.org/W15-3049/.

– (Sept. 2017). “chrF++: words helping character n-grams”. In: *Proceedings of the Second Conference on Machine Translation*. Ed. by Ondřej Bojar et al. Copenhagen, Denmark: Association for Computational Linguistics, pp. 612–618. DOI: 10.18653/v1/W17-4770. URL: https://aclanthology.org/W17-4770/.

Post, Matt (2018). *A Call for Clarity in Reporting BLEU Scores*. arXiv: 1804.08771 [cs.CL]. URL: https://arxiv.org/abs/1804.08771.

Pratap, Vineel et al. (2023). *Scaling Speech Technology to 1,000+ Languages*. arXiv: 2305.13516 [cs.CL]. URL: https://arxiv.org/abs/2305.13516.

Radford, Alec et al. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv: 2212.04356 [eess.AS]. URL: https://arxiv.org/abs/2212.04356.

Salesky, Elizabeth et al. (2021). "Multilingual TEDx Corpus for Speech Recognition and Translation". In: *Proceedings of Interspeech*.

Sildam, Tiia, Andra Velve, and Tanel Alumäe (Aug. 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)*. Ed. by Atul Kr. Ojha et al. Bangkok, Thailand: Association for Computational Linguistics, pp. 166–174. DOI: 10.18653/v1/2024.loresmt-1.17. URL: https://aclanthology.org/2024.loresmt-1.17/.

Team, NLLB et al. (2022). *No Language Left Behind: Scaling Human-Centered Machine Translation*. arXiv: 2207.04672 [cs.CL]. URL: https://arxiv.org/abs/2207.04672.

Team, Silero (2024). *Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier*. https://github.com/snakers4/silero-vad.

University of Tartu (2018). *UT Rocket*. DOI: 10.23673/PH6N-0144.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

Wang, Changhan, Morgane Riviere, et al. (Aug. 2021). "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 993–1003. URL: https://aclanthology.org/2021.acl-long.80.

Wang, Changhan, Anne Wu, and Juan Pino (2020). *CoVoST 2: A Massively Multilingual Speech-to-Text Translation Corpus*. arXiv: 2007.10310 [cs.CL].

Zhang, Yu et al. (2023). *Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages*. arXiv: 2303.01037 [cs.CL]. URL: https://arxiv.org/abs/2303.01037.

Zheng, Baigong et al. (July 2020). "Simultaneous Translation Policies: From Fixed to Adaptive". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, pp. 2847–2853. DOI: 10.18653/v1/2020.acl-main.254. URL: https://aclanthology.org/2020.acl-main.254/.

Zoph, Barret et al. (Nov. 2016). “Transfer Learning for Low-Resource Neural Machine Translation”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, pp. 1568–1575. DOI: 10.18653/v1/D16-1163. URL: https://aclanthology.org/D16-1163/.

License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Henrik Lepson,
(author's name) ,

1. grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis

   Estonian Simultaneous Speech-to-Text Machine Translation ,
   (title of thesis)

   supervised by Mark Fišel ;
   (supervisor's name)

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;
3. am aware of the fact that the author retains the rights specified in points 1 and 2;
4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Henrik Lepson
11/08/2025