University of Amsterdam

Master Thesis Forensic Science

Optimizing forensic audio transcription: A domain-specific solution using Whisper and Large Language Models.

April - November 2024

Supervisor:
Floris Gisolf MSc, OVV

Student:
Christiaan van Luik
14622459

Examiner:
prof. dr. ing. Zeno Geradts, UvA/NFI

Date of submission:
November 18, 2024

Institute:
Onderzoeksraad voor Veiligheid

Course:
Research Project (36EC)

Abstract

This research investigates the development of an automated and complete audio transcription solution tailored for forensic applications, using OpenAI's Whisper model and various enhancements. The study focuses on enabling the Dutch Safety Board (DSB) to efficiently process sensitive audio data, such as cockpit recordings and interviews, by implementing locally deployed, domain-specific automatic speech recognition (ASR). Enhancements were made to improve Whisper's accuracy on Dutch language audio, address challenges like noise and multi-language handling, and incorporate a large language model (LLM) for transcript correction and insight generation.

I fine-tuned Whisper using both synthetic Text-to-Speech (TTS) and DSB interview datasets, achieving a notable improvement in Word Error Rate (WER). Additional modifications included using Voice Activity Detection (VAD), speaker diarization, and phoneme alignment with WhisperX to enhance timestamp accuracy and speech clarity. Furthermore, vocal separation techniques were tested to mitigate background noise in recordings.

For analysis, local LLMs were integrated to correct transcription errors and generate insights, streamlining the investigation process by summarizing key information. This pipeline demonstrates a significant improvement in transcription quality, particularly in forensic contexts where accuracy is important. Future work includes refining language model tuning and expanding the dataset to cover more audio.

Most code written in this study is available on GitHub at https://github.com/cvl01/forensic-audio-transcription.

Keywords

Whisper, Dutch audio recognition, transcript correction, multilingual audio recognition, large language models

Contents

1 Introduction

The field of Automatic Speech Recognition (ASR) has undergone a significant transformation since the launch of the Whisper model [1]. Whisper is a large-scale model trained on weakly supervised datasets, which has yielded exceptional performance that surpasses other models. Unlike traditional approaches, Whisper does not require extensive preprocessing or standardization steps, simplifying the pipeline and achieving robust results across various audio types.

The Dutch Safety Board (DSB) (*Dutch: Onderzoeksraad voor Veiligheid (OVV)*) both collects and generates vast amounts of audio data. In its investigations, audio is regularly retrieved from data recorders after an incident happened, to conduct its safety investigations. Investigators listen to these recordings to reconstruct the events prior to and at the time of the accident. Furthermore, the DSB's investigations lean heavily on interviews that are conducted with parties involved in an investigated accident situation. These interviews are recorded and manually transcribed to be used in the investigation. Listening to and/or manually transcribing these audio files takes a lot of effort, manpower and time, which is why effective integration of automated speech recognition is of great interest. The DSB is not able to make use of cloud-based ASR services, since the data that is being worked with is sensitive and confidential. As a consequence, any ASR model employed should be able to run locally, which is why Whisper is suited well for usage by the DSB.

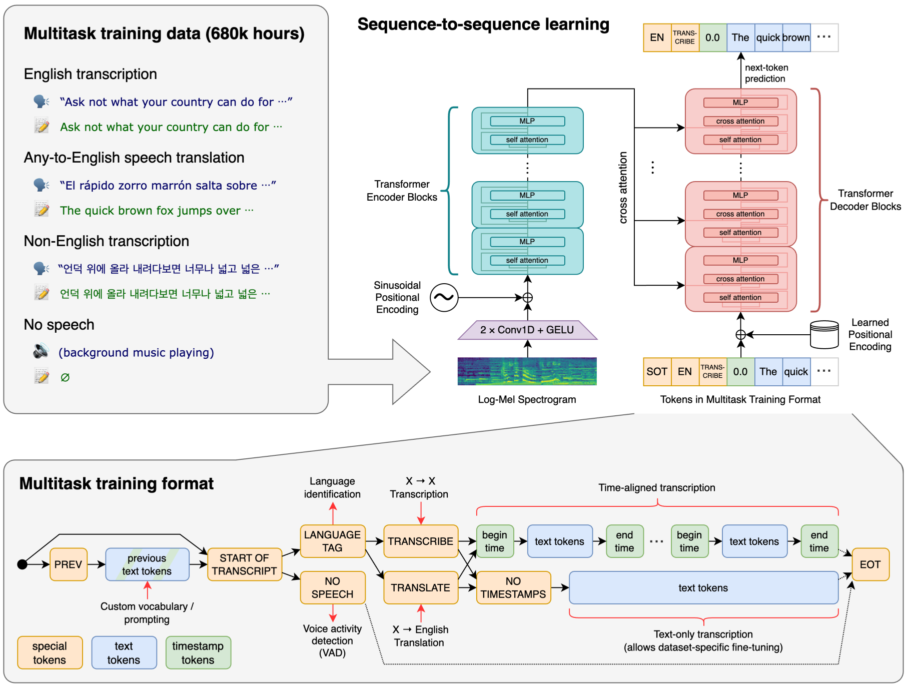

Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ⋯"
Ask not what your country can do for ⋯
Any-to-English speech translation
"El rápido zorro marrón salta sobre ⋯"
The quick brown fox jumps over ⋯
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
⋯
next-token prediction
MLP
cross attention
self attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
⋯
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
⋯
begin time
text tokens
end time
NO TIMESTAMPS
text tokens
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Figure 1: Pipeline of the Whisper model [1]

Notably, Whisper works well on many languages, with remarkable accuracy. Without further fine-tuning, the model demonstrates impressive adaptability, performing well on diverse audio data from different domains. This demonstrates the benefits of training on large-scale datasets and highlights the potential for applications implementing Whisper. The model, of which the pipeline is depicted in figure 1, is available open source and can be run on-premise, which is an important aspect when dealing with sensitive (forensic) data.

As research on and usage of Whisper continues to evolve, numerous re-implementation and extension proposals have emerged. One notable example is whisper.cpp [2], a re-implementation of Whisper in c++. It offers higher performance by utilizing hardware and framework related optimizations, and by offering a lean C++ implementation with minimal dependencies. faster-whisper [3] is another reimplementation, in Python, that uses the CTranslate2 library, which is a faster inference engine for transformer models. This implementation is up to 4x faster than the default OpenAI's Whisper implementation.

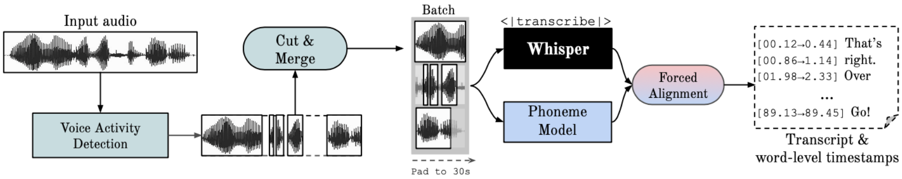


Figure 2: Pipeline of WhisperX [4]

Researchers from the University of Oxford presented WhisperX [4], a more extensive system that uses Whisper, combined with Voice Activity Detection (VAD) and phoneme matching, in order to create a timestamp-accurate speech recognition system. A schematic of the pipeline can be found in figure 2. WhisperX uses VAD preprocessing to cut audio into chunks, and use these to enable batched processing. They have also shown that their VAD cut & merge reduces hallucination and repetition, which is known to be a problem of the original Whisper model [1]. WhisperX force aligns the transcribed word segments using a phoneme model, resulting in word-level time-accurate transcriptions. The model offers 70x real-time transcription speed, as faster-whisper has been implemented in the latest version (v3) [5]. This version also contained the addition of speaker diarization using pyannote-audio.

In the Dutch context, research has been conducted on improving Whisper through Voice Activity Detection [6]. Mul [7] focused on enhancing the transcriptions by applying speaker diarization, and compared PyAnnote and NeMo speaker diarization tools.

Initial results of research efforts fine-tuning Whisper on specific languages or domain jargon shows interesting results, achieving Word Error Rate (WER) reductions with small amount of training audio (10–50 hours) [8], [9]. Others have studied the possibility of using Text-to-Speech (TTS) systems to generate synthetic training data, to train the ASR system on domain-specific vocabulary [10]–[12].

Researchers at Google have conducted research on improving ASR transcripts by utilizing a LLM [13]. They have shown that using an LLM to post-process a transcript can reduce the Word Diarization Error Rate (WDER). The LLM is used to detect and move misplaced words in the ASR transcript, so that they are attributed to the correct speaker. Their fine-tuned model shows around 50% reduction in WDER on two evaluated datasets.

LLM's have also been employed to enhance speaker diarization [14] in NVIDIA's NeMo by adding lexical information obtained from an LLM in the inference stage. The LLM offers complementary information and captures contextual cues otherwise missing, which leads to a reduction in error rates.

Research on noise-reduction preprocessing [15] shows that noise reduction has a positive impact on small models, but for larger models the research found no considerable effect on the transcription quality.

The research proposed will build further on these developments. In my research, I will seek to validate, improve and combine multiple of the technologies and developments detailed above. The final goal will be to combine different techniques in a comprehensive audio transcription system capable of transcribing different types of audio with the latest models.

The foundation will be based on OpenAI's Whisper model [1] and the re-implementation and extension of Whisper by researchers from Oxford, called WhisperX. WhisperX is combining the Whisper transcriptions with Voice Activity Detection, to improve transcription quality and enable batched processing, combined with more accurate timestamps on word level by use of phoneme matching using wav2vec2 alignment. Finally, WhisperX also adds speaker diarization using pyannote-audio [16]. This research aims to validate the WhisperX performance on case-related audio at the DSB, especially the effect of voice activity detection, both on interview-audio, but also on cockpit voice recorder (CVR) audio, which contains longer periods of silence, low-volume audio, and background noise.

The research will further seek to refine and fine-tune Whisper for better performance on Dutch audio, and investigate possibilities of domain-specific fine-tuning. Additionally, the research will look into segmenting multi-language audio into language-segmented audio files, to improve performance on models that require defining the audio language. The current Whisper model does not handle language switches within audio well, and also phoneme matching heavily depends on a preset language. It is thus expected that language-segmented audio files will provide better results than multi-language audio, when transcribed with Whisper.

As different types of CVR and Voyage Data Recorder (VDR) audio can contain a lot of (static) background noise, the effect of noise reduction techniques on noisy audio recordings, to enhance transcription accuracy, will be tested and evaluated. Although Whisper is trained to work with noisy audio, my hypotheses would be that removing heavy static noises, such as engine noises, will improve the model's performance.

Finally, the research will be investigating the effectiveness of using and integrating local Large Language Models (LLMs) for improvement, summarization and analysis of the transcriptions. Types of improvements of the transcriptions by using an LLM could be correcting spelling or word mistakes, but could also be correcting incorrect speaker assignments, like was done by Wang et al. [13]. The sensitivity of the data used at the DSB requires the use of local LLM's, requiring investigation which local LLM's are capable of doing these kinds of tasks.

To make the research results quantifiable, evaluating the quality of transcriptions generated by the system, is a key aspect. The goal is to create a robust, domain-specific audio transcription tool capable of not only transcribing interviews, but also noisy, accented and varying speech encountered in recordings from the aviation and shipping industries.

2 Material & Methods

This section first gives an overview of key tools and technologies used in this research, before delving deeper in the methodology and steps undertaken to perform this research. Different libraries were implemented or adapted, mostly in Python, and most code used for training or inference is available at https://github.com/cvl01/forensic-audio-transcription.

2.1 Tools

2.1.1 Whisper

Whisper [1] is an automatic speech recognition (ASR) model developed by OpenAI, known for its robust performance on multilingual and varied audio inputs. Utilizing a transformer-based architecture, it used large-scale datasets to improve transcription accuracy, even in challenging acoustic conditions. For this study, Whisper serves as the only and primary ASR tool for transcription, on which this research builds further.

2.1.2 WhisperX

OpenAI's Whisper implementation suffers from a problem called hallucinations, where a piece of text is repeated frequently. Although this problem can turn up randomly, it is known to occur more frequently during long amounts of silence or difficult or noisy audio. WhisperX [5], by applying voice activity detection and filtering out silences, together with the batched approach of splitting the audio in 30-second chunks, seems to take away this problem. Hallucinations were not observed in a large amount of training and case data transcribed by the WhisperX pipeline.

2.1.3 PEFT

Parameter-Efficient Fine-Tuning (PEFT) methods are fine-tuning methods that only train a small portion of a large pre-trained model's parameters, in order to save on storage and computation power. Due to the size of the Whisper large-v2 model, PEFT tuning was applied. Two different PEFT methods were applied: LoRa [17] and AdaLoRa [18].

2.1.4 LoRa

Low-Rank Adaptation (LoRa) [17] freezes the pretrained model weights and injects trainable rank decomposition matrices into each layer of the Transformers architecture, in this case the Whisper Encoder and Decoder. This leads to a large reduction in the number of training parameters, leading to lower memory requirements and faster training. It only trains a fraction of the parameters, around 1% and shows similar or improved performance compared to full training.

2.1.5 AdaLoRa

AdaLoRa [18] builds upon the principles of LoRa, but instead of evenly distributing parameters over the pre-trained model, introduces a dynamic mechanism to allocate parameter budgets more adaptively across different layers and weight matrices of the model. This optimizes the usage and training of more influential weight matrices. The method maintains the benefits of reduced computational and memory cost while showing improved performance over full training.

2.1.6 xTTS

Generation of realistic and natural-sounding speech audio was done using the Coqui XTTS-v2 TTS model. The Coqui XTTS-v2 model is a state-of-the-art massively multilingual zero-shot text-to-speech system [19]. XTTS builds upon the Tortoise model with several modifications to enable multilingual training, improve voice cloning capabilities, and allow for faster training and inference.

As part of its dataset, XTTS incorporates 74.1 hours of Dutch speech data. While this is less than some of the more resource-rich languages in the dataset (such as English or German), it still provides a reliable foundation for Dutch speech synthesis. Performance evaluation for Dutch results a CER of 0.946

and a SECS of 0.4825. These scores indicate strong performance in both pronunciation accuracy and speaker similarity for Dutch synthesis.

In this research, the pre-trained XTTS-v2 model was used, to generate high-quality, natural-sounding speech in Dutch with multiple speaker voices with different characteristics.

2.1.7 Large Language Models

Various open-access large language models (LLMs) have been tested and evaluated. Open-access was an important requirement, as due to the confidential nature of data, only on-premise LLMs could be used. Due to hardware restrictions, the biggest models used were 70/72B models, which fit into the available 48GB VRAM when quantized. Llama 3.1 70B Instruct and Qwen 2.5 72B Instruct were used, quantized to a Q4 gguf [20].

2.2 Whisper training

2.2.1 TTS Dataset

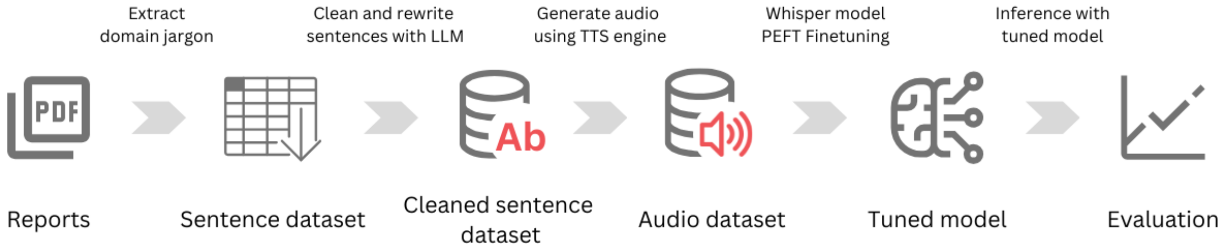


Figure 3: Methodology to fine tune Whisper on domain data using TTS.

Inspired by Vasquez-Correa et al.'s work [10], I trained the Whisper model using Text-to-Speech (TTS) data, leveraging reports and documents published by the Dutch Safety Board (DSB) over the past 25 years. The methodology is depicted in figure 3. The process started with a comprehensive word list that was compiled by merging all words used in these documents, followed by the selection of words that appeared between 3 and 1000 times in the documents. These words were then fed into a TTS engine to generate audio files for each word, utilizing five different speaker voices, with varying accent and tone. The TTS-engine used for this is Coqui XTTSv2 [1].

In the first training attempts, it became obvious that the Dutch documents and reports still contained quite some English terminology. This negatively affected the quality of the model, as similarly-sounding words would be transcribed with English spelling. For example, the Dutch word 'dok' would be incorrectly transcribed to the English spelling 'dock'.

Thus, the word list was adapted by determining the language of each individual word and only keeping those that were classified as Dutch. Classification was done with the lingua-rs library [21], by computing the probability the word was either Dutch or English.

My initial attempt involved training the Whisper model on one-word audio files, resulting in a large dataset (25 GB). However, this grew even larger during training due to the Whisper model's requirement for 30-second audio files. Each one-word audio file is thus padded with approximately 28 seconds of silence by Whisper model to construct a 30-second audio file or log-mel spectrogram.

To limit the size of the dataset, I generated longer audio files containing multiple words separated by spaces as sentences, to reduce the amount of padded silence. However, feeding this dataset to the TTS engine, the generated TTS audio became unnatural, likely because of dealing with artificial sentences comprising words that do not naturally follow each other.

Then, I tried concatenating the one-word audio files with a 300 ms pause between each word until a 30-second audio file was constructed. This approach enabled me to generate a dataset consisting solely

[1] https://huggingface.co/coqui/XTTS-v2

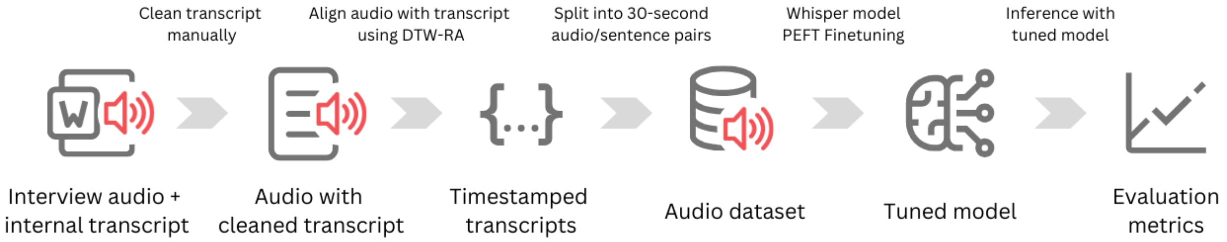

Clean transcript manually
Align audio with transcript using DTW-RA
Split into 30-second audio/sentence pairs
Whisper model PEFT Finetuning
Inference with tuned model
W
{...}
Interview audio + internal transcript
Audio with cleaned transcript
Timestamped transcripts
Audio dataset
Tuned model
Evaluation metrics

Figure 4: Methodology to fine tune Whisper on domain data using the interview-dataset.

of approximately 30-second audio files. The resulting audio dataset was then used to train the Whisper large-v2 model. The tuned model, however, performed much worse than the base model. Interpunction was largely gone in the transcription, and many words were incorrectly transcribed. It became clear that training on this large dataset of sentences consisting out of randomly concatenated single words, not in the context of a sentence, leads to output in the same form as the training data, i.e. sequences of words without interpunction, and not natural, correct sentences as the whisper base model provides. Also adapting the dataset to add comma's between each and every word, to more naturally resemble the textual representation of reading out a list of words, did not return a usable tuned model.

To overcome this, I aimed to construct a dataset with natural, realistic sentences. For each word in the aforementioned word list, I selected a sentence from the reports. These sentences were used as the basis for the dataset. Due to the extraction of sentences from the report, some sentences contained incorrect characters and needed cleaning. Other sentences were of a structure that, although making sense in written text, would not be expressed like that in speech. I prompted an LLM with each of the sentences in the dataset and prompted it to clean and rewrite the sentence to spoken language if necessary. For this, I used Google's gemma2:27b public model. The resulting dataset was used for generating audio using TTS. Words occurring between 2 and 1000 times were selected, of which three datasets were created, of 2000, 4000 and 8000 records.

This made the final TTS dataset that was used for training Whisper's large-v2 and large-v3 models, using LoRa and AdaLoRa. The resulting tuned models were evaluated on a test dataset constructed from internal audio data. This dataset was improved throughout the research. Final performance metrics on the TTS dataset were obtained by using the constructed interview dataset (see section 2.2.2).

2.2.2 Interview-based dataset

Within the DSB, interviews are recorded and transcribed verbatim. Initially, I deemed this audio with transcripts unsuitable for training Whisper. This was primarily due to the fact that the transcripts do not always match the audio verbatim, and some portions of the audio are not reflected in the transcripts. Additionally, the transcripts lack timestamps, whereas training Whisper requires precise audio segments with corresponding text of no more than 30 seconds, the size of Whisper's context window.

To utilize this material for training Whisper, the following method was developed, which is depicted in figure 4. First, I manually converted the transcribed interview texts into a usable text file containing only the spoken text. Then, I employed forced alignment techniques.

Specifically, I used Dynamic Time Warping with Recognition Assist (DTW-RA), a method implemented in the echogarden library [22]. This technique involves running an ASR model on the audio, in this case Whisper large-v2. Subsequently, both the ground-truth transcript and the recognized (ASR) transcript are synthesized via eSpeak. The optimal alignment between the two synthesized waveforms is determined using the DTW algorithm, and this alignment is then remapped onto the original audio based on the timing information generated by the recognizer. The aligner outputs estimated timestamps for each sentence and word within the audio. This method turned out to be much more stable than plain DTW, and was able to deal with difficult audio better.

Other methods apart from DTW-RA were also tested: plain DTW, plain Whisper, and guided encoding using Whisper. In the plain Whisper method, I tried to make use of the timestamp Whisper generates, but these turned out to be unusable as they are not precise enough.

Common techniques for forced alignment include architectures based on Gaussian Mixture Models and/or Hidden Markov Models, such as the Montreal Forced Aligner (MFA) [23]. These could not be used since these models do not work well with long audio. Specifically, MFA can only align audio of maximum 30 seconds.

Using DTW-RA method, I aligned the audio with the transcript. The aligner provides each sentence and word with an estimated timestamp in the audio. The resulting transcripts with timestamps were subsequently used to create audio fragments, each no longer than 30 seconds, with known spoken text. These fragments were created by adding full sentences to the fragment until no more could be added without going over the 30-second limit. This results in audio fragments of lengths between 20 and 30 seconds, with no spoken sentences being cut up and split between files.

Two issues remain unresolved. First, alignment is quite challenging, and the predicted timestamps, especially for more difficult audio recordings, are often inaccurate. Furthermore, as previously mentioned, the transcripts created by the DSB do not always capture all spoken content, and some segments are omitted. This also causes the aligner to occasionally become confused. As a result, not all fragments match the predicted text.

Therefore, I added an extra step that involves passing the short audio fragments through Whisper again. The hypothesis generated by Whisper is then compared to the reference text of the fragment. If the difference in the number of words exceeds 25% or the Word Error Rate (WER) is greater than 50%, the audio fragment is discarded. In practice, for some interviews only 2-4% of the fragments are discarded, but for some others this can get up to 40%. Because results were not satisfactory, boundaries were reduced to a maximum 20% word count difference and maximum 30% WER.

This then gave a dataset of audio-sentence pairs suited for fine-tuning Whisper. This was done using LoRa and AdaLoRa, on Whisper's large-v2 and large-v3 models. Also, regular, non-PEFT tuning was performed to compare accuracy with the (Ada)LoRa tunes.

2.2.3 Mixing datasets

Having these two datasets, TTS and interview, and respective models resulting from training on those, I now proceeded to combine these datasets. The training methodology involved combining multiple audio datasets, specifically the Interview dataset, TTS dataset, and a subset of Common Voice 17 for Dutch. Common Voice was added to diversify the type of audio in the mix. Rather than processing these three datasets sequentially, I implemented an interleaved approach where samples from each source were shuffled and mixed during the training process. This methodology ensures that each training batch contains a diverse representation of speech patterns, acoustic conditions, and speaker characteristics from across all datasets. This approach proved beneficial, as it resulted in improvements in the quality of the trained model. This approach is a step towards mitigating the risk of overfitting to any single dataset's particular characteristics.

2.2.4 Dutch text normalizer

OpenAI's Whisper implementation bundles a normalizer for English text[2]. The normalizer can be run on both reference and predicted text before computing the WER. The normalizer takes care of 'normalizing' the transcript. This includes normalizing contractions, numbers and monetary values, spelling if multiple valid options exist, time and date, etc. To effectively normalize Dutch, I created a Dutch text normalizer based on the English variant. This way, the Dutch text could be sensibly normalizer and a more valuable normalized WER could be computed.

2.2.5 Multi-language audio

I adapted WhisperX for use with multi-language audio. Some audio recordings the DSB deals with have a high probability of containing multiple languages spoken. For example, in audio from cockpit voice recorders, speakers might switch language multiple times within a five-minute segment. Current Whisper implementations could not handle this well. I adapted and extended the WhisperX implementation

[2] https://github.com/huggingface/transformers/blob/main/src/transformers/models/whisper/english_normalizer.py

to have an option to detect the audio language for every 30-second chunk. A predefined list of possible languages can be set to improve detection accuracy. This algorithm change drastically improves the transcription of multi-language audio recordings.

I modified the VAD separation algorithm in WhisperX to include a `min_duration_off` parameter, which prevents splitting segments during brief pauses within sentences. Initially, the algorithm runs with the specified `min_duration_off` value. If any segments exceed 30 seconds, it reruns the algorithm with `min_duration_off` set to 0, ensuring optimal segmentation.

2.3 Voice separation

In an effort to increase the transcription quality of noisy audio, I evaluated various vocal separation models, to be used before running Whisper on the audio. Although these vocal separation models are primarily designed for music isolation (e.g. karaoke applications), they proved to be effective in separating voice from noise and background sounds in different audio types too, like for aviation cockpit recordings. Different models and model architectures were used. Several different models and architectures were tested; however, creating a standardized baseline for comparison was challenging due to the nature of the audio files at hand, being aviation audio. The main challenge is the significant variation in the transcription of aviation call signs. In aviation, call signs are typically communicated using the NATO phonetic alphabet to ensure clarity. For instance, the call sign "N123AB" might be transmitted as "November One Two Three Alpha Bravo." However, transcription models often handle these call signs inconsistently. Some transcriptions capture only the letters and numbers ("N123AB"), while others transcribe the entire phonetic words ("November One Two Three Alpha Bravo").

The evaluation methodology involved pre-processing an audio file with each of the vocal models before applying Whisper on the audio. This was done for multiple different recordings and different vocal models. I also added a post-processing step to the vocal separation to include amplitude normalization across all separated audio streams to ensure consistent volume levels. The resulting transcripts were compared to get an impression of the effect of the vocal separation step and judge its effectiveness.

2.4 LLM Analysis

Large language models (LLM's) have recently emerged as versatile tools. In this research, various applications of LLM's in the context of dealing with audio transcriptions have been explored.

For being able to run LLM's at acceptable speed, the amount of GPU VRAM available is an important factor. In this project, two Nvidia RTX GPU cards (3090 & 4090) were available, giving a total capacity of 48GB VRAM.

To be able to run larger models, we make use of quantization. Quantization is a technique to reduce memory requirements by loading model weights in lower precision. In this research, the GGUF format was used, in combination with the llama.cpp library. GGUF is optimized for fast loading and saving of models, and supports multiple quantization formats [20].

Various models have been tested and evaluated by hand. Although various benchmarks exist, ranking LLM's, these benchmarks do not give enough information on the performance on specifically our tasks. A major factor at play is the model's ability to work with Dutch text, both in understanding and generation. Benchmarks for Dutch specifically do exist [24], but at the time of writing do not have benchmarks of the newer, more recent models. Another assesment factors are: ability to cover all details of long context (32K), reasoning soundness, and hallucination if context is missing. So, finding the best model for our use case isn't as simple as choosing the top model on a benchmark leaderboard that fits our VRAM requirements, and thus manual testing to get a grasp of the model's performance remains necessary.

The project's goal in employing LLM's is two-fold: correction of mistakes in transcripts to improve its quality and analysis of the transcript by generating different types of insights based on the transcript.

Various models have been tested and evaluated for this purpose. In practice, the model's ability to work well with Dutch text was a major deciding factor. Also, variation exists between the creativeness of the models and its ability to follow the instructions carefully, which is important for transcript correction. In this research, Meta-Llama-3.1-70B-Instruct [25] quantized at Q4_K_M and IQ4_XS [3], and Qwen2.5-72B-Instruct [26], [27] quantized at IQ4_XS [4] were used.

The models are run for inference via llama-cpp-python. All 81 layers of the model fit in GPU memory of the 3090 and 4090 combined (24GB each, 48GB total), leaving a maximum context size of 16K and 32K tokens, when using respectively Q4_K_M and IQ4_XS. Exceeding this context size requires offloading layers to the CPU, which results in a notable performance decrease.

2.4.1 Transcript correction

Despite improvements in the Whisper model, many transcriptions still contain significant errors, with Word Error Rates (WER) around 20%. Consequently, these transcriptions often require human correction to be suitable for practical use. However, in numerous cases, listening to the original audio may not be necessary, as the correct words can often be inferred from the context or even derived solely from the mistranscribed word, for example in case of misspelled names. This study explores the use of a Large Language Model (LLM) to automate this correction process.

An initial problem that arises is that transcripts are generally quite long, and in many cases don't fit the maximum available context window, given hardware requirements. So, the texts have to be split in order to be processed by the LLM. However, when splitting the text in smaller batches, the context is partially lost, while some words or facts from a first batch might be essential for correcting a later batch. Another reason for the need of splitting the text into smaller chunks is, that in practice, LLMs tend to be less accurate in instruction following as the prompt length grows. As a solution to mitigate the effects of losing context, a short summary of the whole transcript is generated with a 'lighter' model, currently Llama 3.1 8B is used for this. This summary is used to provide context in the batched transcript correction process. Also, if a transcript is linked to a project, a small project summary is available. This project summary is humanly generated and contains names of involved parties, important dates and locations.

The transcript is split into batches of plus minus 4000 characters, with a batching method that respects paragraphs, and does not split mid-paragraph. The correction prompt, together with the interview summary and project summary, are given as the system prompt, and the transcript as the first user message. Prompts can be found in appendix A. The short summary both serves to provide context of the kind of text the LLM will be dealing with, but secondly also contains the correct spelling of important names mentioned in the transcript. One of the common errors of long Whisper transcripts is that the names of persons, businesses or geographical locations are not always transcribed consistently. The LLM correction step helps to get an aligned, consistent naming within the transcript.

2.4.2 Generating insights

Insights are generated by asking questions to the LLM and adding the transcript as context. The question is inserted in a base prompt, which contains some general instruction as well as the project description. Prompts can be found in appendix A. As the project description contains names of involved parties and people, this helps the LLM to generate answers with the right context and role of the people involved. In a sense, it is a kind of RAG solution, where the context is already known (the interview transcript) and questions are being asked based on this context.

[3] https://huggingface.co/bartowski/Meta-Llama-3.1-70B-Instruct-GGUF

[4] https://huggingface.co/bartowski/Qwen2.5-72B-Instruct-GGUF

3 Results

This chapter contains the results of the various steps described in chapter 2.

3.1 Whisper

3.1.1 TTS Dataset

To explore domain-specific tuning in Dutch, a dataset generated from domain-specific TTS data has been used to train large-v2 and large-v2 models, using LoRa and AdaLoRa. Results are displayed in table 1. Three different datasets were used, named after the number of sentences in it. Datasets with (approximately) 2000, 4000 and 8000 items were evaluated.

None of the tuned models performs significantly better than the base model. In fact, many tuned models performed worse than the base model. There is a notable difference between the models with a frozen encoder and those with a tuned encoder. The latter perform worse than the former, which is expected behavior. The TTS generated audio does not fully match with human-spoken audio, and training the encoder with this type of data will negatively affect its performance when evaluated on human speech.

While WER scores provide a measure of transcription accuracy, they fail to capture the nuanced correctness and value of a transcription. Each incorrectly transcribed word is penalized equally, while some words in fact might be more important to be transcribed correctly than others. In general, my feeling was that the produced transcriptions by the TTS tuned models did indeed recognize domain jargon much better, but this came accompanied by an overall worse transcription, with the model leaving out parts of sentences or transcribing words incorrectly. Also, punctuation seemed to worsen, with it missing out or incorrectly being added in unexpected locations. The model was trained with sentences with proper interpunction.

Tuned on	Model	Technique	FE	WER	CER	nWER	nCER
untuned	large-v2			33.6	18.85	23.6	16.45
untuned	large-v3			33.16	16.97	22.1	14.3
TTS dataset 2000	large-v3	adalora	fe	129.76	86.65	132.01	79.74
TTS dataset 2000	large-v3	lora	fe	118.96	85.47	106.1	79.71
TTS Dataset 4000 v2	large-v2	adalora	fe	39.51	25.95	30.56	23.69
TTS Dataset 4000 v2	large-v2	adalora	te	46.92	35.15	39.71	33.32
TTS Dataset 4000 v2	large-v2	lora	fe	36.53	20.03	25.13	16.53
TTS Dataset 4000 v2	large-v2	lora	te	**35.29**	**19.47**	**24.96**	**16.35**
TTS Dataset 4000 v2	large-v3	adalora	fe	**33.16**	**17.01**	**22.12**	**14.33**
TTS Dataset 4000 v2	large-v3	adalora	te	115.03	74.06	106.17	71.01
TTS Dataset 4000 v2	large-v3	lora	fe	38.19	22.36	28.11	19.9
TTS Dataset 4000 v2	large-v3	lora	te	41.1	23.33	30.63	20.46
TTS Dataset 8000	large-v2	lora	fe	37.84	20.8	24.37	16.58
TTS Dataset 8000	large-v2	lora	fe	47.21	28.72	37.7	26
TTS Dataset 8000	large-v3	lora	fe	150.48	109.09	143.01	106.07

Table 1: Performance of different models tuned with TTS datasets, measured by (normalized) WER and CER. *FE = Frozen Encoder, TE = trained encoder.*

Model	Technique	FE	WER	CER	nWER	nCER
large-v2 untuned model			33.6	18.9	23.6	16.5
large-v2	adalora	fe	28.5	15.1	19.6	13.1
large-v2	adalora	te	**27.8**	**14.8**	**19.2**	**12.9**
large-v2	lora	fe	28.0	14.9	19.6	13.1
large-v2	lora	te	28.4	15.2	19.8	13.4
large-v2	regular	te	29.4	15.7	20.2	13.6
large-v3 untuned model			33.2	17.0	22.1	14.3
large-v3	adalora	fe	27.1	14.3	**18.0**	12.4
large-v3	adalora	te	27.8	14.8	18.6	12.8
large-v3	lora	fe	26.8	**13.9**	18.2	**12.1**
large-v3	lora	te	**26.7**	14.0	18.2	12.2

Table 2: Performance of different models tuned with the interview dataset, measured by (normalized) WER and CER. *FE = Frozen Encoder, TE = trained encoder.*

3.1.2 Interview dataset

The dataset produced from fragments of interviews and their transcripts has also been used to train the Whisper large models in the same way, using again LoRa and AdaLoRa and tuning with and without freezing the Whisper encoder. Results are in table 2. During testing, it became obvious that the quality of the dataset was not high enough. That is why a new dataset was created (v3) that was filtered to have a maximum of 30% in WER and 20% word count difference. This dataset gave consistent improved results over the base model when used for training.

The results show the best WER can be obtained when using OpenAI's large-v3 model as a base. This is in line with the slightly better performance of large-v3 over large-v2 before training. The best tuned model shows a 20% decrease in WER over the untuned model. The differences between training the encoder or freezing the encoder are minor on this dataset. The default Whisper encoder is well-trained on encoding human speech, so the larger performance gains can be achieved by enriching the decoder's vocabulary. For large-v2, AdaLoRa performs slightly better than the equivalent LoRa model. For large-v2, the LoRa model is slightly better when using the WER as metric, whereas the AdaLoRa-tuned model is slightly better on the normalized WER metric.

Using interview audio for training increases domain-specific recognition capabilities. For example, maritime terminology that was rarely to never recognized correctly by the large-v3 model, is better recognized in the fine-tuned model.

3.1.3 Common Voice

Since Whisper's training data doesn't contain the Common Voice dataset, I fine-tuned Whisper with the Dutch subset of Common Voice 18 using the existing Speechbrain [28] Whisper fine-tuning script. The Dutch subset is quite large, but training saw only a small improvement on WER when evaluated on the test split of the dataset, which is why no further research efforts were made.

3.1.4 Mixed dataset

A mixed dataset was created by interleaving samples from different datasets, like described in 2.2.3. This dataset contains samples from the interview, TTS and Common Voice datasets in the proportion of respectively 10%, 35% and 55%. Whisper large-v3 was finetuned using this dataset, with LoRa and frozen encoder for three epochs. Results can be found in table 3. The model fine-tuned on the large-v3-big-dataset for two epochs demonstrated the best performance across all metrics. Specifically, it achieved

	mixed-dataset test split				interview-dataset test split			
Epochs	**WER**	**CER**	**nWER**	**nCER**	**WER**	**CER**	**nWER**	**nCER**
large-v3 untuned model	22.89	10.95	15.61	9.17	33.2	17.0	22.1	14.3
1	19.95	9.45	13.45	8.03	28.43	14.53	18.56	12.4
2	**18.72**	**9.29**	**13.2**	**8.08**	**26.48**	**14.2**	**18.05**	**12.33**
3	19.09	9.53	13.61	8.37	26.93	14.65	18.61	12.84

Table 3: Performance of different checkpoints for large-v3 fine-tuned with the combined, mixed-dataset, measured by (normalized) WER and CER on both mixed-dataset and interview-dataset test splits.

the lowest Word Error Rate (WER) at 18.72% and normalized WER at 13.2%, outperforming other checkpoints with one and three epochs, and the untuned OpenAI Whisper large-v3 baseline model, which reported a WER of 22.89% and normalized WER of 15.61%.

This performance indicates that fine-tuning on a domain-specific dataset can significantly enhance transcription accuracy, particularly in reducing errors in both raw and normalized formats. The improvements in Character Error Rate (CER) and normalized CER further highlight the effectiveness of the tuning process in producing a more precise and reliable model for ASR in this domain.

While the interview dataset did show improvement in WER scores and comparison of the transcripts before and after fine-tuning also shows that the model did learn domain-specific knowledge, the model tuned on the interview dataset showed decreasing performance on some other sentence parts. That is why mixing and interleaving datasets was needed, and the resulting model, next to having lower WER metrics, also seems to be more stable and less biased to a specific type of terminology or training data. The model also successfully incorporates TTS audio, and thus learned new terminology, without decreasing overall model quality.

3.1.5 Multilanguage audio

The modified WhisperX package detailed in section 2.2.5 generates way more accurate transcripts for multilanguage audio. Since language detection is repeated for every segment, the language accuracy is very high. In figure 5 a short excerpt from Air Traffic Control (ATC) audio of Budapest airport in Hungary is transcribed. Of the 1-hour-long ATC audio recording, approximately 90% of the speech is English, but in some cases also Hungarian is spoken. If you transcribe using the original whisper implementation, the language is detected to be English for the full audio file, and you get the transcription on the right side of figure 5. When using the improved algorithm, you get the transcription on the left. Note that the two Hungarian sentences (*"Hát valószínűleg ... Köszönöm"*) would have gone missing in the original transcript. Instead, the original Whisper implementation hallucinates and produces output not in line with the source audio (*"1034, this ... 1033"*).

3.2 Vocal separation

Different vocal separation models have been tested, which can be found in table 4. While for the human ear the resulting audio of the different models sound almost equal, notable variations emerged in Whisper's transcripts. Compared to the transcript obtained without vocal separation, the vocal separation steps lead to longer transcripts, meaning that more speech is picked up and transcribed. Some vocal separation models however make more errors in word recognition than the baseline transcription. These models sometimes inadvertently remove too many parts of the speech, containing essential signal. This gives "too-cleaned" audio that lacks certain nuances of speech or background context in the audio, resulting in an increase in erroneous words or entirely missed words in the transcript.

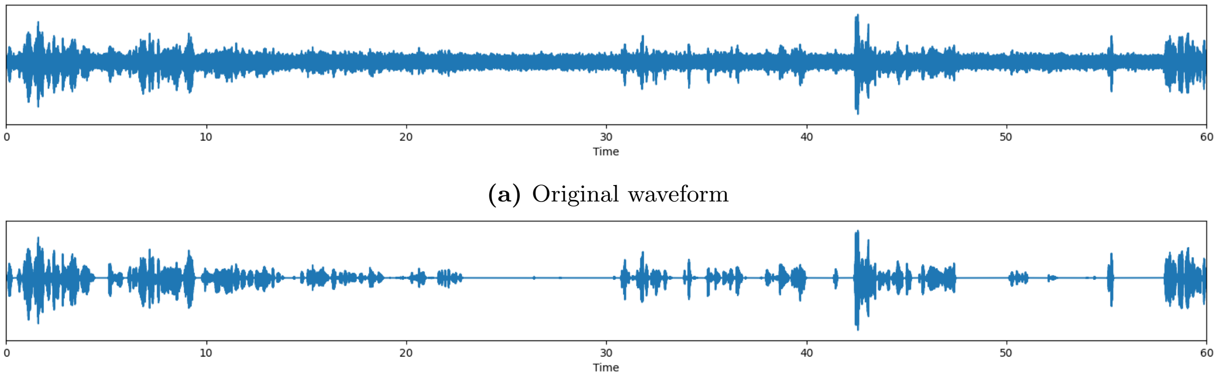
Based on human evaluation of the models over multiple recordings, BS RoFormer demonstrated the best performance, followed by Denoise SDR and MDX23C. A more detailed assessment can be found in

```
00:39:37,768 --> 00:41:29,867
[SPEAKER_00]: 133.2, 741, bye-bye.



[SPEAKER_00]: 1034, this is 9000
   feet altitude, 1033.

1033, 1033, 1033, 1033, 1033,
   1033, 1033, 1033, 1033, 1033,
   1033, 1033, 1033, 1033.


[SPEAKER_03]: Thank you.

[SPEAKER_00]: Miser 7407 turn
   right heading 055.
```

```
00:39:47,115 --> 00:41:29,867
[SPEAKER_00]: 133-2, Silenzio,
   7-4-1, bye bye.

[SPEAKER_03]: Hát valószínűleg
   végig elpályázunk, és ilyen
   elég nagy térközöket kell
   csinálnunk, próbálunk valamit
   alkotni.

[SPEAKER_00]: Jó, jó, akkor arra
   számítok, köszönöm.

[SPEAKER_03]: Köszönöm.

[SPEAKER_00]: Miser 7407 turn
   right heading 055.
```

Figure 5: Comparison of single-language Whisper transcription (english) vs. multi-language, auto-detected Whisper transcript. Audio is from an ATC recording, the speech before and after this excerpt is English-spoken.
The Hungarian text reads in English: *Well, we'll probably go all the way, and we'll have to make some pretty big gaps, trying to create something. Okay, okay, I'll expect that then, thank you. Thank you*

table 4, in which models were manually scored based on multiple segments from three audio recordings. The BS RoFormer model exhibited particularly effective noise reduction capabilities relating to Whisper while maintaining audio clarity for human listeners. Figure 6 shows the effect of vocal separation, showing the waveform of the original audio and the waveform of the vocal-separated audio. Figure 7 visualizes the Voice Activity Detection (VAD) segmentation using pyannote.audio. For this short (one-minute) audio fragment, one can see that the segments do not differ a lot, but the waveforms show that a drastic noise reduction has been applied.



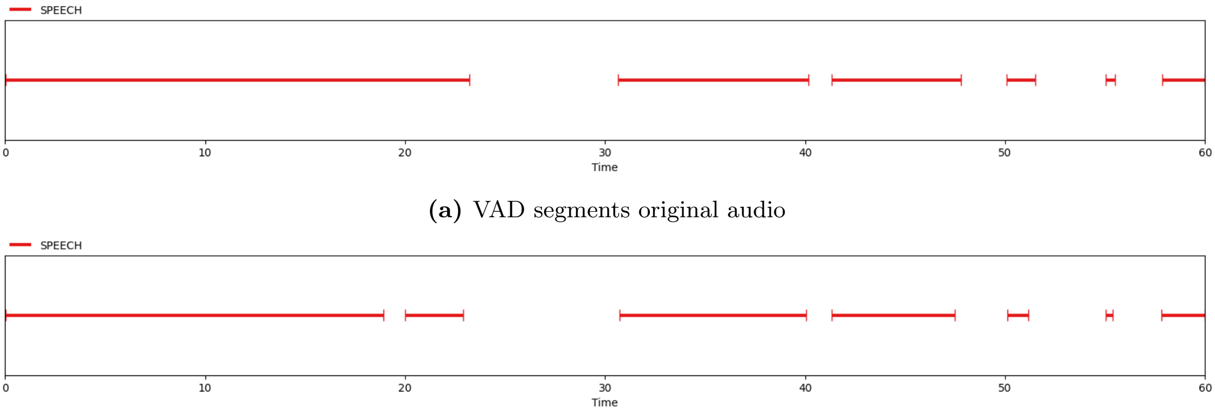
(a) Original waveform

(b) Waveform after vocal separation, vocals only.

Figure 6: Waveform before and after vocal separation.

Model name	Rated
BS-Roformer-Viperx-1297	★★★★★
Mel-Roformer-Denoise-Aufr33	★★★★⯪
Mel-Roformer-Denoise-Aufr33-Aggr	★★★★☆
MDX23C-InstVoc HQ 2	★★★★☆
UVR-MDX-NET Voc FT	★★★⯪☆
Kim Vocal 2	★★★☆☆
Demucs v4: htdemucs_ft	★★★☆☆
UVR-MDX-NET Inst HQ 4	★★⯪☆☆

Table 4: Performance comparison of various vocal separation models, with ratings presented relative to each other.



(a) VAD segments original audio

(b) VAD Segments after vocal separation, vocals only.

Figure 7: pyannote.audio VAD segments after segmentation of original and vocal-separated audio.

3.3 LLM Analysis

3.3.1 Transcript correction

In the first step, LLMs were used for correcting transcripts. From a manual comparison of outputs generated by Qwen 2.5 and Llama 3.1, Qwen 2.5 is found to correct the transcripts the best, and remove the most errors. For generating a short summary, which is used as context to the text correction prompt, Llama 3.1 8B is used, and suffices. The advantage of running a smaller model is the increased inference speed.

As an example, a transcription of a Dutch Nieuwsuur interview with Mark Rutte[5] was made and corrected using the LLM correction method described. A short excerpt of this corrected transcript can be seen in figure 8. As you can see, some words were incorrectly transcribed by Whisper, but corrected in the LLM correction step. Sometimes the LLM also makes corrections of which one can argue they are unneeded, like replacing ’heb’ with ’had’ in the example, but it does not change the meaning of the sentence too much, and in that sense, it is not a significant problem.

Calculating a WER after transcript correction was considered, but ultimately not pursued. The LLM transcription correction step, besides fixing spelling errors (which would reduce the WER), also removes duplicate words and sometimes reorders words in a sentence to correct the grammatical structure.

[5] https://www.youtube.com/watch?v=Gr31iRSzD5A

> SPEAKER_01: En ik kan me zo voorstellen, dan ga je naar huis, dat er dan toch een moment moet zijn dat je denkt, wacht even. Als iedereen het op mij gericht heeft en iedereen wil dat ik weg ga, ga je dan niet aan jezelf twijfelen? Ga je dan niet aan iemand vragen, wacht even, hou ik nou in iets vast aan iets waar ik niet aan vast moet houden?
> SPEAKER_00: Nou, niet twijfelen, maar het puntstukpunt is, ik ben lijsttrekker van mijn partij geworden met volle overtuiging. Er ligt een enorme kluswerkklus voor de komende vier jaar om Nederland uit die coronacrisis te leiden. Dus ik heb dat met volle overtuiging gedaan. We hebben ook die verkiezingen gewonnen. Ik hebhad die dag, of eigenlijk de week daarvoor, een grote fout gemaakt. Ik had een verkeerde herinnering aan of ik het nou wel of niet over Pieter Omtzigt had gehad. Overigens niet in functie eldersinelders.

Figure 8: Excerpt from a transcription of Mark Rutte's Nieuwsuur interview, to demonstrate the effect of the transcript correction step.

While this improves readability and comprehensibility, it generally has a neutral or even negative effect on WER — at best, it leaves the WER unchanged, and at worst, it increases it. Therefore, WER is not a suitable metric for evaluating the corrected transcript and has not been included in this analysis.

3.3.2 Insight generation

As a second step, LLMs are employed to answer questions based on the transcript, to generate insights. For the generation of insights, Llama 3.1 is found to be correct in the answers it gives, but quite often misses some information. Qwen 2.5 gives more extensive and useful answers, but if necessary information to answer the question is not found in the transcript, it tends to make up some facts or provide irrelevant answers. In general Qwen 2.5 was chosen to be used as main LLM for this purpose.

To effectively use the generated data, a small web UI was created. The UI allows the user to upload an audio file, choose processing options like Whisper model, LLM model etc. After transcription, correction and LLM insight generation the results can be viewed in the UI. The UI shows the Whisper transcript, the LLM corrected version of the transcript, but also a diff view between the two, to clearly see what corrections the LLM makes. This helps the user to quickly assess the quality of the corrections and make manual changes if needed. The second column shows the insights, the answers to the questions the LLM generated. A screenshot of the UI is visible in figure 9.

3.3.3 LLM model training

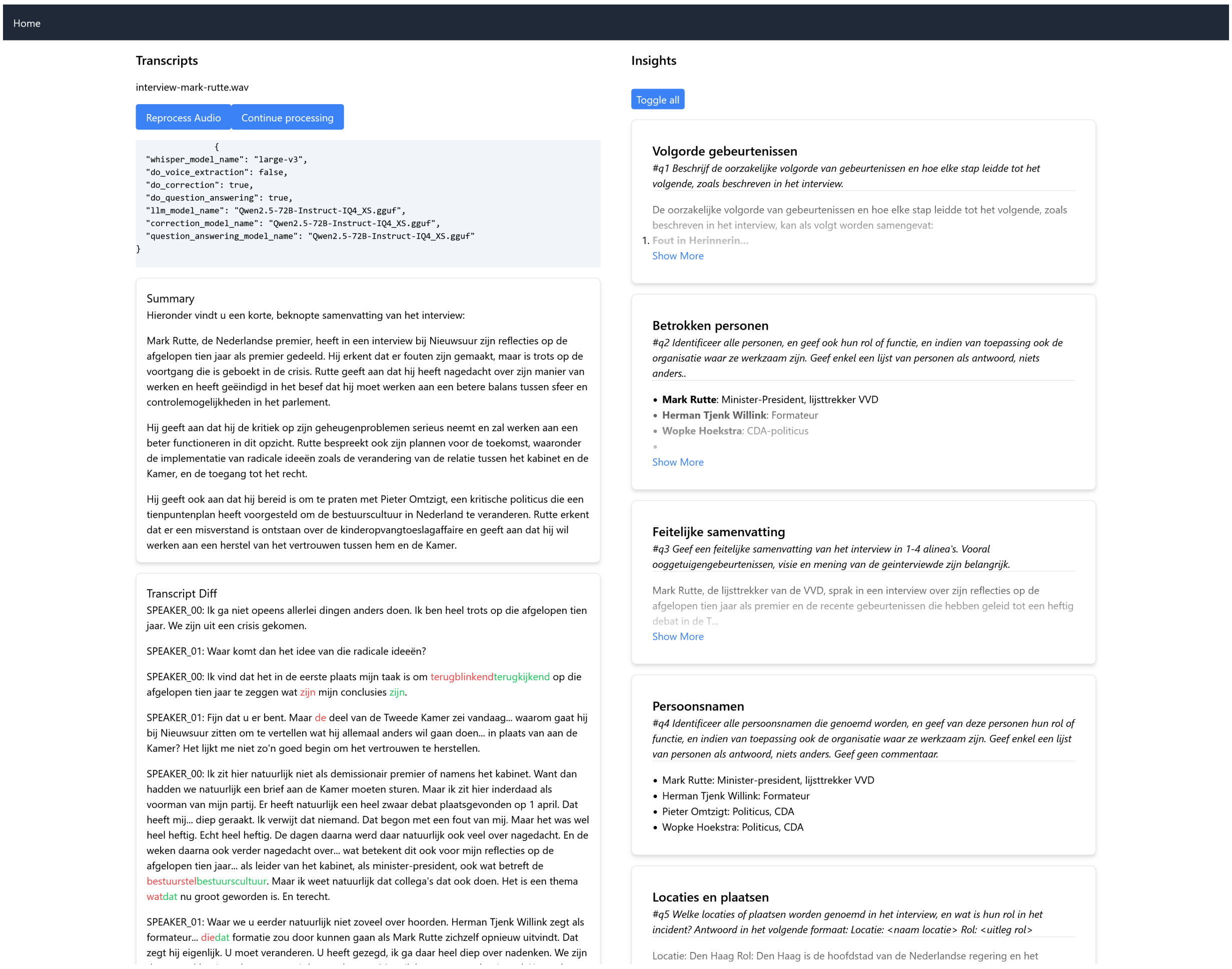
Tuning a Dutch-specific model was attempted by tuning Llama 3.1 8B with a dataset of Dutch prompts, released by Bram van Roy[6], using unsloth[29] tuning code. The resulting tuned Llama 3.1 8B model does improve Dutch performance, but still far from Llama 3.1 70B performance, so further research or benchmarking was not attempted.

4 Discussion

This research explores the application of AI in enhancing investigative processes, and specifically focuses on transcription and insight generation from audio data. I will first discuss the steps undertaken to improve audio transcription with Whisper, and then reflect on generating insights using local LLMs.

Using synthetic data, like was done with the TTS dataset, to tune and improve model performance not only addresses the scarcity of real-world training data in some fields or for niche topics, but also tackles rising concerns related to privacy and copyright. Groq, an AI hardware startup, released a Llama 3 fine-tune that was trained using only ethically generated synthetic data, thereby showing that it is possible to achieve industry comparable performance without requiring vast amounts of real-world data [30].

[6] https://huggingface.co/datasets/BramVanroy/ultra_feedback_dutch_cleaned



Transcripts

interview-mark-rutte.wav

```
              {
"whisper_model_name": "large-v3",
"do_voice_extraction": false,
"do_correction": true,
"do_question_answering": true,
"llm_model_name": "Qwen2.5-72B-Instruct-IQ4_XS.gguf",
"correction_model_name": "Qwen2.5-72B-Instruct-IQ4_XS.gguf",
"question_answering_model_name": "Qwen2.5-72B-Instruct-IQ4_XS.gguf"
}
```

Summary

Hieronder vindt u een korte, beknopte samenvatting van het interview:

Mark Rutte, de Nederlandse premier, heeft in een interview bij Nieuwsuur zijn reflecties op de afgelopen tien jaar als premier gedeeld. Hij erkent dat er fouten zijn gemaakt, maar is trots op de voortgang die is geboekt in de crisis. Rutte geeft aan dat hij heeft nagedacht over zijn manier van werken en heeft geëindigd in het besef dat hij moet werken aan een betere balans tussen sfeer en controlemogelijkheden in het parlement.

Hij geeft aan dat hij de kritiek op zijn geheugenproblemen serieus neemt en zal werken aan een beter functioneren in dit opzicht. Rutte bespreekt ook zijn plannen voor de toekomst, waaronder de implementatie van radicale ideeën zoals de verandering van de relatie tussen het kabinet en de Kamer, en de toegang tot het recht.

Hij geeft ook aan dat hij bereid is om te praten met Pieter Omtzigt, een kritische politicus die een tienpuntenplan heeft voorgesteld om de bestuurscultuur in Nederland te veranderen. Rutte erkent dat er een misverstand is ontstaan over de kinderopvangtoeslagaffaire en geeft aan dat hij wil werken aan een herstel van het vertrouwen tussen hem en de Kamer.

Transcript Diff

SPEAKER_00: Ik ga niet opeens allerlei dingen anders doen. Ik ben heel trots op die afgelopen tien jaar. We zijn uit een crisis gekomen.

SPEAKER_01: Waar komt dan het idee van die radicale ideeën?

SPEAKER_00: Ik vind dat het in de eerste plaats mijn taak is om terugblinkendterugkijkend op die afgelopen tien jaar te zeggen wat zijn mijn conclusies zijn.

SPEAKER_01: Fijn dat u er bent. Maar de deel van de Tweede Kamer zei vandaag... waarom gaat hij bij Nieuwsuur zitten om te vertellen wat hij allemaal anders wil gaan doen... in plaats van aan de Kamer? Het lijkt me niet zo'n goed begin om het vertrouwen te herstellen.

SPEAKER_00: Ik zit hier natuurlijk niet als demissionair premier of namens het kabinet. Want dan hadden we natuurlijk een brief aan de Kamer moeten sturen. Maar ik zit hier inderdaad als voorman van mijn partij. Er heeft natuurlijk een heel zwaar debat plaatsgevonden op 1 april. Dat heeft mij... diep geraakt. Ik verwijt dat niemand. Dat begon met een fout van mij. Maar het was wel heel heftig. Echt heel heftig. De dagen daarna werd daar natuurlijk ook veel over nagedacht. En de weken daarna ook verder nagedacht over... wat betekent dit ook voor mijn reflecties op de afgelopen tien jaar... als leider van het kabinet, als minister-president, ook wat betreft de bestuurstelbestuurscultuur. Maar ik weet natuurlijk dat collega's dat ook doen. Het is een thema watdat nu groot geworden is. En terecht.

SPEAKER_01: Waar we u eerder natuurlijk niet zoveel over hoorden. Herman Tjenk Willink zegt als formateur... diedat formatie zou door kunnen gaan als Mark Rutte zichzelf opnieuw uitvindt. Dat zegt hij eigenlijk. U moet veranderen. U heeft gezegd, ik ga daar heel diep over nadenken. We zijn

Insights

Volgorde gebeurtenissen

#q1 Beschrijf de oorzakelijke volgorde van gebeurtenissen en hoe elke stap leidde tot het volgende, zoals beschreven in het interview.

De oorzakelijke volgorde van gebeurtenissen en hoe elke stap leidde tot het volgende, zoals beschreven in het interview, kan als volgt worden samengevat:

1. **Fout in Herinnerin...**

Betrokken personen

#q2 Identificeer alle personen, en geef ook hun rol of functie, en indien van toepassing ook de organisatie waar ze werkzaam zijn. Geef enkel een lijst van personen als antwoord, niets anders..

- **Mark Rutte**: Minister-President, lijsttrekker VVD
- **Herman Tjenk Willink**: Formateur
- **Wopke Hoekstra**: CDA-politicus
-

Feitelijke samenvatting

#q3 Geef een feitelijke samenvatting van het interview in 1-4 alinea's. Vooral ooggetuigengebeurtenissen, visie en mening van de geinterviewde zijn belangrijk.

Mark Rutte, de lijsttrekker van de VVD, sprak in een interview over zijn reflecties op de afgelopen tien jaar als premier en de recente gebeurtenissen die hebben geleid tot een heftig debat in de T...

Persoonsnamen

#q4 Identificeer alle persoonsnamen die genoemd worden, en geef van deze personen hun rol of functie, en indien van toepassing ook de organisatie waar ze werkzaam zijn. Geef enkel een lijst van personen als antwoord, niets anders. Geef geen commentaar.

- Mark Rutte: Minister-president, lijsttrekker VVD
- Herman Tjenk Willink: Formateur
- Pieter Omtzigt: Politicus, CDA
- Wopke Hoekstra: Politicus, CDA

Locaties en plaatsen

#q5 Welke locaties of plaatsen worden genoemd in het interview, en wat is hun rol in het incident? Antwoord in het volgende formaat: Locatie: <naam locatie> Rol: <uitleg rol>

Locatie: Den Haag Rol: Den Haag is de hoofdstad van de Nederlandse regering en het

Figure 9: A screenshot of the web UI showing the transcript and insights generated from Mark Rutte's Nieuwsuur interview.

An interesting finding from this research is that models fine-tuned with the interview dataset show a 20% improvement, even though the dataset is relatively small at only 8 hours. This highlights the effectiveness of the proposed method for splitting and aligning audio into training samples using DTW-RA. By aligning and filtering, this approach can be applied to various audio-transcript pairs that may not initially be aligned or where parts of the transcript are diverting from the audio, as was the case with the data used in this study. The amount of data available from the DSB was limited, but as new interviews are transcribed, they can expand the dataset. Given these promising results, I anticipate further performance gains with an enlarged dataset.

The final model, tuned on the mixed dataset, performs 18% better than the untuned large-v3 model, but still has a WER of 18.72% on the test dataset. To use such a model on Dutch texts in production, this is quite a high WER. It means that on average, one out of five or six words needs correction. The transcription correction step using an LLM certainly helps to reduce this, but that step is quite resource heavy, and there might be some situations where this step is not possible, for lack of computational power. I expect big improvements to be possible when more transcribed interviews become available for training. As mentioned before, the size of the interview dataset is currently quite small, because of the small amount of data available.

Since the PEFT tuning methods are very efficient, both in memory and computation time, retraining of the model each time more entries are added to the dataset is quite feasible. This could even be set up in a (semi-) automated way, to update the dataset every time new audio transcripts become available, and release a new update of the tuned model based on this. In certain cases with specific domain vocabulary, a few interviews can be transcribed/corrected manually, and be used to fine-tune the model, that can then be used for the rest of the audio. For this, a kind of live update or continuously trained Whisper model could be set up.

The research on TTS further highlights that fine-tuned models can be created that specifically focus or excel at specific domains, for example the maritime sector, proving the value of synthetic or targeted datasets for domain-specific accuracy.

AI models for transcription and text generation, as explored in this research, can greatly assist forensic investigations, particularly by providing valuable insights in the early stages of the investigation process. Automated transcription reduces the need for manual work, allowing investigators to gain rapid access to information that guides their next steps.

For situations where high accuracy is required, such as in court, machine-generated transcripts could be proofread and corrected by a human, to meet the high standards of evidence material. Though the Dutch Safety Board may not require court-grade transcripts, this could be relevant for other authorities. Eventually, I expect the models to improve even further, and with that, the need for human proofreading or correction will decrease.

The transcription correction step using LLMs has proven effective in this research, especially when leveraging larger models capable of handling Dutch well. A strong understanding of the target language is crucial for this step. For Dutch, open-access LLMs with substantial language proficiency are available, but this may not be the case for all languages. Some languages are underrepresented in LLM training data, and others are only supported in the largest model categories, which, because of their hardware demands, may not be feasible for many organizations to run on-premise.

The insights generated based on the transcript using an LLM will help investigators in quickly understanding the main points of the interview. Timelines and list of involved parties will be helpful reference material, and other insights highlighting possible inconsistencies might even draw attention to aspects the investigator might not have considered. Eventually, linking multiple interview transcripts to a project enables generating project-wide summaries, timelines or other insights, helping investigators to get a good overview of the main points of the combined interviews.

4.1 Limitations

The research faced several limitations, primarily related to hardware capacity and dataset constraints. For Whisper, a key limitation was the relatively small size of the interview dataset available for model training and evaluation. This limited dataset meant that the model's performance could not be as robustly fine-tuned as desired. Also, the audio transcriptions available, were not always correct at word-level. As a remedy, the DTW-RA step was introduced to enhance alignment, but this does not fully solve the limitations of the data that was available. The results, however, definitely show the potential of this method in extending the domain and accuracy of the Whisper model, but more data would be needed for a robuster and more complete tuning.

For LLMs, due to limited VRAM capacity, models larger than 72 billion parameters were not evaluated, restricting the ability to test even more advanced models that may have provided improved performance. Additionally, the models used were quantized to fit within the available GPU memory, which introduced a slight reduction in model quality due to the lower precision of quantized weights.

During this research, it became increasingly obvious that WER as a metric to judge the performance of an ASR system has severe limitations. Automatically, ASR models that produce highly literal transcript, report higher WER scores than ASR systems that generate more natural or human-like transcriptions. For instance, if a person says an incorrect word, and corrects him/herself, a human transcriber will be likely to only transcribe the correct word. Similarly, if the person speaking joins words together in a grammatically incorrect manner, a human transcriber will probably restructure the sentence to adhere to basic grammatical rules. Additionally, in cases where a speaker repeats a word several times, such as can happen because of thinking aloud, or because of misunderstanding by the listener, the transcript will likely include the word only once. Moreover, omission or wrong transcription is more severe for some words than for others. For example, if the word 'the' is left out from a sentence, it is unlikely that the overall comprehension of the sentence is impacted. However, omitting a similarly short word, like 'not', has the potential of completely inverting the meaning of the sentence.

For ASR systems like Whisper, and especially for the interview dataset created in this research, it holds that the transcription style is more 'natural' rather than strictly word-accurate. A model trained using this data will also generate a very natural, readable transcript, but because of the nature of this transcript, is likely to yield higher WER scores. Also, an ASR model's ability to accurately transcribe domain-specific jargon can vary, a factor not fully reflected in popular metrics, as the common test datasets contain vocabulary that closely matches the vocabulary of the training datasets.

Reflecting on this issue, I brainstormed about scoring transcripts based on their closeness to the reference sentence. In other words: how well is the meaning of the transcript aligned with the meaning of the reference sentence. Opportunities might exist in employing LLM's to score two sentences on semantic closeness, although one should keep in mind that such a solution would be much more resource-intensive than a simple benchmark like WER. Finding an alternative for WER has been the subject of study already, as surveys on alternatives show [31]. Research initiatives into the direction of using language models already exists, like SeMaScore [32]. Evaluating, using or extending these solutions has not been done in this study, but could be interesting future work.

4.2 Future work

Future work could focus on two main areas of improvement: improving datasets and enhancing the flexibility of insight generation.

For the TTS audio, generating TTS data through a commercial provider rather than open-source models could produce higher-quality audio for training. Commercial TTS solutions generally provide more realistic and diverse vocal outputs, which would likely improve the robustness and generalizability of the model's training data. So, one might consider generating audio using Google TTS services, repeat the training procedure, and see if that gives significant quality improvements.

The method for aligning, filtering and splitting audio-transcript pairs into training segments, that was applied on DSB interview transcripts, is promising, but would greatly benefit from a larger dataset. Possibly combining this with better TTS audio, has the potential of enhancing model accuracy and can allow for better handling of varied audio inputs, further benefiting the transcription quality and reliability for forensic applications.

Currently, the insight generation process relies on a fixed set of questions. Expanding this to allow for more dynamic, context-sensitive question generation could enrich the model's insights and enable more adaptable responses to different audio content. A more flexible system would better handle the nuances of forensic investigations and could potentially be tailored to prioritize specific types of information relevant to different case or interview types.

Finally, a limitation mentioned is the ineffectiveness of WER as a benchmark for this kind of research. Further research could focus on developing an ASR benchmark that is able to capture the worth and nuanced correctness of more natural transcripts instead of favoring strict word-level transcripts.

4.3 Ethics

In this research, interviews performed by the DSB were used to construct a dataset. Because of the confidential nature of these interviews, the tuned model stays inside the organization and is not published or shared elsewhere. Additionally, to further prevent the model learning personal details, the interview audio and transcript manually are redacted to remove the first part of the interview, in which people generally introduce themselves. The TTS dataset was fully based on publicly available texts, using an open TTS model, so does not contain any private data, and thus can be shared or published without concerns.

5 Conclusion

This research explored innovative approaches to improving audio transcription and analysis using advanced AI technologies, specifically focusing on enhancing transcription quality for investigative processes at the Dutch Safety Board (DSB). The research demonstrated several key strategies for domain-specific audio transcription and insight generation.

The research firstly shows the potential of fine-tuning Whisper models using both synthetic Text-to-Speech (TTS) and interview-based datasets for domain-specific transcription. While the TTS dataset did not significantly improve transcription quality, the interview-based dataset achieved a notable 20% reduction in Word Error Rate (WER), highlighting the value of carefully curated, domain-specific training data.

Summing up, the methodological innovations in this study include:

- Developing a novel approach to creating training datasets through TTS and interview audio alignment
- Using advanced parameter-efficient fine-tuning techniques like LoRa and AdaLoRa
- Exploring vocal separation techniques to improve audio quality before transcription using Whisper
- Utilizing large language models for transcript correction and insight generation

6 References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, *Robust speech recognition via large-scale weak supervision*, Dec. 6, 2022. DOI: 10.48550/arXiv.2212.04356. arXiv: 2212.04356[cs,eess]. [Online]. Available: http://arxiv.org/abs/2212.04356 (visited on 04/09/2024).

[2] G. Gerganov, *Ggerganov/whisper.cpp*, original-date: 2022-09-25T18:26:37Z, Apr. 9, 2024. [Online]. Available: https://github.com/ggerganov/whisper.cpp (visited on 04/09/2024).

[3] *SYSTRAN/faster-whisper*, original-date: 2023-02-11T09:17:27Z, Apr. 5, 2024. [Online]. Available: https://github.com/SYSTRAN/faster-whisper (visited on 04/05/2024).

[4] M. Bain, J. Huh, T. Han, and A. Zisserman, *WhisperX: Time-accurate speech transcription of long-form audio*, Jul. 11, 2023. DOI: 10.48550/arXiv.2303.00747. arXiv: 2303.00747[cs,eess]. [Online]. Available: http://arxiv.org/abs/2303.00747 (visited on 04/05/2024).

[5] M. Bain, *M-bain/whisperX*, original-date: 2022-12-09T02:34:23Z, Apr. 9, 2024. [Online]. Available: https://github.com/m-bain/whisperX (visited on 04/09/2024).

[6] I. Dielen, "Improving the automatic speech recognition model whisper with voice activity detection," Accepted: 2023-09-06T10:08:44Z, Master Thesis, 2023. [Online]. Available: https://studenttheses.uu.nl/handle/20.500.12932/45045 (visited on 04/05/2024).

[7] A. Mul, "Enhancing dutch audio transcription through integration of speaker diarization into the automatic speech recognition model whisper," Accepted: 2023-08-11T00:02:43Z, Master Thesis, 2023. [Online]. Available: https://studenttheses.uu.nl/handle/20.500.12932/44643 (visited on 04/02/2024).

[8] "Fine-tune whisper for multilingual ASR with transformers." (), [Online]. Available: https://huggingface.co/blog/fine-tune-whisper (visited on 04/23/2024).

[9] S. Grundmann. "Fine-tuning whisper for dutch language: The crucial role of size," Medium. (Jul. 19, 2023), [Online]. Available: https://blog.ml6.eu/fine-tuning-whisper-for-dutch-language-the-crucial-role-of-size-dd5a7012d45f (visited on 04/05/2024).

[10] J. C. Vásquez-Correa, H. Arzelus, J. M. Martin-Doñas, J. Arellano, A. Gonzalez-Docasal, and A. Álvarez, "When whisper meets TTS: Domain adaptation using only synthetic speech data," in *Text, Speech, and Dialogue*, K. Ekštein, F. Pártl, and M. Konopík, Eds., Cham: Springer Nature Switzerland, 2023, pp. 226–238, ISBN: 978-3-031-40498-6. DOI: 10.1007/978-3-031-40498-6_20.

[11] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2020, pp. 439–444. DOI: 10.1109/CISP-BMEI51763.2020.9263564. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9263564 (visited on 04/23/2024).

[12] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, Jun. 2021, pp. 5674–5678. DOI: 10.1109/ICASSP39728.2021.9414778. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9414778 (visited on 04/23/2024).

[13] Q. Wang, Y. Huang, G. Zhao, E. Clark, W. Xia, and H. Liao, *DiarizationLM: Speaker diarization post-processing with large language models*, Feb. 6, 2024. DOI: 10.48550/arXiv.2401.03506. arXiv: 2401.03506[cs,eess]. [Online]. Available: http://arxiv.org/abs/2401.03506 (visited on 04/05/2024).

[14] T. J. Park, K. Dhawan, N. Koluguri, and J. Balam, "Enhancing speaker diarization with large language models: A contextual beam search approach," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, Apr. 2024, pp. 10 861–10 865. DOI: 10.1109/ICASSP48485.2024.10446204. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10446204 (visited on 04/05/2024).

[15] A. Trabelsi, L. Werey, S. Warichet, and E. Helbert, "Is noise reduction improving open-source ASR transcription engines quality?," presented at the 16th International Conference on Agents and Artificial Intelligence, Apr. 9, 2024, pp. 1221–1228, ISBN: 978-989-758-680-4. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0012457100003636 (visited on 04/09/2024).

[16] H. Bredin, "Pyannote.audio 2.1 speaker diarization pipeline: Principle, benchmark, and recipe," presented at the Proc. Interspeech 2023, 2023, pp. 1983–1987. DOI: 10.21437/Interspeech.2023-105. [Online]. Available: https://www.isca-archive.org/interspeech_2023/bredin23_interspeech.html (visited on 04/15/2024).

[17] E. J. Hu, Y. Shen, P. Wallis, *et al.*, *LoRA: Low-rank adaptation of large language models*, Oct. 16, 2021. DOI: 10.48550/arXiv.2106.09685. arXiv: 2106.09685[cs]. [Online]. Available: http://arxiv.org/abs/2106.09685 (visited on 07/23/2024).

[18] Q. Zhang, M. Chen, A. Bukharin, *et al.*, *AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning*, Dec. 20, 2023. DOI: 10.48550/arXiv.2303.10512. arXiv: 2303.10512[cs]. [Online]. Available: http://arxiv.org/abs/2303.10512 (visited on 07/23/2024).

[19] E. Casanova, K. Davis, E. Gölge, *et al.*, *XTTS: A massively multilingual zero-shot text-to-speech model*, Jun. 7, 2024. DOI: 10.48550/arXiv.2406.04904. arXiv: 2406.04904[cs,eess]. [Online]. Available: http://arxiv.org/abs/2406.04904 (visited on 06/28/2024).

[20] "Ggml/docs/gguf.md at master · ggerganov/ggml," GitHub. (), [Online]. Available: https://github.com/ggerganov/ggml/blob/master/docs/gguf.md (visited on 11/14/2024).

[21] P. M. Stahl, *Pemistahl/lingua-rs*, original-date: 2020-06-17T10:47:30Z, Nov. 13, 2024. [Online]. Available: https://github.com/pemistahl/lingua-rs (visited on 11/14/2024).

[22] R. Dan, *Echogarden-project/echogarden*, original-date: 2023-04-20T02:41:47Z, Jul. 31, 2024. [Online]. Available: https://github.com/echogarden-project/echogarden (visited on 08/02/2024).

[23] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Interspeech 2017*, ISCA, Aug. 20, 2017, pp. 498–502. DOI: 10.21437/Interspeech.2017-1386. [Online]. Available: https://www.isca-archive.org/interspeech_2017/mcauliffe17_interspeech.html (visited on 08/20/2024).

[24] D. Nielsen, "ScandEval: A benchmark for scandinavian natural language processing," in *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, T. Alumäe and M. Fishel, Eds., Tórshavn, Faroe Islands: University of Tartu Library, May 2023, pp. 185–201. [Online]. Available: https://aclanthology.org/2023.nodalida-1.20 (visited on 10/03/2024).

[25] A. Dubey, A. Jauhri, A. Pandey, *et al.*, *The llama 3 herd of models*, Aug. 15, 2024. DOI: 10.48550/arXiv.2407.21783. arXiv: 2407.21783. [Online]. Available: http://arxiv.org/abs/2407.21783 (visited on 10/24/2024).

[26] A. Yang, B. Yang, B. Hui, *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[27] Q. Team, *Qwen2.5: A party of foundation models*, Sep. 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen2.5/.

[28] M. Ravanelli, T. Parcollet, P. Plantinga, *et al.*, *SpeechBrain*, original-date: 2020-04-28T17:48:45Z, Nov. 18, 2024. [Online]. Available: https://github.com/speechbrain/speechbrain/ (visited on 11/18/2024).

[29] *Unslothai/unsloth*, original-date: 2023-11-29T16:50:09Z, Nov. 5, 2024. [Online]. Available: https://github.com/unslothai/unsloth (visited on 11/05/2024).

[30] M. Nuñez. "Groq's open-source llama AI model tops leaderboard, outperforming GPT-4o and claude in function calling," VentureBeat. (Jul. 18, 2024), [Online]. Available: https://venturebeat.com/ai/groq-open-source-llama-ai-model-tops-leaderboard-outperforming-gpt-4o-and-claude-in-function-calling/ (visited on 08/08/2024).

[31] A. Aksënova, D. van Esch, J. Flynn, and P. Golik, "How might we create better benchmarks for speech recognition?" In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, K. Church, M. Liberman, and V. Kordoni, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 22–34. DOI: 10.18653/v1/2021.bppf-1.4. [Online]. Available: https://aclanthology.org/2021.bppf-1.4 (visited on 08/01/2024).

[32] Z. Sasindran, H. Yelchuri, and T. V. Prabhakar, *SeMaScore : A new evaluation metric for automatic speech recognition tasks*, Jan. 15, 2024. DOI: 10.48550/arXiv.2401.07506. arXiv: 2401.07506[cs, eess]. [Online]. Available: http://arxiv.org/abs/2401.07506 (visited on 08/06/2024).

A Supplementary Materials

A.1 Correction prompt

```
SYSTEM: Je bent een behulpzame AI assistent voor de Onderzoeksraad voor
   Veiligheid. Je antwoordt altijd in het Nederlands.
Je taak is het verbeteren van een audio transcript dat is gegenereerd
   met spraakherkenning. Het transcript bevat vaak fouten, vooral bij
   bedrijfs- en persoonsnamen. Je moet de tekst verbeteren, maar wel zo
   veel mogelijk bij de originele stijl en terminologie blijven.
   Corrigeer spelling- en grammaticafouten, maar verander geen Engelse
   termen naar Nederlands. Vervang duidelijk verkeerde woorden zoals
   'sleeboot' door het juiste woord 'sleepboot'.

Houd je aan de volgende richtlijnen:
- Behoud de originele betekenis en stijl zoveel mogelijk
- Behoud Engelse terminologie waar deze gebruikt wordt
- Verander incorrect herkende woorden (bijvoorbeeld bedrijfsnamen) naar
   de juiste vorm
- Verbeter spelling- en grammaticafouten waar nodig
- Zorg dat de tekst goed leesbaar en vloeiend wordt

{f"Dit interview maakt deel uit van het project {project.name}:
   {project.description}" if project else ""}

Ter context een korte samenvatting van de context:
{short_summary}

Einde samenvatting, nu volgt het transcript:

USER: <transcript>" + batch + "</transcript>

ASSISTANT: Het verbeterde transcript is:
```

A.2 Insights: Question prompts

Volgorde gebeurtenissen
Beschrijf de oorzakelijke volgorde van gebeurtenissen en hoe elke stap leidde tot het volgende, zoals beschreven in het interview.
Combine prompt: Combineer de antwoorden op de vragen over de volgorde van gebeurtenissen om een volledige samenvatting te creëren.

Betrokken personen
Identificeer alle personen, en geef ook hun rol of functie, en indien van toepassing ook de organisatie waar ze werkzaam zijn. Geef enkel een lijst van personen als antwoord, niets anders.
Combine prompt: Combineer de lijsten van personen. Groepeer per organisatie. Neem gelijke namen samen. Verwijder namen van de vorm SPEAKER_xx.

Feitelijke samenvatting
Geef een feitelijke samenvatting van het interview in 1-4 alinea's. Vooral ooggetuigengebeurtenissen, visie en mening van de geinterviewde zijn belangrijk.
Combine prompt: Gegeven de verschillende feitenlijke samenvatting, combineer tot een samenvatting waar alle verschillende feiten ingenoemd worden. Geef zo nodig aan door welke persoon een mening of stelling geuit wordt.

Persoonsnamen
Identificeer alle persoonsnamen die genoemd worden, en geef van deze personen hun rol of functie, en indien van toepassing ook de organisatie waar ze werkzaam zijn. Geef enkel een lijst van personen als antwoord, niets anders. Geef geen commentaar.

Combine prompt: Combineer de lijsten van personen. Groepeer per organisatie. Neem gelijke namen samen. Verwijder namen van de vorm SPEAKER_xx.

Locaties en plaatsen

Welke locaties of plaatsen worden genoemd in het interview, en wat is hun rol in het incident?
Antwoord in het volgende formaat:
Locatie: `<naam locatie>`
Rol: `<uitleg rol>`
Combine prompt: Combineer de verschillende locaties tot een lijst. Sorteer op belangrijkheid.

Tijdlijn

Creëer een tijdlijn van de gebeurtenissen zoals beschreven in het interview, inclusief data, tijden en betrokken personen.
Combine prompt: Combineer de verschillende tijdlijnen tot 1 tijdlijn. Sorteer op chronologische volgorde.

Overzicht

Vat de belangrijkste feiten over het incident samen, inclusief betrokken personen, locaties, tijden, en gebeurtenissen.

Citaten

Identificeer en markeer belangrijke citaten die belangrijke feiten, meningen of gevoelens over het incident weergeven. Gebruik Markdown-blockquotes voor de citaten.

Data en tijdstippen

Haal alle data en tijdstippen uit het interview en plaats deze in chronologische volgorde.
Combine prompt: Combineer de verschillende tijdstippen tot 1 tijdlijn. Plaats deze in chronologische volgorde.

Technische problemen

Welke technische problemen worden besproken in het interview, en hoe hebben deze bijgedragen aan het ongeval? Als het interview niet genoeg informatie bevat om een goed antwoord te kunnen geven, zeg dat dan.

Tegenstrijdigheden

Identificeer eventuele tegenstrijdige verklaringen of inconsistenties in het interview. Als het interview niet genoeg informatie bevat om een goed antwoord te kunnen geven, zeg dat dan
Combine prompt: Maak een lijst met alle verschillende tegenstrijdige verklaringen. Benoem uit welk interview het afkomstig is.

Veiligheidsmaatregelen

Welke veiligheidsmaatregelen worden besproken in het interview en hoe worden deze geëvalueerd? Als het interview niet genoeg informatie bevat om een goed antwoord te kunnen geven, zeg dat dan
Combine prompt: Geef een compleet overzicht van alle besproken veiligheidsmaatregelen in de verschillende interviews.

Summary

1.) Analyseer de tekst en formuleer 5 vragen die samen de kern en belangrijkste punten van de inhoud weergeven.
2.) Houd bij het opstellen van de vragen rekening met:
a. Het centrale thema of hoofdargument.
b. Belangrijke ondersteunende ideeën.
c. Relevante feiten of bewijzen.
d. De bedoeling of het perspectief van de geinterviewden.
e. Mogelijke implicaties of conclusies.
3.) Beantwoord vervolgens elke vraag uitgebreid en gedetailleerd.