Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Alexander Michael Ås

A Norwegian Whisper Model for Automatic Speech Recognition

Master's thesis in Informatics
Supervisor: Jon Atle Gulla
Co-supervisor: Benjamin Kille
June 2023

NTNU
Norwegian University of Science and Technology

NorwAI

Alexander Michael Ås

A Norwegian Whisper Model for Automatic Speech Recognition

Master's thesis in Informatics
Supervisor: Jon Atle Gulla
Co-supervisor: Benjamin Kille
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

In the past few decades, automatic speech recognition (ASR) systems made significant progress, achieving high transcription accuracy across a wide range of languages. Today, ASR systems are indispensable components of various smart devices, particularly social robots. Social robots are designed to interact with humans in a natural and intuitive manner and are used in various ways, including language learning [8], tutoring [7], and for therapy of children with autism [11]. An example of a modern social robot is the Furhat robot by Furhat Robotics. It is used at the Norwegian Research Center for AI Innovation (NorwAI) to test and demonstrate language models developed at the center. Still, despite its modern technology, the speech recognition system of the Furhat robot is not ideal as it struggles with a range of Norwegian dialects, is very susceptible to background noise, and has difficulties understanding names. Moreover, while it is capable of transcribing spoken Norwegian to Bokmål, which is one of the two official written languages in Norway, it has no built-in support for the second official written language, that is, Nynorsk. In an effort to combat the issues with the current speech recognition system, this thesis investigates the adaption of Whisper [31] to the Furhat robot. Whisper is a state-of-the-art speech recognition model trained on $680,000$ hours of training data and supporting 96 different languages for multilingual speech recognition. The medium-sized Whisper model was fine-tuned on Bokmål and Nynorsk using the Norweigan Parliament Speech Corpus (NPSC) [35] dataset and evaluated on both languages with regard to the overall performance, noise robustness, the transcription of names, as well as speaker-related characteristics, such as dialect, age, and gender. The performance of the fine-tuned model was further compared to other state-of-the-art architectures, including a fine-tuned version of the small Whisper model and Wav2Vec 2.0 [4]. The model was compared and evaluated using the word error rate (WER), which is the number of insertions, deletions, and substitutions required for the prediction to match the ground-truth sentence. Fine-tuning the model improved the overall WER considerably in both written languages and model performance was generally not influenced by the age or gender of the speaker. Moreover, even though the WER starts to increase at high levels of noise with a signal-to-noise ratio of 10 dB or less, model performance remains stable at low levels of noise. However, while the overall dialect performance was significantly improved by fine-tuning, some dialects still caused the WER to spike. What is more, the WER increased in many cases if a name or abbreviation was present in the sentence, indicating that the transcription of names remains an issue.

Sammendrag

Talegjenkjenning har hatt betydelig fremgang i de siste årene og er blitt vesentlig flinkere i å transkribere lyd til tekst på forskjellige språk. I dag er teknologien uunnværlig og brukes i ulike smarte enheter, deriblant sosiale roboter. Sosiale roboter er designet til å kommunisere med mennesker på en naturlig og intuitiv måte og brukes blant annet for språklæring [8], undervisning [7], og behandling av barn med autisme [11]. Ett eksempel for en sosial robot er den såkalte Furhat roboten fra Furhat Robotics som brukes av det norske forskningssentret for AI-innovasjon (NorwAI) ved NTNU for å teste og demonstrere språkmodeller utviklet ved sentret. Til tross for at roboten er utstyrt med moderne og avansert teknologi er talegjenkjenningsmodellen ikke ideelt. Den sliter blant annet med en rekke norske dialekter, har store vansker med navn og forkortelser og er svært upålitelig når det er mye bakgrunnsstøy. Utover det støtter modellen bare Bokmål og er ikke i stand til å transkribere til Nynorsk. Målet ved denne oppgaven er derfor å undersøke om den nåværende modellen kan erstattes med Whisper [31]. Whisper er en avansert talegjenkjenningsmodell som ble trent på mer enn $680,000$ timer med data og støtter 96 forskjellige språk for talegjenkjenning. Den mellomstore Whisper modellen ble finjustert på Bokmål og Nynorsk ved hjelp av Stortingskorpuset [35] og ytelsen ble analysert med hensyn til støyrobusthet, transkribering av navn og talerelaterte egenskaper, som dialekt, alder og kjønn. Dessuten ble modellen sammenlignet med den lille Whisper modellen og Wav2Vec 2.0 [4] som begge ble trent av Nasjonalbiblioteket. Modellene ble sammenlignet og evaluert ved hjelp av ordfeilraten (WER), som måler antall ord som må legges til, slettes og erstattes for at prediksjonen stemmer overens med referansesetningen. Ordfeilraten ble betraktelig redusert både på Bokmål og Nynorsk, og resultatene viser at ytelsen ikke påvirkes av verken kjønn eller alder. I tillegg er ordfeilraten relativt stabil når støynivået er lavt, og det er først når signal-til-støyforholdet er på 10 dB eller mindre at den begynner å stige. Resultatene viser derimot at ytelsen er påvirket av talerens dialekt som fører til at ordfeilraten er litt høyere for noen dialekter mens den er lavere for andre. Videre er ordfeilraten litt større for setninger som inneholder navn eller forkortelser, noe som tyder på at Whisper ikke er unntatt problemet.

Preface

In dedication to my loving mother who could not take part in this journey.

This Master's thesis has been submitted to the Norwegian University of Science and Technology (NTNU) at the Faculty of Information Technology and Electrical Engineering (IE), Department of Computer Science (IDI). The thesis is part of the study program Master of Science in Informatics with a specialisation in artificial intelligence.

First and foremost, I would like to thank my supervisor Jon Atle Gulla and co-supervisor Benjamin Kille for the opportunity to write a thesis in collaboration with the Norwegian Research Center for AI Innovation (NorwAI) and for their support and extensive feedback during the project. I would also like to thank my fiancée Katharina for her invaluable support and creative input to my work. Lastly, I would like to give a special thanks to my father and Bjørg, my brothers and sisters, and all my friends for their guidance and continuous support during my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic Speech Recognition (ASR) is the recognition and transcription of spoken language in real-time [24]. It has made substantial progress in the past decade and is used on a daily basis throughout the world. With the proliferation of smart end-user devices with built-in virtual assistants, such as Siri and Google Assistant, ASR systems have become widely available, changing the way humans interact with computing devices. Speech recognition also plays a crucial role in the field of robotics, in particular social robots. Social robots are designed to autonomously or semi-autonomously interact and communicate with humans in a natural and intuitive way based on the behavioural norms that are expected by the people it interacts with [6]. Compared to other technologies, social robots allow for a more natural interaction due to their human-like appearance, which enables the user to pick up nonverbal cues alongside verbal interaction [8]. The ability to communicate nonverbal information is a crucial advantage of social robots and various studies investigate how social robots can be adapted to different domains, including language learning [8], tutoring [7], and for therapy of children with autism [11].

An example of a modern, social robot is the Furhat[1] robot by Furhat Robotics — a sophisticated and highly customizable social robot. It uses a back-projection system to project a face onto a translucent mask and an advanced face engine that enables the projection of different faces with highly expressive facial gestures mimicking human behaviour. It also ships with a built-in camera allowing the system to detect and track faces within its field of view, over 200 synthesised voices in over 35 different languages, and a speech recognition system. The Furhat robot is used extensively at the Norwegian Research Center for AI Innovation (NorwAI), a large research centre at NTNU focusing on innovative, AI-driven solutions[2]. Parts of the research are focused on developing large language models for the Norwegian language focused on specific tasks, such as question-answering or text summarization. The language models are often deployed to the Furhat robot as it enables more natural interaction with the model, which is particularly useful for demonstrating the models to external audiences. However, despite its cutting-edge technology, some challenges remain.

[1] https://furhatrobotics.com/

[2] https://www.ntnu.edu/norwai

1.1 Background and Motivation

An essential component of the Furhat robot is its speech recognition system. It is responsible for transcribing the audio signals communicated to the robot to the respective written domain, which is then used for querying the language models. At NorwAI, the goal is to develop models supporting both Bokmål and Nynorsk, which are the official written languages in Norway. Thus, it is paramount that the ASR system used in the robot is capable of transcribing to both languages. Norwegian is also a quite diverse language despite its relatively low number of native speakers and is characterised by a wide range of dialects. Although the dialects can be broadly split into four dialect families, i.e., Northern, Eastern, Western, and Trøndelag, the dialects vary quite a bit even within these groups [39]. As the interaction with the robot should be as natural and intuitive as possible, another requirement of the ASR system is the ability to transcribe the Norwegian dialects without needing the speaker to switch to a neutral dialect to be understood. Moreover, it should also work independently of the speaker's age and gender. The system should also be robust to background noise. The robot is frequently presented at conferences to demonstrate the technologies developed at NorwAI. Since conferences are typically characterised by frequent chatter in the background, the speech recognition system needs to be able to distinguish background noise from the speaker's voice without causing the transcription performance to deteriorate considerably. Lastly, if the speaker uses any names or abbreviations while talking with the robot, the ASR system should ideally be capable of transcribing most without any issues.

At the moment, the default speech recognition model that ships with the robot is used. However, even though it comes with built-in support for over 120 languages, including Norwegian Bokmål, it does not support Norwegian Nynorsk[3]. Hence, the queries need to be translated from Bokmål to Nynorsk using a translator, which is not ideal. It adds an additional processing step to the querying pipeline, causing the overall query time to increase, and the translations are in many cases sub-optimal. Moreover, the default model also has difficulties with some Norwegian dialects except for the Eastern dialect. This often leads to incorrect transcriptions and unexpected answers, forcing the speaker to switch to a neutral dialect to be understood. It was also observed that the system performs poorly in the presence of noise. The ASR system often does not recognise when the speaker stops talking, which results in the system waiting for additional input instead of generating an answer. The transcriptions also tend to diverge significantly from what has been said if the system is exposed to background noise. Finally, the current ASR system struggles quite a bit with the transcription of names, such as NorwAI or NTNU. These are often transcribed incorrectly, leading to an incorrect query being sent to the language models.

In the past few years, the development of more complex systems utilising the power of neural networks has led to sophisticated ASR systems that achieve high transcription accuracy in various languages, even in the presence of large amounts of noise. Whisper [31] is one of the latest speech recognition systems developed by OpenAI. It uses a state-of-the-art Transformer [38] architecture trained on a vast amount of labelled speech data exceeding $680,000$ hours, including $117,000$ hours on 96 languages other than English for multilingual speech recognition. What is more,

[3]https://furhatrobotics.com/docs/Furhat-Robotics-Technical-Product-Overview.pdf

the model is not only capable of transcribing spoken language but also translating it directly into English if the input data is in a different language. It is also quite robust to noise and outperforms other state-of-the-art ASR systems when exposed to high noise with a signal-to-noise (SNR) ratio of less than 10 dB. Whisper was trained on more than 266 hours of speech data in Bokmål and achieves an overall word error rate (WER) of 9.5 on the Fleurs [13] dataset using the largest model. The WER measures the total number of additions, deletions, and insertions required for the predicted transcription and ground truth to match. This score could be improved even more by fine-tuning the model on a high-quality dataset as suggested by the authors. Even though Whisper currently only supports Norwegian Bokmål for multilingual speech recognition tasks, it does support Nynorsk for translation purposes and was trained on 1889 hours of speech data in Nynorsk. This could potentially be leveraged to augment the model with the ability to transcribe to Nynorsk as well.

With the increased noise robustness and multi-language speech recognition capabilities, Whisper could be an ideal candidate to address some of the aforementioned issues with the current ASR system used in the robot. However, the model size plays a decisive role. The queries by the user should be transcribed correctly, but it is also important that an answer is provided by the robot within a reasonable time. The largest Whisper model uses more than 1550M parameters and although it has the best transcription quality, inference time is orders of magnitude slower than smaller models. The smallest model, on the other hand, is fast but at the cost of transcription quality as it achieves a significantly higher WER of 62.0 in Bokmål on the Fleurs [13] dataset. A solution to the problem could be to pick the medium-sized Whisper model, which has a quicker inference time than the largest model while also performing better than the smallest model, and fine-tune it on a high-quality dataset to improve transcription performance. Hence, the goal of this thesis is to investigate if the medium-sized Whisper model can be used to mitigate the issues of the current speech recognition system discussed at the beginning of this section.

1.2 Research Questions

As mentioned in the previous section, the overall objective of this thesis is to determine if the medium-sized Whisper model can be used instead of the default speech recognition model of the Furhat robot to address the poor performance on dialects and names, susceptibility to noise, and the inability to transcribe to Nynorsk. The overall goal is further split into a set of verifiable sub-goals, which are listed and discussed in this section.

The default Whisper models are not trained on Nynorsk for multilingual speech recognition. Even though the language parameter of the model can be set to Nynorsk, the WER of the model is extraordinarily high and the generated transcriptions are generally not in Nynorsk. Hence, the first objective is to explore if fine-tuning the medium-sized model on a high-quality dataset in Nynorsk can reduce the overall WER of the model and thereby extend Whisper with the ability to transcribe to Nynorsk.

RQ1: Can the WER on Nynorsk be reduced by fine-tuning the medium-sized Whisper model on a high-quality dataset?

Secondly, even though Whisper supports Norwegian Bokmål for multilingual speech recognition, the model was only trained on 266 hours of data, which is relatively low compared to the 438,218 hours of training data for English. Hence, the second objective of this thesis is to investigate if the overall WER of Whisper on Norwegian Bokmål can be decreased by fine-tuning it on a high-quality dataset in Bokmål.

RQ2: Does fine-tuning decrease the overall WER of the medium-sized Whisper model on Norwegian Bokmål?

Another requirement of the speech recognition system discussed in Section 1.1 is the ability to transcribe Norwegian independently of the dialect, age, and gender of the speaker. Thus, the third objective is to analyse if the transcription performance of the fine-tuned Whisper model is affected in any way by the dialect, age, or gender of the speaker measured by the WER metric.

RQ3: Is the WER of the fine-tuned, medium-sized Whisper model increased by the dialect, age, or gender of the speaker?

An additional major issue with the current speech recognition system is its susceptibility to background noise, which results in unexpected system behaviour and incorrect transcriptions. Therefore, the fourth objective is to investigate how the fine-tuned Whisper model behaves in the presence of real-world noise and how it affects the transcription performance in Norwegian.

RQ4: Does real-world background noise increase the WER of the fine-tuned, medium-sized Whisper model?

Lastly, the current system has major difficulties with the transcription of names and they are often transcribed incorrectly, causing the language models to receive incorrect queries. Consequently, the final objective is to check if this issue persists with the fine-tuned Whisper model.

RQ5: Do names or abbreviations increase the WER of the fine-tuned, medium-sized Whisper model?

1.3 Structure

The thesis is structured into 6 chapters. Chapter 2 starts by introducing speech recognition fundamentals and the core concepts of neural networks. It also discusses a range of neural network architectures, including recurrent-neural networks (RNNs), autoencoders, and sequence-to-sequence models, before continuing with the attention mechanism, a key component of the Transformer [38] architecture discussed at the end of the chapter. This is followed by an overview of end-to-end speech recognition architectures in Chapter 3 with a particular focus on modern speech recognition models, such as Wav2Vec 2.0 [4] and Whisper [31]. Chapter 4 provides an overview of the dataset and methodology used to fine-tune the medium-sized Whisper model before presenting and discussing the results of the thesis in Chapter 5. Lastly, Chapter 6 concludes with a discussion of the overall findings of this thesis and potential future work.

Chapter 2

Background Theory

This chapter introduces some of the concepts that are fundamental to modern speech recognition architectures, which are discussed in more detail in Chapter 3. Understanding the core ideas also plays a vital role in the evaluation in Chapter 5.

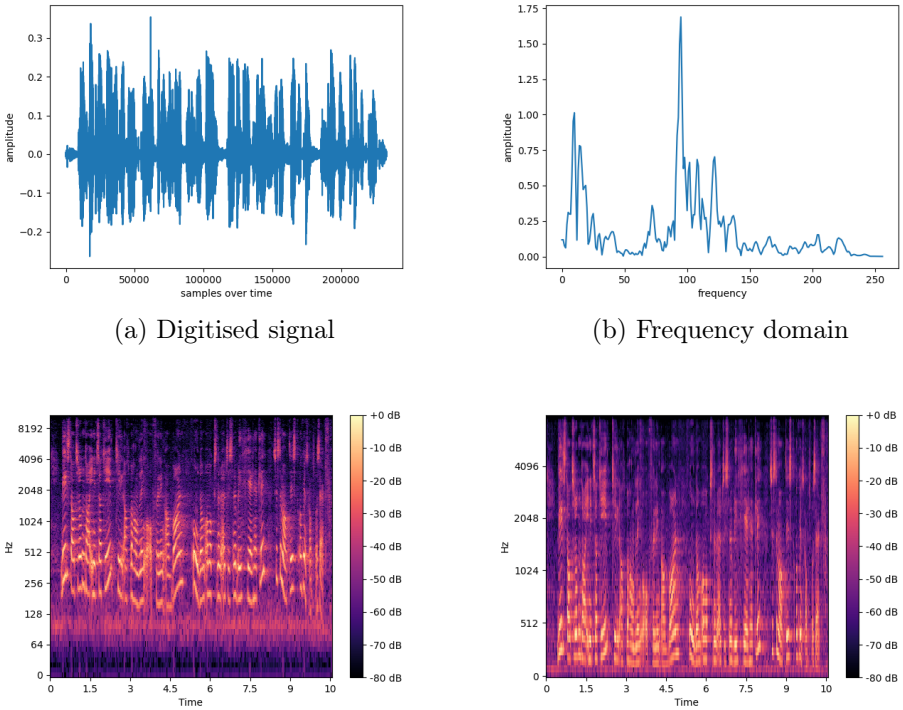
Section 2.1 starts by providing an overview of digitising analogue acoustic signals and converting them to a compatible digital representation, which is the initial step required before any speech data can be processed. This is followed by a short introduction to traditional speech recognition architectures using probabilistic models in Section 2.2. The following sections 2.3, 2.4, and 2.5, discuss the fundamentals of neural networks and introduce two important architectures, that is, recurrent neural networks (RNNs) and autoencoders. These are pivotal building blocks for so-called sequence-to-sequence models, which are introduced in Section 2.6. Finally, a summary of the core attention mechanisms is provided in Section 2.7 before introducing Transformers [38] in Section 2.8, which is the main architecture used in Whisper [31].

2.1 Digitisation of Analogue Acoustic Signals

The very first step of any ASR system is to convert the analogue speech signal to a suitable, digital representation. The goal of this section is to provide a brief overview of the steps involved in converting analogue signals to a digital representation such that they can be used by the speech recognition models to generate a transcription of the audio.

2.1.1 Analogue-to-Digital Conversion

Speech is an analogue signal generated by changes in air pressure resulting in compression waves [24]. To obtain a digital signal, the waves are picked up by a microphone which converts the changes in pressure to a corresponding analogue voltage [24]. The voltage is then put through an analogue-to-digital (A/D) converter, which samples the signal at a specific rate and outputs the digitised signal as a one-dimensional array defined by the bit depth of the A/D converter [24]. The digitised signal can be plotted in the time domain, where the amplitude of the samples is plotted over time [34]. An example based on a short extract from a speech given at the Norwegian parliament is given in Figure 2.1(a).



(a) Digitised signal

(b) Frequency domain

(c) Spectrogram

(d) Mel Spectrogram

Figure 2.1: All figures are obtained from the same speech extract at the Norwegian parliament. The speech was obtained from the Norwegian Parliament Speech Corpus (NPSC) [35] dataset. (a) depicts the raw, digitised signal. (b) shows the time-frequency domain of a small window of the audio signal using 512 samples. (c) is the spectrogram resulting from the signal in (a) and (d) is the resulting spectrogram with frequencies mapped to the Mel scale.

2.1.2 Mapping Raw Signals to the Frequency Domain

To facilitate extracting information from the audio signal, the signal (Figure 2.1(a)) can be mapped from the time domain to the frequency domain using a discrete Fourier transformation (DFT). The DFT decomposes a signal comprising N samples into two $\frac{N}{2} + 1$ sine and cosine output signals [34]. The resulting output signals are plotted in the frequency domain, which describes the amplitudes of the corresponding waves [34]. An example of a signal in the frequency domain is given in Figure 2.1(b), which is the result of applying the DFT to the signal in Figure 2.1(a).

2.1.3 Spectrograms

Figure 2.1(b) shows the spectral components of the signal at a specific instance of time. Splitting the input signals into multiple windows and computing the DFT for each window allows for the spectral contents to be plotted over time to visualise how the decomposition of the signal changes. The so-called spectrogram can be computed using the Short-Time Fourier Transform (STFT), which computes the DFT over short overlapping windows [1]. The spectrogram is often mapped to the log scale as the majority of the most significant signals are typically at lower frequencies [24]. This can also be seen in Figure 2.1(b). Applying the STFT to the entire audio extract from the Norwegian parliament results in the spectrogram plotted in Figure 2.1(c).

In many cases, the frequencies are further mapped to the Mel scale, which is a scale based on the human auditory system and, thus, more appropriate for visualising the spectral components of speech [24]. The frequencies f in Hz can be mapped to Mel by computing

$$m = 2595 \log_{10} \left(1 + \frac{f}{700}\right), \tag{2.1}$$

which acts logarithmic at higher frequencies and linear at lower frequencies [24]. Figure 2.1(d) illustrates the spectrogram of the parliament speech using the Mel scale.

2.2 Probabilistic Speech Recognition

This section is largely based on Chapter 8 in the book by Kamath, Liu, and Whitaker [24]. Before the widespread adoption of neural networks for speech recognition, ASR systems were traditionally realised using probabilistic models. In general, the goal of an ASR is to predict the most probable sequence of words $W = \{w_n \in V \mid n = 1, ..., N\}$ of length N using a vocabulary V from a sequence of acoustic features $X = \{x_t \in \mathbb{R}^D \mid t = 1, ..., T\}$ of length T. Thus, the goal can be defined as

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X). \tag{2.2}$$

Contrary to end-to-end ASR approaches (see Chapter 3), which aim at optimising $P(W|X)$ directly, statistical approaches split Equation 2.2 into multiple sub-models. By using Bayes' theorem and the law of total probability, Equation 2.2 can be rearranged as

$$\hat{W} = \underset{W,S}{\operatorname{argmax}} P(X|S)P(S|W)P(W), \tag{2.3}$$

[1] Source: https://librosa.org/doc/main/generated/librosa.stft.html

where $S = \{s_t \in Q \mid t = 1, ..., T\}$ is a sequence of sub-word states, such as phonemes, with Q denoting the set of possible states (phonemes). The objective of the acoustic model $P(X|S)$ is to map acoustic features to phonemes, the pronunciation or lexicon model $P(S|W)$ maps phonemes to actual words, and the language model $P(W)$ provides the probability for a given word sequence.

The most probable sequence of words is computed with the help of Hidden Markov Models (HMM), which aim at modelling the probability distribution of word sequences that caused the observable events, i.e., the sequence of acoustic features. HMMs consist of a set of non-observable state variables and a set of observable evidence variables, which are dependent on the state variables. In terms of ASR, the set of phonemes Q are the non-observable states and the acoustic features X reflect the observable evidence. An HMM can only be in a single state at a time and it may transition to another state with a certain transition probability $a_{ij} = P(s_t|s_{t-1})$. To be fully defined, an HMM further requires the initial state probability $\alpha = P(s_1)$ and the output probabilities $b_x = P(x|s)$ for an acoustic feature x given a state s to be defined. α, a_{ij}, and b_x are then optimised by training the HMM on several acoustic features and a target sequence of phonemes. Once the model is optimised, a sequence of words is obtained by mapping the acoustic features to a sequence of phonemes followed by decoding the words from the phoneme sequence. Decoding can be done by using the Viterbi algorithm [14] or, for increased efficiency, using beam search.

Still, HMMs have multiple limitations, which restrict their use for ASR. One of the major issues is that HMMs used for ASR are based on the assumption that the state sequences are first-order Markov chains, that is, the states are only dependent on the previous state and conditionally independent of the remaining [9], [37]. However, this restricts the amount of context that is taken into consideration during prediction and language is highly dependent on the context.

2.3 Neural Networks

It did not take long until HMM-based speech recognition models were augmented and gradually replaced by neural network-based approaches. Before dwelling into advanced end-to-end ASR systems in Chapter 3, this and the following sections briefly review some of the fundamental concepts that were pivotal for the development of the sophisticated speech recognition models used today. This section as well as Sections 2.4, and 2.5 are primarily based on the deep learning book by Goodfellow, Bengio, and Courville [15].

Neural networks also referred to as deep feed-forward neural networks, are utilised to approximate a function f^* mapping some input $\mathbf{x}$ to some output $\mathbf{y}$ by learning a set of parameters θ, i.e.,

$$\mathbf{y} = f^*(\mathbf{x}; \theta). \tag{2.4}$$

f^* could for instance work as a classifier, mapping an image to a probability distribution expressing the most likely class that corresponds to the input.

2.3.1 Structure of a Neural Network

A neural network comprises a set of interconnected layers, each consisting of a set of nodes, also referred to as artificial neurons. In a fully-connected neural network,

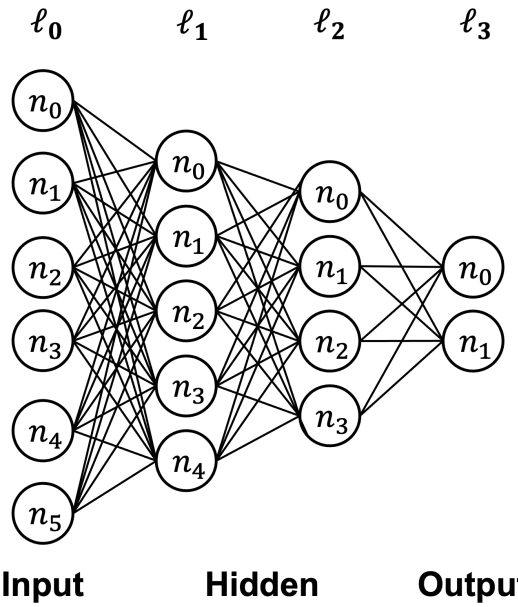


Figure 2.2: Neural network of depth 4 with 2 hidden layers.

each neuron is connected to every neuron in the preceding and succeeding layers. The first and last layers are referred to as the input and output layers respectively. Layers in between the input and output layers are called hidden layers. The total number of interconnected layers determines the depth of the network. Figure 2.2 illustrates a neural network with a depth of 4, where the connections between the nodes are illustrated using edges.

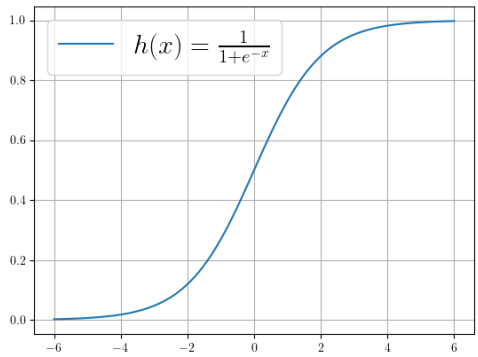
2.3.2 Forward Pass

To compute the output of the network, a forward pass is conducted. During the forward pass, the output of the nodes in each layer is computed in consecutive order, starting with the input layer. The output, or activation, $a_j^{(i)}$ at layer $i \in \{1..\Psi\}$ of node j is computed by

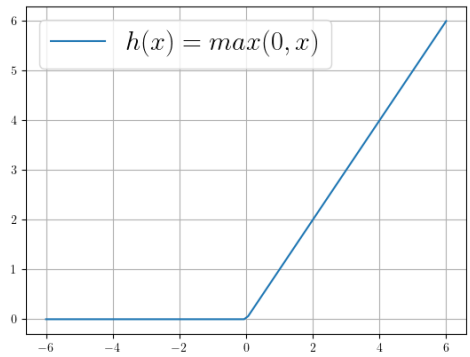
$$a_j^{(i)} = h^{(i)} \left(\sum_{k=1}^{N^{(i-1)}} w_k^{(i)} a_k^{(i-1)} + b_j^{(i)} \right), \tag{2.5}$$

where $N^{(i-1)}$ denotes the number of nodes in the preceding layer $i-1$. The weight $w_k^{(i)}$ determines the overall influence of the activation value $a_k^{(i-1)}$ of node k at layer $i-1$ on the activation of node j at layer i. $b_j^{(i)}$ is the additive bias associated with node j at layer i. Lastly, $h^{(i)}$ is a non-linear activation function of layer i. Activation functions are inspired by the neurons in the human brain as they regulate the information transmitted from one node (neuron) to another [26]. A set of typical activation functions and their respective mathematical definition are given in Figure 2.3.

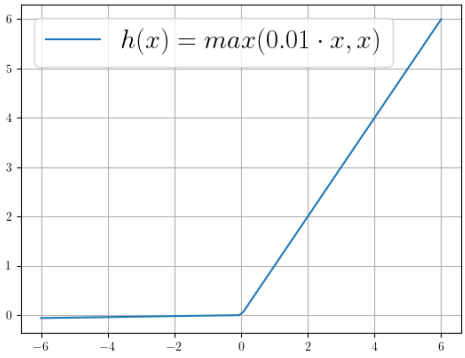
Since each node in a layer of size M is connected to every node in the previous layer of size N, each layer i has a $N \times M$ weight matrix $\mathbf{W}_i$ and a bias vector $\mathbf{b}_i$ of



(a) Sigmoid



(b) ReLU



(c) Leaky ReLU

Figure 2.3: Examples of non-linear activation functions used in neural networks.

size M associated with it. Equation 2.5 can be re-written in matrix form to obtain the activation of a single layer i

$$\mathbf{a}_i = h_i(\mathbf{W}_i^T \mathbf{a}_{i-1} + \mathbf{b}_i). \tag{2.6}$$

2.3.3 Network Loss

The weights and biases of the network are initially set to a random value. However, in order for the network to learn, they need to be adjusted over time. This is done by computing the loss of the network and using its gradient information to update the weights and biases. The loss expresses the amount the output of a network deviates from the desired target values and is computed by a loss function $\mathcal{L}(\theta)$, which computes the deviation of a model prediction y from its target x for all output neurons i using model parameters θ, i.e.,

$$\mathcal{L}(\theta) = \sum_i g(\theta; x_i, y_i), \tag{2.7}$$

where g denotes the function used for computing the loss.

Various loss functions for different types of networks exist. A typical loss function used for classification problems is the softmax cross-entropy loss [16]. These networks

are usually equipped with a softmax layer as the final output layer. In classification problems, the model is used to predict the most likely target label out of a set of possible labels $\mathcal{A}$ that fits best to the input fed to the network. The objective of the softmax is to normalize the output $\mathbf{s} = \{s_1, ..., s_{\mathcal{A}}\}$ of an intermediate layer, such as a linear layer, to a probability distribution over all labels $\mathcal{A}$. This is achieved by computing

$$x_i = \frac{e^{s_i}}{\sum_{j=1}^{\mathcal{A}} e^{s_j}} \tag{2.8}$$

for all output nodes $x_i \in \{x_1, ..., x_{\mathcal{A}}\}$ of the network [10]. Finally, the probabilities computed by the softmax are used to compute the cross-entropy loss of the network

$$\mathcal{L}(\theta) = -\sum_{i=1}^{\mathcal{A}} y_i \log(x_i), \tag{2.9}$$

where $\mathbf{y} = (y_1, ..., y_{\mathcal{A}})$ denotes the target labels [10]. $\mathbf{y}$ is typically expressed as a one-hot vector with all labels being set to 0 except for the target label, which is set to 1 [16].

2.3.4 Back-Propagation

During back-propagation, the gradient of the loss is computed and the weights and biases of each layer $i \in \{1..\Psi\}$ are updated using gradient descent by computing

$$\mathbf{W}_i \leftarrow \mathbf{W}_i - \alpha \frac{\delta \mathcal{L}}{\delta \mathbf{W}_i} \tag{2.10}$$

and

$$\mathbf{b}_i \leftarrow \mathbf{b}_i - \alpha \frac{\delta \mathcal{L}}{\delta \mathbf{b}_i} \tag{2.11}$$

respectively, where α is the learning rate of the network. Note that the input layer has no weights or biases associated with it as its output corresponds to the raw input values at each node. To compute the gradient of the loss with respect to the weights and biases of an arbitrary network layer, the chain rule of calculus is used. Given a function $y = g(x)$ and $z = f(g(x))$, the gradient of z is

$$\frac{\delta z}{\delta x} = \frac{\delta z}{\delta y} \frac{\delta y}{\delta x}. \tag{2.12}$$

Applied to Equation 2.6, the gradient of the loss $\mathcal{L}$ with respect to the weights of layer i $\mathbf{W}_i$ is

$$\frac{\delta \mathcal{L}}{\delta \mathbf{W}_i} = \frac{\delta \mathcal{L}}{\delta h_i} \frac{\delta h_i}{\delta \mathbf{W}_i}. \tag{2.13}$$

and for the bias $\mathbf{b}_i$

$$\frac{\delta \mathcal{L}}{\delta \mathbf{b}_i} = \frac{\delta \mathcal{L}}{\delta h_i} \frac{\delta h_i}{\delta \mathbf{b}_i}. \tag{2.14}$$

2.3.5 Regularisation

Regularisation is concerned with improving the generalisation capabilities of a model, that is, its ability to perform well on unseen data, by preventing it from overfitting on the training data. A simple yet effective regularisation technique is the so-called parameter norm penalty. The idea is to add a penalty $\Omega(\theta)$ to the loss function $\mathcal{L}$, which depends on the parameters θ of the network, typically the weights. The size of the penalty is controlled by the hyperparameter γ. Hence, a regularised loss function $\tilde{\mathcal{L}}$ is defined as

$$\tilde{\mathcal{L}} = \mathcal{L}(\theta; \mathbf{X}, \mathbf{y}) + \gamma\Omega(\theta). \tag{2.15}$$

Two common regularisation approaches are the L^1 and L^2 norm, with the latter also known as weight decay. Let w_i denote the weights of layer i in the neural network. L^2 regularisation is defined as

$$\Omega(\theta)_{L^2} = \frac{1}{2}\sum_i w_i^2. \tag{2.16}$$

It adds a penalty to the total loss of the network based on the square of the weights of the network.

L^1 regularisation, on the other hand, is defined as

$$\Omega(\theta)_{L^1} = \sum_i |w_i|. \tag{2.17}$$

By using the absolute value of the weights of the network, L^1 regularisation adds a greater penalty to the network compared to L^2 regularisation and may ultimately drive a large portion of the weights to 0 as the weights are generally real numbers below 1.0, which causes the square to decrease.

2.4 Recurrent Neural Networks (RNNs)

Recurrent neural networks (RNNs) are neural networks specialised in processing a sequence of input data $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(t)}$ over time t. In addition to the connections between nodes of different layers as in feed-forward neural networks (see Section 2.3.1), RNNs have recurrent connections at the hidden layers. Thus, the activation of a hidden layer ℓ at time $t-1$ serves as part of the input to itself at time t. Figure 2.4 provides an illustrative example. Adding recurrent connections to a neural network allows for parameters to be shared across the model, enabling it to be applied to and generalise across sequences of varying lengths. This is crucial for NLP-related tasks, as the same information can be expressed using different sentence structures of varying lengths.

A forward pass in an RNN is similar to a forward pass in conventional neural networks as defined in Equation 2.6. Layers with recurrent connections have an additional term comprising the weight matrix $\mathbf{V}_i$ associated with the activation $\mathbf{a}_i^{(t-1)}$ of the layer i at time $t-1$. Hence, the activation of a recurrent layer is obtained by

$$\mathbf{a}_i^{(t)} = h_i(\mathbf{W}_i^T\mathbf{a}_{i-1} + \mathbf{V}_i\mathbf{a}_i^{(t-1)} + \mathbf{b}_i). \tag{2.18}$$

Figure 2.5 illustrates the computations performed in a hidden, recurrent layer.

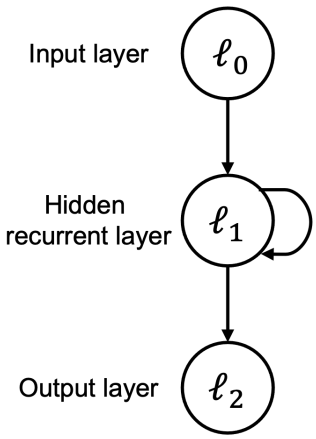


Figure 2.4: Example of a recurrent neural network with a single hidden, recurrent layer.

Training an RNN works exactly the same as in a conventional neural network. The RNN can be unrolled as shown in Figure 2.6, revealing a similar structure to a feed-forward neural network. In order to compute the loss $\mathcal{L}$, the losses for each individual time step $\mathcal{L}^{(t)}$ based on the output sequence $a_\Psi^{(1)}, ..., a_\Psi^{(t)}$ and the target sequence $y^{(1)}, ..., y^{(t)}$ are summed up. Lastly, by applying the back-propagation algorithm introduced in Section 2.3.4, the gradient information can be computed and passed through the network to adjust the weights and biases accordingly.

2.5 Autoencoders

Autoencoders are neural networks comprising an encoder $\mathbf{e} = f(\mathbf{x})$ and a decoder $\mathbf{d} = g(\mathbf{e})$. They are trained to attempt to copy the input $\mathbf{x}$ to the output of the decoder with the purpose of capturing useful features of the data along the way. Thus, $\mathbf{x}$ is both the input and the target at the same time.

During training, input data $\mathbf{x}$ is initially put through the neural network of the encoder, resulting in an encoding $\mathbf{e}$ describing the input. The encoding serves as the input to the decoder, which tries to reconstruct the input $\mathbf{x}$ based on the encoding $\mathbf{e}$. Hence, the goal is to minimise some loss function $\mathcal{L}$ with respect to $g(f(\mathbf{x}))$ and $\mathbf{x}$, i.e.,

$$\mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))). \tag{2.19}$$

The dataflow of an autoencoder is visualised in Figure 2.7.

To prevent the autoencoder from learning to copy the data perfectly, restrictions are applied, such as compression or regularisation. Undercomplete autoencoders have encoders with a smaller dimension than $\mathbf{x}$, forcing them to compress the data and capture the most salient features. Regularised autoencoders, on the other hand, add a penalty Ω to the loss of the network, resulting in the encoder capturing statistical features of the training data.

2.6 Sequence-to-Sequence Models

In speech recognition, the input to the model is typically a sequence of acoustic features and the objective is to map them to a sequence of words transcribing what

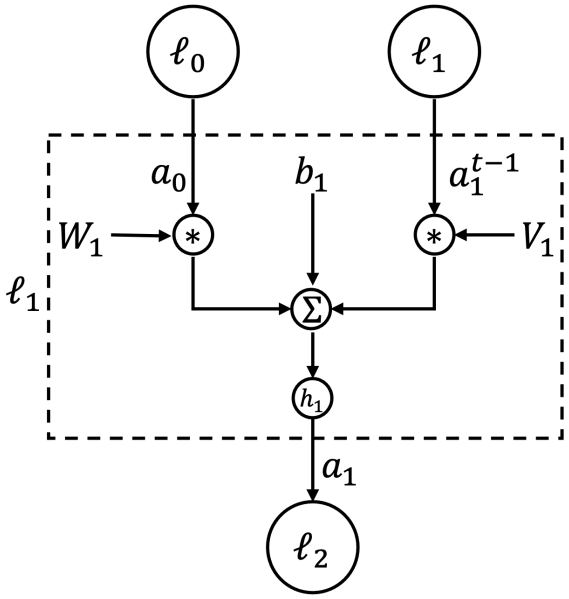


Figure 2.5: Visualisation of the computations performed in a hidden layer ℓ_1 with recurrent connections at time t. The activation from layer ℓ_0 and the activation a_1^{t-1} of layer ℓ_1 at time $t-1$ serve as the input to layer. The resulting output a_1 is forwarded to the next layer. It also serves as the input to the recurrent layer ℓ_1 at time $t+1$ (not visualised).

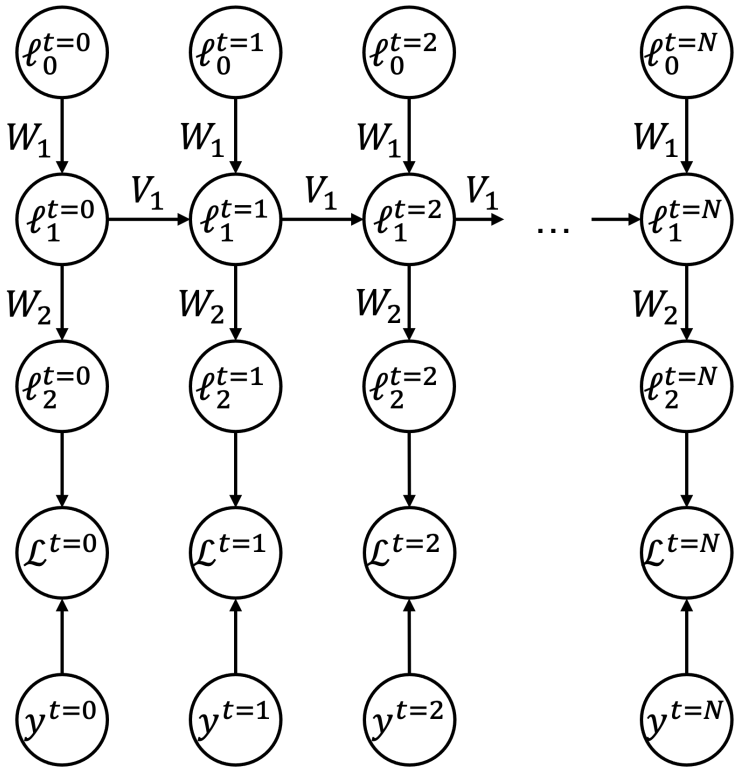


Figure 2.6: Unfolded recurrent neural network with a single hidden layer.

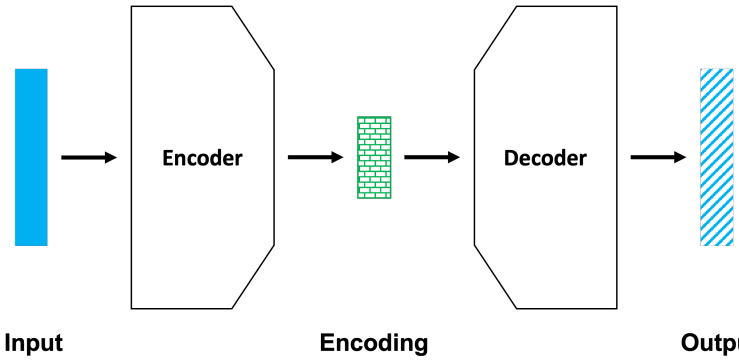


Figure 2.7: Dataflow of an autoencoder. The input is mapped to an encoding using the encoder, while the decoder takes the encoding as input in an attempt to reconstruct the input to the encoder.

has been said. Thus, the model works as a function mapping one sequence to another sequence, where the length of both sequences may be different. Still, processing sequences is rather challenging for conventional neural networks, as they expect the dimensions of the input and output to be known and fixed [36]. Knowing the length of the input and output sequence of an ASR system proves rather difficult, as the lengths can vary substantially. Although RNNs (see Section 2.4) are fully capable of mapping sequences to sequences, a single RNN model cannot be used when the input and output lengths are different [36].

Two approaches for solving the aforementioned issue were proposed by Sutskever, Vinyals, and Le [36] and [12]. Both of the proposed sequence-to-sequence models build on the same idea of using an encoder-decoder architecture (see Section 2.5) in combination with RNNs.

Cho, Merriënboer, Gulcehre, *et al.* [12] proposed an encoder-decoder architecture comprising two RNNs. The encoder RNN reads the input sequence and constructs an encoding based on it. The decoder RNN uses the encoding to generate an output sequence $Y = \{y_1, ..., y_t\}$. Each predicted symbol y_t is also conditioned on the previously predicted symbol y_{t-1}.

The model developed by Sutskever, Vinyals, and Le [36] works in a similar way. However, it uses Long Short-Term Memory (LSTM) [22] modules for the encoder and decoder instead of RNNs due to its performance on learning from data with long-range temporal dependencies.

2.7 Attention

Attention is the ability of humans to focus on certain parts of the information while ignoring the rest of it [24]. A major drawback of sequence-to-sequence models is that important information is lost when compressing sequences of increasing length to a vector of fixed size, leading to a deteriorating performance of the model [5], [24].

2.7.1 Global Attention

In order to address the issue of deteriorating performance, Bahdanau, Cho, and Bengio [5] introduced one of the first attention mechanisms as an extension to the

proposed encoder-decoder architectures. The mechanism also referred to as *global attention* [29], allows the model to include relevant information across the sequence by searching the input sentence for a set of positions with the most relevant information. The traditional encoder-decoder approach proposed by [12] and [36] uses the encoder to compute a fixed-length context vector $\mathbf{c}$ based on a set of hidden states $\mathbf{h}_1, ..., \mathbf{h}_t$ obtained from the input sequence $\mathbf{x}$. The context vector is then used by the decoder to predict the most probable translation y_t. Instead of computing a single context vector, Bahdanau, Cho, and Bengio [5] proposed to compute an individual context vector $\mathbf{c}_i$

$$\mathbf{c}_i = \sum_{j=1}^{t} \alpha_{ij} \mathbf{h}_j \tag{2.20}$$

for each target word y_i. The context vector is obtained by assigning a softmax score α_{ij} to each hidden state $\mathbf{h}_j$ of the RNN encoder, expressing how relevant the inputs at position j of the sequence are for position i. Thus, the attention is shifted towards the most relevant parts of the input sequence.

2.7.2 Local Attention

Luong, Pham, and Manning [29] introduced the notion of global and local attention. Global attention considers all words in the input sequence and is essentially similar to the approach proposed by Bahdanau, Cho, and Bengio [5], but less computationally expensive. Still, global attention requires a lot of resources, especially for longer sentences, leading to the introduction of local attention. Local attention only attends a small subset of the input sequence for each target word y_i by placing a window of size $2D$ at a generated position p_t within the input sequence, where D is an empirically determined parameter. Based on the words occurring within the window, a context vector $\mathbf{c}_t$ is generated based on the hidden states of the encoder. The context vector is thereafter used to predict the target word y_i.

2.7.3 Self-Attention

Self-attention was proposed by Lin, Feng, Santos, *et al.* [27]. It enables the extraction of relevant aspects from a sentence by allowing the sentence to attend itself [24]. Given a set of hidden states $\mathbf{H} = (\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n})$, the idea is to compute an attention vector $\mathbf{a}$

$$\begin{aligned} \mathbf{x} &= \mathbf{v} \cdot tanh(\mathbf{W}\mathbf{H}^T) \\ \mathbf{a} &= softmax(\mathbf{x}), \end{aligned} \tag{2.21}$$

comprising a set of n weights. The softmax function scales $\mathbf{x}$ to values in the range of $[0, 1]$ and ensures that they sum up to 1. It is defined as

$$softmax(\mathbf{x}) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \; \forall x_i \in \mathbf{x} = (x_1, ..., x_k). \tag{2.22}$$

Thus, the attention vector determines which of the hidden states to focus on, that is, how related two states are to each other. Computing the dot product between the hidden states $\mathbf{H}$ and the attention vector results in an output vector $\mathbf{m}$, which contains the combined, weighted information from all hidden states. However, in order to capture the overall semantics of the sentence, multiple output vectors are

required. Hence, Equation 2.21 is extended to compute a matrix $\mathbf{M}$ of output vectors $\mathbf{m}$, i.e.,

$$\begin{aligned} \mathbf{A} &= softmax(\mathbf{V} \cdot tanh(\mathbf{W}\mathbf{H}^T)) \\ \mathbf{M} &= \mathbf{A}\mathbf{H}. \end{aligned} \tag{2.23}$$

2.8 Transformers

Transformers were initially proposed by Vaswani, Shazeer, Parmar, *et al.* [38] as a pure attention-based approach to neural machine translation. The model adapts a common encoder-decoder approach, which maps an input sequence $(x_1, ..., x_N)$ to an encoding $\mathbf{z} = (z_1, ..., z_N)$ and uses the encoding to generate an output sequence $(y_1, ..., y_M)$. It is important to note that the length of the input and output sequence can be different. However, in contrast to the traditional sequence-to-sequence approaches to machine translation, such as [12] and [36], the Transformer architecture does not make use of any recurrent or convolutional layers.

2.8.1 Multi-Head Self-Attention

An important concept of the Transformer architecture is the multi-head self-attention layer, which works in a similar way as introduced in Chapter 2.7.3. Given an input sequence $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})$ of length n and dimension m, a key and query vector of dimension d_k and a value vector of dimension d_v are computed for every $\mathbf{x_i}$ with $\{i \in \mathbb{N} | 1 \leq i \leq n\}$ using a linear layer. The value vectors are essentially a representation of the input sequence, while the key and query vectors are used to compute the attention vector with values in the range of $[0, 1]$ that sum up to 1. Similar to the approach discussed in Section 2.7.3, the attention vector represents a set of weights, which are used to compute an output vector by calculating the dot product between the attention and value vectors. Consequently, the output matrix $\mathbf{O}$ for an entire sequence is obtained by

$$\mathbf{O} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}. \tag{2.24}$$

Figure 2.8 visualises the operations performed for a single input sequence $\mathbf{X}$.

Instead of computing the attention once, the multi-head self-attention approach constructs h different sets of queries, keys, and values using h linear projections, each using different weights. Hence, for a single so-called head i the output O_i is computed by

$$\begin{aligned} K_i &= XW_K^i \\ Q_i &= XW_Q^i \\ V_i &= XW_V^i \\ O_i &= softmax\left(\frac{\mathbf{Q_i}\mathbf{K_i}^T}{\sqrt{d_k}}\right)\mathbf{V_i}. \end{aligned} \tag{2.25}$$

Finally, the output of all heads is concatenated and projected once more, resulting in the final output $O_{multi-head}$ of the multi-head self-attention layer.

$$O_{multi-head} = \text{Concat}(O_1, O_2, ..., O_h)W_{multi-head} \tag{2.26}$$

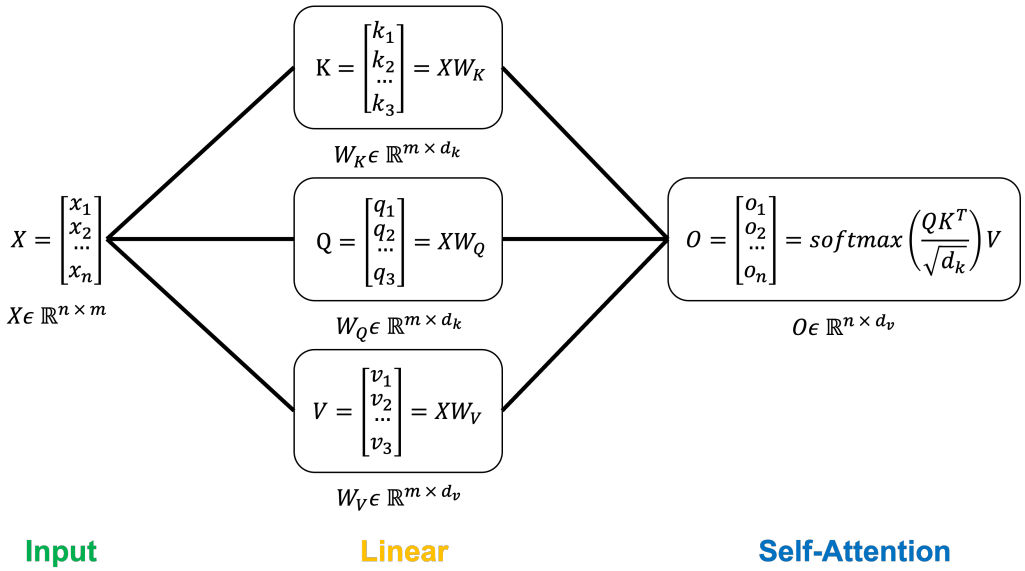


Figure 2.8: Self-Attention in a Transformer takes the input sequence **X** and computes the key (K), query (Q), and value (V) matrices using a linear layer. Lastly, the output matrix O is obtained by computing a softmax using K and Q and computing the dot product with V. The matrix resulting from the softmax determines how much each value vector in V contributes to each output vector in O.

2.8.2 Transformer Architecture

The Transformer uses an encoder-decoder architecture, with both the encoder and decoder using a stack of six identical layers. A single block in the encoder consists of two layers, that is, a multi-head self-attention followed by a fully connected feed-forward network. Each layer is followed by layer normalisation [3] and an additional residual connection [20] around each layer. The decoder uses a similar architecture with the addition of a second multi-attention layer after the first layer performing multi-head self-attention on the output of the encoder. In addition, the first multi-head self-attention layer is masked to prevent the model from attending positions in the input sequence that are beyond the current position. Lastly, the final block is followed by a linear layer and a softmax to compute an output probability distribution. Figure 2.9 visualises the architecture of the encoder and decoder used in the Transformer.

2.9 Summary

This chapter reviewed some of the fundamental building blocks of modern speech recognition systems. Section 2.1 started off by summarising how audio signals are converted to digital representations using analogue-to-digital converters and how the short-time Fourier transform is used to compute a spectrogram, a frequently used audio representation in modern ASR systems. In Section 2.2, an introduction to the classic, probabilistic approach to speech recognition using HMMs was given. It highlighted their limited use due to the conditional independence assumption, which ultimately led to their retirement with the advancements made in neural network-based speech recognition. Section 2.3 discussed the fundamentals and important

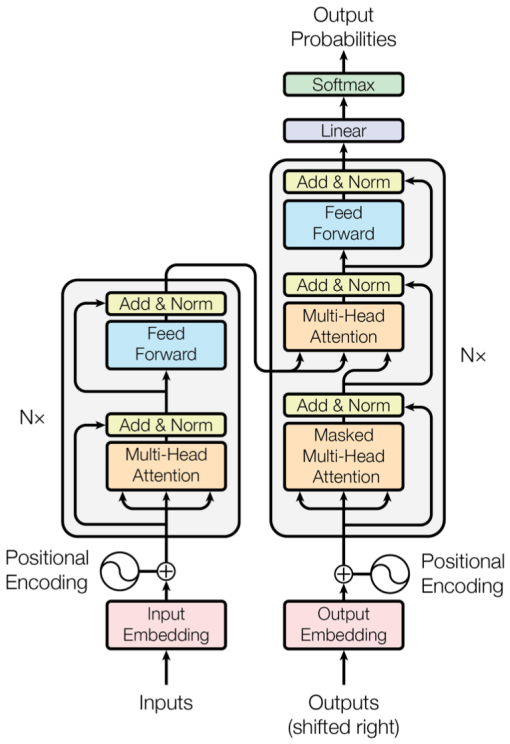


Figure 2.9: A Transformer comprises an encoder and a decoder, each consisting of a set of consequential, identical blocks. A single block in the encoder contains multi-head attention and a feed-forward layer. The decoder blocks use the same architecture, with the addition of a masked multi-head attention layer to prevent the model from attending parts of the input sequence that are beyond the current sequence position. (Source: [38])

concepts of neural networks, including the forward pass (Section 2.3.2), computing the loss of a network (Section 2.3.3), and how the network weights and biases are updated using the back-propagation algorithm (Section 2.3.4). Lastly, Section 2.3.5 introduced L^1 and L^2 regularisation, which are two commonly used regularisation methods for improving the generalisation capabilities of a neural network. Sections 2.4 and 2.5 provided a brief introduction to recurrent neural networks and autoencoders, two neural network architectures that are fundamental to the development of the sequence-to-sequence models discussed in Section 2.6. This was followed by a discussion on the attention mechanism in Section 2.7, a key component for modern speech recognition architectures that enables the model to focus on different parts of the input. It introduced various attention mechanisms, including global, local, and self-attention, and discussed some of the differences between them. Finally, building on the previous chapters on sequence-to-sequence models and the attention mechanism, Section 2.8 provided an introduction to Transformers [38], a modern, attention-based encoder-decoder architecture that is the backbone of Whisper [31]. It discussed the concept of multi-head self-attention, which enables the model to focus on different parts of the input sequence and provided an overview of the different building blocks of the Transformer architecture.

Chapter 3

Related Work

The purpose of this chapter is to provide an overview of the state-of-the-art related to this thesis with a particular emphasis on Whisper [31]. Section 3.1 briefly summarises hybrid approaches that were used before the introduction of Connectionist Temporal Classification (CTC) [17], which is discussed in Section 3.2. Sections 3.3 and 3.4 describe two of the earliest deep learning-based ASR approaches, which made use of the CTC approach. This is followed by a summary of the Wav2Vec 2.0 [4] and Whisper [31] architectures in Sections 3.5 and 3.6 respectively.

3.1 Hybrid Models using Neural Networks and HMMs

Chapter 2.2 provided a short introduction to traditional, probabilistic approaches to automatic speech recognition using HMMs. Still, due to the limited real-world applicability of HMMs, different paradigms were explored. While the sole application of neural networks for speech recognition was considered, the research concluded that neural networks on their own are not suited for ASR due to the inability to model long-term dependencies [37]. Instead, hybrid architectures combining HMMs with neural networks were developed in the late 1980s and early 1990 in an attempt to improve the flexibility and performance of ASR systems [37].

Various architectures have been proposed. Initial efforts attempted to emulate HMMs with neural networks, while more advanced approaches focused on delegating parts of the speech recognition pipeline to neural networks, such as estimating the transition probability $P(s_t|s_{t-1})$ [37]. Other approaches focused on adding neural networks on top of HMMs to obtain a more suitable representation of the acoustic features and to allow neural networks and HMMs to be trained jointly instead of training them separately [37].

For a more thorough overview of hybrid models, the reader is referred to the excellent survey on hybrid ASR systems by Trentin and Gori [37].

3.2 Connectionist Temporal Classification (CTC)

Connectionist Temporal Classification (CTC) [17] was introduced as a new approach to labelling unsegmented sequence data, which, in the context of ASR, refers to the assignment of phonemes or letters to the corresponding sequence of acoustic features. CTC played an important role in the development of neural network-based ASR

approaches as RNNs were only capable of making independent label classifications. Consequently, training data had to be pre-segmented and post-processed to achieve the desired label sequence.

The idea of the CTC approach is to interpret the output of the RNN as a probability distribution over all possible labels Z, given an input sequence S of length T. More specifically, the RNN outputs a probability over all the labels $z_i \in Z$ for every $s_t \in S$ with $1 \leq t \leq T$. An illustrative example of the resulting three-dimensional probability distribution is given in Figure 3.1.

Since different acoustic features can map to the same label, the resulting sequence of labels may include multiple repetitions of the same label. However, they cannot be simply combined into a single instance of the label, as some words may contain repetitions of the same label, such as *"hello"*. To be able to differentiate between repeating labels and to detect pauses between them, a *blank* label was introduced in CTC, resulting in $Z' = Z \cup \{blank\}$.

Using the probability distribution over Z' $\forall s_t \in S$, the total probability for a single alignment π, also referred to as a path, can be computed by

$$p(\pi|S) = \prod_{t=1}^{T} y_{\pi_t}^t, \tag{3.1}$$

where $y_{\pi_t}^t$ is the probability of observing label π_t at time t. To compute the probability of a single labelling, repeated labels and blank labels are first removed from all paths, resulting in a set of edited paths π^*. Since a single sequence of labels z may have multiple paths, the probability of z can be computed by

$$p(z|S) = \sum p(\pi^*|S). \tag{3.2}$$

Finally, the most likely labelling, i.e., the output of the temporal classifier, corresponds to the most likely labelling $z \in Z$ for a given input sequence S

$$h(S) = \underset{z \in Z}{\operatorname{argmax}}\, p(z|S). \tag{3.3}$$

The classifier is then trained by minimising the CTC loss function

$$\mathcal{L} = -\sum_{(s_t,z)\in S} \ln(p(z|s_t)), \tag{3.4}$$

which maximises the log-likelihood of the correct classifications.

3.3 Deep Speech

Deep Speech [19] was one of the first end-to-end deep learning-based ASR approaches. It outperformed traditional speech recognition approaches in both conversational speech and speech in noisy environments without the need for engineering domain-specific processing pipelines. Deep Speech employs an RNN comprising five hidden layers. The first three layers are conventional feed-forward without any recurrent connections, while the fourth layer is a bi-directional recurrent layer [32] computing a forward recurrence $\mathbf{h}_f$ and backward recurrence $\mathbf{h}_b$ for each time t. The forward

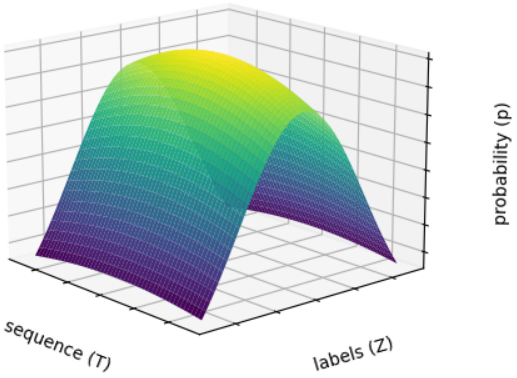


Figure 3.1: Using the CTC approach, the output of an RNN is interpreted as a probability distribution over all the labels Z for a particular instance s_t of a sequence S of length T. Plotting the probability distribution for all $s_t \in S$ results in a three-dimensional probability distribution as illustrated in this figure.

and backward recurrence values are then forwarded to the fifth, non-recurrent feed-forward layer, whose activation value is based on the sum of the backward and forward recurrence, i.e.,

$$\mathbf{a}_5 = h_5(\mathbf{W}_5^T \ (\mathbf{h}_f + \mathbf{h}_b) + \mathbf{b}_5). \tag{3.5}$$

Finally, a softmax layer computes the likelihood $p(c_t = k|\mathbf{x})$ for each character k in the English alphabet and time-step t. The model architecture used in Deep Speech is depicted in Figure 3.2. In order to train the network, Deep Speech utilises the CTC objective function to handle the challenge of aligning text transcripts with the audio input.

Deep Speech achieved a lower WER on the Hub5'00 test dataset [28] compared to a range of traditional, hybrid architectures. It also performed significantly better on noisy speech compared to commercial speech recognition systems from Google and Apple.

3.4 Deep Speech 2

Building on the success of Deep Speech [19], Amodei, Ananthanarayanan, Anubhai, *et al.* [2] proposed Deep Speech 2, which achieved a significant performance boost on the English language and a 7× speedup compared to the first model. What is more, the model was extended with the ability to transcribe Mandarin with great accuracy. Deep Speech 2 uses a similar core architecture as Deep Speech, consisting of a set of convolutional, recurrent, and fully-connected layers. However, the authors experimented with different model depths and layer types, constructing multiple model architectures.

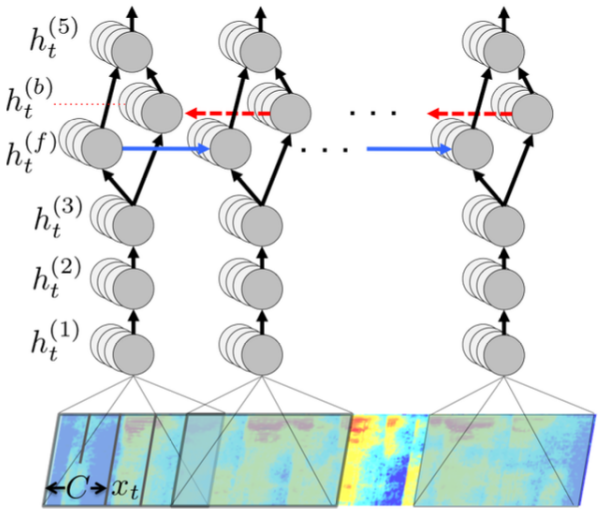


Figure 3.2: Deep Speech uses a relatively simple model architecture comprising four non-recurrent linear layers and a single bi-directional recurrent layer. The audio is converted to a spectrogram and fed to the model, which outputs a probability distribution over the English alphabet. (Source: [19])

For English, the best-performing model uses 11 layers, consisting of 3 2D convolution layers followed by 7 recurrent layers and a fully-connected layer with batch normalisation. It achieved a WER of 13.59 compared to a WER of 24.01 scored by Deep Speech on the internal Baidu dataset consisting of various accented, noisy, spontaneous, and conversational speeches. Moreover, it scored WERs close to human performance on read speech and achieved substantial improvements on both accented and noisy speech compared to the original Deep Speech model.

The best-performing model in Mandarin uses batch normalisation and consists of 9 layers, that is, a single 2D convolutional, 7 recurrent, and one fully-connected layer. The model scored a WER of 7.93 on the test set of the internal Baidu dataset.

3.5 Wav2Vec 2.0

Wav2Vec 2.0 [4] is a speech recognition architecture that uses self-supervised learning to combat the requirement for large amounts of annotated audio data. This is achieved by training the model on large amounts of unlabelled data followed by CTC-based fine-tuning on labelled data. The raw audio waveform is processed by a multi-layer convolutional neural network (CNN) encoder to generate a sequence of latent representations $\mathbf{x_1}, ..., \mathbf{x_T}$ for T time-steps. Parts of the latent representations are masked before being fed to a Transformer, which constructs context representations $\mathbf{c_1}, ..., \mathbf{c_T}$ based on the entire sequence. The latent representation is also used by the quantisation module, which discretises the output of the encoder creating a set of quantised representations. These are then used during self-supervised training, where the objective is to learn to identify the true latent representation that has previously been masked and distinguish it from a set of distractors. Finally, the model is fine-tuned using labelled data by adding a set of linear layers, which project the context representations to the labels, and computing the CTC loss. Figure 3.3 illustrates the model architecture of Wav2Vec.

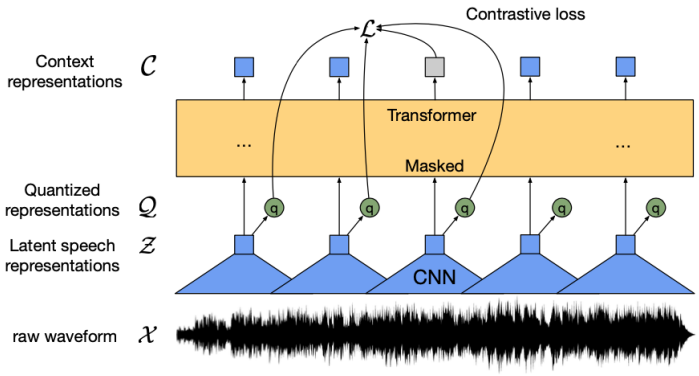


Figure 3.3: Wav2Vec 2.0 processes raw audio waveform data using a convolutional neural network to compute a latent representation $\mathbf{Z}$. The latent representation is then used to compute both a set of quantised representations and context representations. (Source: [4])

Wav2Vec 2.0 achieved a WER of 1.8 and 3.3 on the test/clean and test/other splits of the LibriSpeech [30] dataset, which is significantly lower than the WER of 5.33 and 13.25 scored by Deep Speech 2 [2]. While the original Wav2Vec 2.0 model has been trained on the English language, additional models for other languages have been developed, including for Norwegian Bokmål and Nynorsk. Hence, the respective Wav2Vec 2.0 models are going to be used as baseline models in Chapter 5.

3.6 Whisper

Whisper [31] is a large-scale speech recognition model trained on 680,000 hours of annotated audio data, including 117,000 hours of audio covering 96 languages other than English for multilingual speech recognition. The model also features a translation mechanism that allows for the translation of foreign languages to English. Moreover, it is also quite robust to real-world noise as shown in Figure 3.4. Whisper has a stable performance if exposed to both white noise and real-world noise at SNRs greater than 20 dB and it outperforms other state-of-the-art approaches in the case of high noise with SNR of 10 dB or less.

The model uses an adapted, off-the-shelf encoder-decoder Transformer architecture [38] with a similar architecture to the Transformer discussed in Chapter 2.8. Figure 3.5 provides an overview of Whisper's architecture. Input audio needs to be downsampled to 16,000 Hz and converted to a log-mel spectrogram (see Chapter 2.1.3) to be processed by the model. The spectrogram is then further processed by the encoder, which contains two additional convolutional layers with a GELU [21] activation function. The output of the convolutional layers is then combined with position embeddings and forwarded to the multi-head self-attention blocks to compute the encoding, which serves as an input to the decoder. The task of the decoder is to predict a sequence of tokens based on the input encoding and the previously predicted tokens. The decoder is identical to the decoder architecture of a conventional Transformer. It comprises a set of sequential decoder blocks followed by a final softmax output layer, which produces a probability distribution over a set of token ids. Finally, a tokenizer is used to map the token ids to the corresponding words to get the actual transcription.

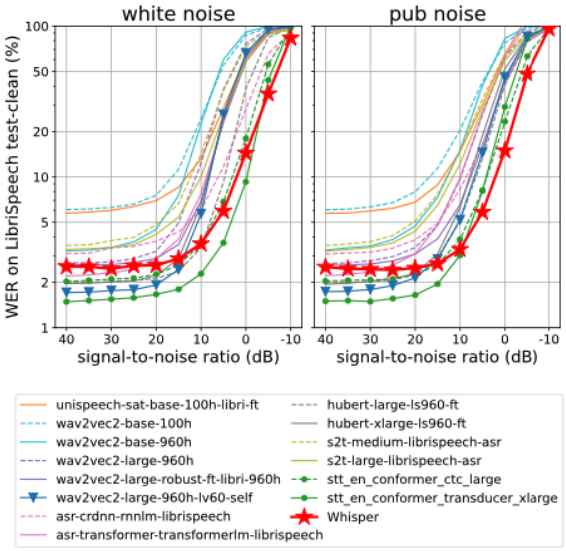


Figure 3.4: Noise performance of Whisper compared to other state-of-the-art speech recognition architectures when exposed to white noise (left) and real-world noise obtained from a crowded pub (right). The WER is based on the LibriSpeech [30] test-clean split. Whisper is outperformed by NVIDIA STT Conformer-CTC models from the NeMo toolkit [18], [25] under low noise conditions, but it scores a lower WER than other architectures for signal-to-noise ratios smaller than 10 dB. (Source: [31])

As mentioned above, Whisper can not only be used for transcription but also for translation. What is more, it is also capable of predicting the language of the given audio signal and also supports the prediction of timestamps for time-aligned transcriptions. To differentiate between the different tasks, a set of special tokens denoting the task of the model was added. For instance, the `<|transcribe|>` and `<|translate|>` tokens are used to specify whether the audio should be transcribed to the spoken language or English. To deactivate the prediction of timestamps, the `<|notimestamps|>` is added. The order of the tokens is visualised in Figure 3.6.

Whisper ships in five different model sizes, each with a different number of parameters and layers. The smallest model uses 4 layers for both the encoder and decoder with 6 attention heads each and a total of 39M parameters, while the largest model uses 32 layers with 20 attention heads and a total of 1550M parameters. An overview of the model sizes and their respective WER on Norwegian Bokmål can be found in Table 3.1. Overall, the larger the model the better the transcription performance. Compared to the smallest model, which scored a WER of 62.0 on the Fleurs [13] dataset, the largest model performs significantly better, obtaining a WER of 9.5. Nevertheless, a greater number of parameters also leads to a significant increase in inference time, which can be a crucial factor that needs to be taken into consideration especially if the model cannot be run on a GPU.

3.7 Summary

This chapter started by briefly reviewing early attempts to use neural networks for speech recognition in combination with HMMs in so-called hybrid approaches

[1]https://openai.com/research/whisper

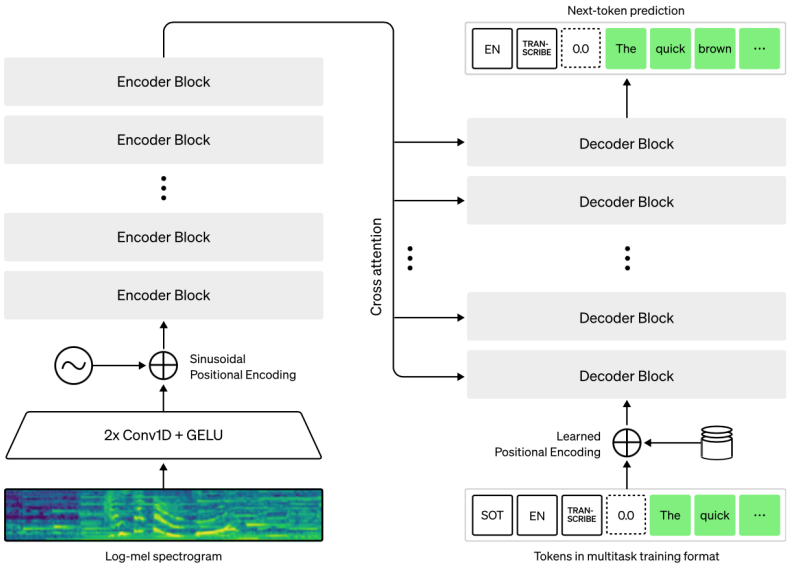


Figure 3.5: Whisper uses an off-the-shelf Transformer architecture with a slightly modified encoder. Since the model input is audio and not text, the audio needs to be converted to a log-mel spectrogram, which is then processed by the added convolutional layers with a GELU [21] activation function. The remaining architecture is identical to the original Transformer. (Source:[1])

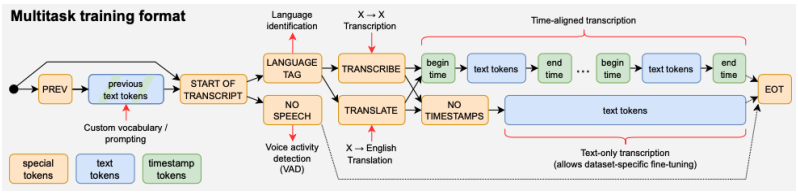


Figure 3.6: Whisper adds a set of special tokens before the actual text tokens to specify what type of language the model should use and if it should be transcribed using the spoken language or translated to English. (Source: [31])

Table 3.1: Number of layers, layer dimensionality, number of attention heads, parameters, and the corresponding WER on Norwegian Bokmål of the different model sizes offered by Whisper. The WER is based on the Fleurs [13] dataset. In general, the WER is considerably lower for the larger models, but the inference time is also noticeably higher. (Source: [31])

Model	Layers	Dimensionality	Heads	Parameters	WER Bokmål
Tiny	4	384	6	39M	62.0
Base	6	512	8	74M	44.0
Small	12	768	12	244M	24.2
Medium	24	1024	16	769M	12.9
Large-v2	32	1280	20	1550M	9.5

in Section 3.1, which emerged primarily due to the conclusion that pure neural network-based ASR approaches were infeasible at that time. With the introduction of the CTC [17] approach covered in Section 3.2, pure end-to-end speech recognition approaches were becoming more feasible as it solved the key issue of aligning acoustic signals with the correct target labels, such as phonemes or letters, and distinguishing between repeating labels. The alignment issue was solved by interpreting the output of an RNN as a probability distribution over all labels and by introducing a $\{blank\}$ token, which allows differentiating between repetitions of the same label. CTC was fundamental to the development of Deep Speech [19] and its successor Deep Speech 2 [2], which are discussed in Sections 3.3 and 3.4. Deep Speech was one of the first deep learning-based ASR systems. It not only outperformed other state-of-the-art approaches in terms of classification performance but also eliminated the need for constructing cumbersome, domain-specific processing pipelines, which was necessary when using HMM-based and hybrid approaches. Deep Speech 2 further improved the overall system performance and extended the model with support for Mandarin, showing that the model can easily be extended with support for languages other than English. Finally, Sections 3.5 and 3.6 introduced Wav2Vec 2.0 [4] and Whisper [31], which are two state-of-the-art speech recognition utilising the Transformer [38] architecture discussed in Chapter 2.8. Both models achieved considerably lower WERs compared to other approaches. However, while Whisper comes with built-in support for 96 different languages for multilingual speech recognition, the original Wav2Vec 2.0 model only supports English and has to be trained in other languages first. Moreover, Whisper is also less susceptible to high noise than other state-of-the-art models, including various Wav2Vec 2.0 models, and has stable performance in the presence of low noise.

Due to the multilingual support, noise robustness, and low WERs across multiple languages, Whisper was picked as a candidate platform for replacing the current speech recognition technology used in the Furhat robot to mitigate the issues discussed in Chapter 1.1. Since the system should ideally have both a low WER and a short inference time, the medium-sized Whisper model is used for further analysis. It scored a relatively low WER of 12.9 on Norwegian Bokmål on the Fleurs [13] dataset and the inference time was significantly lower than the largest model during initial testing. However, the WER on the validation set of the dataset used for testing the baseline model was not as low as on the Fleurs dataset. Whisper was only trained on 266 hours of Norwegian Bokmål for speech recognition, which is considerably less than the $23,446$ hours in Chinese or $438,218$ hours in English the model was trained on [31]. To improve the performance on low-resource languages such as Norwegian, the authors of Whisper suggested fine-tuning the model on a high-quality dataset [31]. Hence, prior to evaluating Whisper with respect to the primary issues of the current system, the model is fine-tuned on the Norwegian Parliament Speech Corpus (NPSC) [35] dataset in an effort to improve the overall WER. The dataset as well as the methodology for fine-tuning the model are discussed in detail in the following chapter.

Chapter 4

Methodology

With the introduction of Whisper [31] in the previous chapter, the goal of this chapter is to provide an overview of the methodology used to fine-tune and evaluate Whisper on the ability to transcribe spoken Norwegian to Bokmål and Nynorsk, which are the official written languages in Norway. The first section starts by presenting a high-level summary of the used approach with reference to the research questions guiding this thesis. This is followed by a detailed description of the Norwegian Parliament Speech Corpus (NPSC) [35] dataset in Section 4.2, which is the dataset used for fine-tuning and evaluating Whisper. Section 4.3 continues by describing the data pre-processing steps to prepare the data for model training. This is followed by an introduction to the word error rate (WER) in Section 4.4, which is the primary metric used in this thesis to compare the performance of different models, and a description of the training procedure of the model in Section 4.5.

4.1 Overview of the Approach

As discussed in Chapter 1.1, the current speech recognition system used in the Furhat robot at NorwAI is not ideal. It struggles with a range of Norwegian dialects and is very susceptible to background noise. Moreover, it also performs poorly on names and abbreviations and has no built-in support for speech recognition in Nynorsk. To make the interaction more natural and less prone to error, the objective of this thesis is to analyse the medium-sized Whisper model in detail to determine if it performs better than the current system.

The primary reason for using Whisper instead of any other speech recognition model is its capability to transcribe 96 different languages, including spoken Norwegian to Bokmål [31]. However, even though Whisper was not trained on Nynorsk for multilingual speech recognition, it was trained on 1889 hours of data for translation purposes [31]. Hence, it is assumed that fine-tuning Whisper with a high-quality Nynorsk dataset could enable Whisper to transcribe to Nynorsk. What is more, Whisper also achieved relatively low WERs on the LibriSpeech [30] dataset with the medium-sized model scoring a WER of 12.9. However, initial test results of running the model on the NPSC validation split showed that the WER was not as low as on the LibriSpeech dataset. Thus, the medium-sized model is fine-tuned on the NPSC dataset to determine if the performance can be improved, which was suggested by the authors of Whisper for low-resource languages, such as Norwegian [31].

To fine-tune the model, the unmodified medium-sized Whisper model was trained

on the training split of the NPSC dataset and evaluated using the validation split. Prior to training the model, the data was pre-processed to remove any special tokens and characters introduced by the dataset and the audio was downsampled and prepared as required by the model. The model was then trained multiple times using different hyperparameters to find the best-performing model.

The fine-tuned model is evaluated by running it on the test split of the NPSC dataset and computing the mean WER (see Section 4.4). The overall WER is then compared to the WER achieved by the default Whisper model to determine if fine-tuning improved the overall performance on Nynorsk (see research question **RQ1**) and Bokmål (see research question **RQ2**). This is followed by a detailed analysis of the WER regarding the age, gender, and dialect of the speaker (see research question **RQ3**) and how well Whisper transcribes sentences with names and abbreviations (see research question **RQ5**), as these are two of the key issues with the current system. Lastly, to evaluate the performance in the presence of noise (see research question **RQ4**), the fine-tuned model is run on a noise-augmented version of the NPSC test split multiple times using different signal-to-noise ratios.

4.2 The Norwegian Parliament Speech Corpus (NPSC)

An ideal dataset that is suitable for fine-tuning needs to meet several criteria. The Norwegian language does not have a standardised spoken language and a wide variety of dialects exist [39]. While the dialects can, in general, be split into four main groups, that is, Eastern (østnorsk), Western Norwegian (vestnorsk), Northern Norwegian (nordnorsk), and Trøndersk, differences exist even within these groups [39]. To capture the variations of the Norwegian language, the dataset should ideally cover speakers from different parts of the country. Another important consideration is the year the dataset was released in. The publication date of the dataset should be after the year the Whisper model was released to minimise the risk that the baseline model has been exposed to the data during training. Thirdly, the dataset should contain annotations for both official written languages, i.e., Bokmål and Nynorsk. Lastly, the dataset should contain sufficient training data, as the amount of available data impact model performance.

In the end, the Norwegian Parliament Speech Corpus (NPSC) [35] was selected for evaluating and fine-tuning Whisper. It is an open-source ASR dataset developed from 2019 to 2021 with more than 140 hours of data obtained from speeches given at Stortinget, the Norwegian parliament, from 2017 and 2018. It covers a wide range of dialects from Western, Eastern, Southern, and Northern Norway as well as Trøndelag. It also includes annotations for both Bokmål and Nynorsk performed by trained linguists and philologists.

The following sub-sections are primarily based on information retrieved from the dataset description file, which can be found on the website of the Norwegian Language Bank [1].

[1] https://www.nb.no/sbfil/talegjenkjenning/npsc/v1_1/NPSC_doc_1_1.pdf

4.2.1 Annotation Process

To generate the transcriptions for the parliament speeches, an automatic transcription of Bokmål was generated using Google Cloud Speech-to-Text, followed by a comparison with the proceedings file. In case of high similarity, non-matching words in the machine-generated transcription were replaced by the corresponding words from the proceedings file to improve the transcription. The automatic transcription was reviewed and corrected manually by a transcriber and co-reviewed by another staff member at the Norwegian Language Bank[2].

Speeches were transcribed either to Bokmål or Nynorsk based on the written language preferred by the member of parliament. If the preferred language is Nynorsk, the machine translation would replace some of the words with a corresponding word in Nynorsk. Incorrect transcriptions were corrected in the subsequent manual review.

Since the majority of the Norwegian population uses Bokmål as their written language, only 12% of the annotations in the NPSC are in Nynorsk. To have the same number of sentences annotated in Bokmål and Nynorsk, sentences annotated in Bokmål were machine translated to Nynorsk and sentences written in Nynorsk to Bokmål. The translations were performed using the rule-based translation system Apertium[3].

4.2.2 Data Normalisation

The manual transcriptions were performed in the spoken domain, that is, numbers, years, and dates are annotated using letters instead of actual numbers and abbreviations are not used. An additional normalised transcription is provided in the dataset, where numbers, years, and dates are expressed in the written domain using digits and standardised date formats.

4.2.3 Special Tokens

The datasets define a set of special tokens that are used for transcribing hesitations and inaudible or overlapping parts of the speech. Hesitations are transcribed as `<ee>`, `<mm>` or `<qq>`, corresponding to vocalic, nasal, and non-linguistic hesitations respectively. Inaudible or overlapping parts of the speech are marked with the token `<INAUDIBLE>`. What is more, words not recognised by the Apertium translation system used for the Bokmål and Nynorsk translations are marked with a Kleene star $*$.

4.2.4 Dataset Splits

The dataset is split into a training, evaluation, and test set using an $80 - 10 - 10$ split, i.e., the training set contains 80% of the data while the evaluation and test set comprises 10% of the data respectively. The percentage of female speakers, average word length per sentence, and percentage of manual transcriptions in Nynorsk are similar across each split. However, the splits have slightly different dialect distributions. The exact split statistics are given in Tables 4.1 and 4.2.

[2]https://www.nb.no/sprakbanken/en/sprakbanken/

[3]https://apertium.org

Table 4.1: Dataset statistics showing the overall size in hours of the dataset, the respective percentage of female speakers and Nynorsk, and the average sentence length in words across the different dataset splits.

Split	Duration (hrs)	Nynorsk (%)	Female (%)	Avg. sentence len. (words)
Train	100.3 hrs	12.8 %	37.7 %	18.7
Eval	13.1 hrs	12.7 %	41.8 %	18
Test	12.3 hrs	13 %	39.9 %	18

Table 4.2: Dialect distribution across the dataset splits.

Split	Western N. (%)	Eastern N. (%)	Southern N. (%)	Northern N. (%)	Trøndelag (%)
Train	26.5 %	46.1 %	6.5 %	11.9 %	9.1 %
Eval	32.2 %	43.6 %	7.6 %	7.7 %	8.8 %
Test	35.0 %	44.4 %	4.6 %	9.4 %	6.5 %

4.2.5 Dataset Format

As already mentioned above, the dataset contains transcriptions obtained from plenary meetings at the Norwegian parliament from 2017 and 2018. The transcriptions cover the entire meeting day. Each transcription is, however, limited to six hours and ten minutes and the audio recordings are cut off if the meeting exceeded the limit.

The dataset is split into a set of folders named after the date the plenary meeting took place using the format *yyyymmdd*. Each folder contains five files named using the date and start time of the meeting and the format *yyyymmdd-hhmmss*. The content of each file is described in Table 4.3. Only the *yyyymmdd-hhmmss_sentence_data.json* was used during training. The features relevant to this thesis are described in further detail in Table 4.4. An overview of all features can be found in Tables B.2 and B.3.

Apart from the aforementioned files, each dataset folder also contains an audio folder with the corresponding audio file for each sentence. Files are named after the meeting date, start time of the session, as well as the start and end time in milliseconds of the sentence using the format *yyyymmdd-hhmmss_starttime_endtime*. All audio files are wav files and were sampled at a sampling rate of 48 kHz. A detailed overview of the technical specifications can be found in Table B.1 in the appendix.

Lastly, the dataset also contains a project files directory, which contains transcription guidelines, postprocessing scripts, a Sqlite database containing the transcriptions, and a *NPSC_speaker_data.json* file comprising metadata about the speakers. Only the *NPSC_speaker_data.json* file was kept as it is required to analyse the WER with regard to the dialect, age, and gender of the speaker by matching the speaker id of the sentences with the corresponding id in the metadata file. An overview of the relevant features from the speaker data file can be found in Table 4.5, while the complete list of features is given in Table B.4.

Table 4.3: Description of the files contained in each folder within the NPSC dataset.

Filename	Filetype	Description
yyyymmdd-hhmmss.ref	Text	Official proceedings file from Stortinget covering the entire meeting, even if the meeting exceeded the set limit of six hours and ten minutes.
yyyymmdd-hhmmss.wav	Audio	Audio file covering the entire meeting or the first six hours and ten minutes if the meeting exceeds the set limit.
yyyymmdd_sentence_data.json	JSON	JSON file containing the corrected and normalised audio transcriptions, machine-generated translations, and additional metadata, including to which split the folder belongs.
yyyymmdd_token_data.json	JSON	Contains word-tokenised transcriptions of the sentences and additional metadata about the word.
yyyymmdd_normalized_token_data.json	JSON	Contains normalised word-tokenised transcriptions of the sentences.

Table 4.4: Overview of the dataset features relevant to this thesis.

Feature	Description
speaker_id	ID of the speaker.
data_split	Specifies which dataset split the sub-folder belongs to (train, eval or test).
sentences	List containing a set of dictionaries for each sentence. Each dictionary comprises sentence-specific transcription and metadata.
sentence_language_code	Defines the language of the sentence. Possible values are nb-NO (Bokmål), nn-NO (Nynorsk), and en-US (English).
sentence_text	Non-normalised text transcription of the sentence.
audio_file	Name of the audio file in the accompanying audio folder belonging to the sentence.
normsentence_text	Normalised text transcription of the sentence.
transsentence_text	Normalised machine translation of the transcribed sentence. If the manual transcription is in Bokmål, the machine-translated sentence is in Nynorsk. If the manual transcription is in Nynorsk, the machine-translated sentence is in Bokmål.

Table 4.5: Overview of the features from the *NPSC_speaker_data.json* file that are relevant to this thesis.

Feature	Description
speaker_id	ID of the speaker.
date_of_birth	Date of birth of the speaker.
electoral_district	Electoral district the speaker is assigned to. If unknown, the value is null.
gender	Gender of the speaker. Male or Female.
dialect	Dialect region of the speaker.

4.3 Pre-Processing

Before fine-tuning Whisper, the data from the NPSC dataset was pre-processed in several ways. This section provides an overview of the different pre-processing steps that were performed before the model was trained.

4.3.1 Dataset Restructuring

The NPSC dataset is split and compressed into five *zip*-files, each at a size of $18 - 22$ GB. Each zip file contains a certain number of folders with each folder comprising the audio and annotation data for a single plenary meeting day at the parliament as elaborated in Section 4.2.5. The original structure of the dataset was not ideal to work with, as the audio and annotation files were scattered across a set of folders. What is more, each folder also contained several files that were not needed for training and consumed a lot of disk space, such as the full-length recording of the speech.

To facilitate the training procedure and reduce overall disk space consumption, the dataset was restructured. All files except for the sentence_data and sentence-specific audio files within the audio folder were deleted, as they were not required for training. Three empty folders were created, one for each split of the dataset. The sentence-specific audio files of each *yyyymmdd* sub-folder were moved to one of the folders based on the *data_split* feature in the corresponding folder-specific annotation file. Since the defined split is valid for the entire *yyyymmdd* folder, all sentence-specific files within the accompanying audio folder were moved to the respective split folder. Lastly, a sentence_data file was created for each split folder by combining the folder-specific annotation files of the split.

The split-specific folders were then compressed into a *tar.gz* file to save additional storage space. Restructuring and further compressing the dataset resulted in an overall size of 47.3 GB compared to 103.41 GB of the original dataset.

4.3.2 Deletion of Special Tokens, Characters and English Transcriptions

As described in Sections 4.2, a set of special tokens were used in some of the sentence transcriptions. The tokens as well as the Kleene stars were removed from each sentence, if present. Aside from that, the dataset also contains a few audio files in English with corresponding English transcriptions. These were excluded from the dataset during training and evaluation.

4.3.3 Audio Down-Sampling

The audio files of the NPSC dataset were sampled at 48 kHz. Still, Whisper requires the audio to be sampled at 16 kHz. Thus, all audio files were down-sampled to 16 kHz before training to be compatible with the model.

4.3.4 Additive Real World Noise

To test the ability of Whisper to recognise speech in a noisy environment, real-world background noise was added to the NPSC audio files. The audio file used for adding

noise is 8 minutes and 5 seconds long, sampled at 48.000 Hz, and was recorded in a canteen during the lunch break at the Norwegian University of Science and Technology. The audio was downsampled to 16.000 Hz to match the sampling frequency used by Whisper. Due to privacy concerns, the recorded data and the modified NPSC audio files are not being published.

The signal-to-noise (SNR) is a metric used to compare the power of a signal to the level of background noise and it is defined as

$$SNR = \frac{P_{\text{signal}}}{P_{\text{noise}}} = 10 \cdot \log_{10} \left(\frac{RMS_{\text{signal}}}{RMS_{\text{noise}}} \right)^2, \tag{4.1}$$

where RMS denotes the root-mean-squared value of the signal [23]. It was used to add the background noise to the audio signal with the desired SNR.

Let X denote the original, unmodified audio signal and Y the signal of the background noise that should be added to X. By rearranging Equation 4.1, the RMS_{noise} value of a signal can be computed given an SNR value, i.e.,

$$RMS_{\text{noise}} = \sqrt{\frac{RMS_{\text{signal}}^2}{10^{\frac{SNR}{10}}}}. \tag{4.2}$$

To achieve a mixed signal with the desired SNR value, Equation 4.2 was first used to compute the $RMS_{\text{noise}}(X)$ value of signal X. This was then used to modify signal Y by multiplying it with the ratio between the $RMS(Y)$ value of signal Y and the $RMS_{\text{noise}}(X)$ value of signal X, that is,

$$\hat{Y} = Y \cdot \frac{RMS(Y)}{RMS_{\text{noise}}(X)}. \tag{4.3}$$

Lastly, $\hat{Y}$ was added to X to obtain a noisy signal $\hat{X}$ with the desired SNR value:

$$\hat{X} = X + \hat{Y}. \tag{4.4}$$

Multiple experiments with different signal-to-noise ratios were conducted. The results of the experiments are presented and discussed in detail in Chapter 5.1.6.

4.4 Evaluation Metrics

To compare the performance of differently trained models, the word error rate (WER) is typically used [1], [2], [4], [19], [31]. It compares the predicted transcription with a reference transcription of length N by computing the total number of modifications required for the prediction and reference transcription to match [1]. The word error rate is defined as

$$WER = \frac{I + D + S}{N} * 100 \tag{4.5}$$

based the sum of modifications to the prediction in terms of insertions (I), deletions (D), and substitutions (S) [1]. Since the prediction can be of arbitrary length, the WER exceed 100 in some cases.

4.5 Model Training

The medium-sized Whisper model was fine-tuned separately on Bokmål and Nynorsk and in the end, two models were created. Both models were trained multiple times on the normalised sentences of the NPSC training dataset using different parameters to find the best-performing model. In order to train the models, the Huggingface library[4] was used as it comes with built-in support for Whipser and provides all the required tools for training and fine-tuning the model.

4.5.1 Feature Extraction

Before training the model, the audio needs to be mapped to the format expected by the model. As discussed in Chapter 3.6, Whisper requires the audio to be mapped to a log-mel spectrogram. What is more, all audio chunks need to be exactly 30 seconds long. To map the audio files to the expected format, the WhisperFeatureExtractor[5] was used. It assures that all audio chunks are cut or padded to 30 seconds and extracts the mel-filter bank features using an STFT (see Chapter 2.1.3).

4.5.2 Tokenizer

Whisper uses an off-the-shelf Transformer [38] architecture, which uses a softmax layer as its output layer. The softmax produces a probability distribution over a set of token ids, which need to be mapped to the corresponding words to obtain the actual transcription. In addition, since the model is trained to predict token ids, target labels need to be mapped to the respective token ids before training. Mapping the targets and predicted labels back and forth is handled by the WhisperTokenizer[6]. The tokenizer is also used during pre-processing for padding the target labels accordingly.

4.5.3 Loss Function

Since the goal of the model is to predict the most likely token belonging to a set of input features from a set of possible tokens, softmax cross-entropy loss is used for training Whisper (see Chapter 2.3.3).

4.5.4 Hyperparameter Tuning

The models were trained several times on the NPSC training split with varying parameters to find the best-performing model. To facilitate training, a Hugging Face Sequence-To-Sequence trainer[7] was used. The training and evaluation batch size was restricted to 8 due to limited hardware resources. The learning rate, weight decay rate, as well as the number of warm-up, training, and gradient accumulation steps, were varied several times to find the best model. The results are discussed in further detail in Chapter 5.3. The remaining training parameters were left at their default values.

[4]https://huggingface.co
[5]https://huggingface.co/docs/transformers/main/model_doc/whisper
[6]https://huggingface.co/docs/transformers/main/model_doc/whisper
[7]https://huggingface.co/docs/transformers/main_classes/trainer

4.5.5 Training Hardware

The training was conducted on NTNUs Idun cluster [33] using an NVIDIA A100 GPU with 80 GB of memory, an Intel Xeon E5-2695 v4 CPU, and 20 GB of working memory.

4.6 Summary

In the first section of this chapter, a high-level overview of the approach was given. Whisper is fine-tuned and evaluated due to some major restrictions of the speech recognition system that is currently used in the Furhat robot. Since Whisper is capable of transcribing various languages with a low WER and is more robust against high noise compared to other state-of-the-art ASR approaches, it may be an ideal candidate speech recognition model for the Furhat robot. Thus, it is analysed with respect to different aspects, including speaker performance, transcription of names and abbreviations, and how well the model performs in the presence of noise (see research questions **RQ3**—**RQ5**). What is more, since the medium-sized model scored a relatively high WER on the validation split of the NPSC dataset and because it was only trained on Nynorsk for translation-related tasks, Whisper is fine-tuned to determine if the performance can be improved on both Bokmål and Nynorsk (see research questions **RQ1** and **RQ2**).

In Section 4.2, the Norwegian Parliament Speech Corpus dataset was introduced and described in detail. It is the primary dataset used for fine-tuning and evaluating Whisper and contains more than 140 hours of data obtained from speeches at the Norwegian parliament. What is more, it covers a wide range of dialects and also provides annotations in both Bokmål and Nynorsk, which makes it ideal for the purposes of this thesis.

Section 4.3 covered the data pre-processing steps that were performed in order to prepare the data for model training. The dataset had to be restructured and unwanted files were deleted to decrease the large amount of storage used by the dataset. In the end, the overall size was decreased from 103.41 GB to 47.3 GB. Apart from restructuring, the special tokens and English transcriptions had to be removed and the audio had to be downsampled to $16,000$ Hz as required by Whisper. Finally, the section also covered how real-world noise was added to the audio files to test the noise robustness of the model.

Sections 4.4 and 4.5 introduced the WER metric which is used for comparing the performance of differently trained models and discussed how the medium-sized Whisper model was trained on the NPSC dataset. In the end, two fine-tuned models were created as Whisper was trained separately on Bokmål and Nynorsk.

Chapter 5

Evaluation

The medium-sized Whisper [31] model was fine-tuned using the NPSC [35] dataset in accordance with the methodology described in Chapter 4. In the end, two separate models were developed by fine-tuning the medium-sized Whisper [31] model on Bokmål and Nynorsk. Both models were trained on the training split of the NPSC dataset and were run on the validation split to evaluate model performance during training. In order to find the best-performing model, multiple training sessions with different hyperparameters were conducted. The results are presented and analysed in detail in Section 5.1. This is followed by a comparison with other speech recognition models in Section 5.2. Lastly, an overview of the experimental results is given in Section 5.3.

5.1 Results

The results presented in this section are based on the performance of the fine-tuned models on the test split of the NPSC dataset. Model performance is analysed using the WER metric (see Chapter 4.4). Ground truth sentences containing an *<INAUDBILE>* token were excluded from the evaluation since it is impossible to compare the prediction with the ground truth if parts of the ground truth sentence are missing. The ground truth sentences in the dataset are generally without punctuation and capital letters at the beginning of a sentence, except when the sentence starts with a name or abbreviation. However, the baseline models generally capitalise the first letter of the predictions. In order to allow for a fair comparison of the models, every word in both the ground truth and prediction, including names and abbreviations, is converted to lowercase. This is done to avoid correctly predicted words being classified as incorrect due to capitalisation. The models are analysed with regard to overall performance, sentence length, dialects, transcription of names and abbreviations, as well as various characteristics of the speakers, that is, age and gender.

5.1.1 Overall Performance

Table 5.1 shows the WERs achieved by the fine-tuned Whisper models as well as the WER of the baseline, medium-sized model on the validation and test split of the NPSC dataset. Overall, the performance has been greatly improved by fine-tuning the models on the NPSC dataset, especially for Nynorsk. The fine-tuned Bokmål

Table 5.1: Mean WER scored on the NPSC validation and test set by the baseline model and the fine-tuned models. The fine-tuned models scored a considerably lower WER on the test set than the default Whisper model with a difference Δ in WER of -27.49 and -56.24. What is more, the standard deviation σ of the fine-tuned models is also substantially lower than the baseline models, indicating low variation in the WER across the sentences in the test split.

Language	$\text{WER}_{\text{Baseline}}$	$\text{WER}_{\text{Fine-Tuned}}$	Δ	σ_{Baseline}	$\sigma_{\text{Fine-Tuned}}$
Bokmål	37.55	10.06	-27.49	222.01	16.58
Nynorsk	67.77	11.53	-56.24	292.58	18.09

model achieved an average WER on the test set of 10.06 compared to 37.55 of the baseline model. The mean WER of the Nynorsk model is 11.53 and is slightly higher than the Bokmål model, but it is still a massive improvement compared to the WER of 67.77 scored by the baseline model. The fine-tuned models have also a much lower standard deviation σ across the sentences of the test split than the baseline models, which indicates that the overall variance in WER has been reduced.

While the baseline model in Bokmål correctly predicted many words, it often misspelt words even though they would sound similar, such as *alvoret* and *alvore* or *formannsland* and *formandsland* (see Table 5.2). The fine-tuned model generally improved upon this issue, but it also predicts some words incorrectly that were predicted correctly by the baseline model, such as *altså* instead of *ansvar* in the fourth example in Table 5.2. When running the baseline model on Nynorsk, the predicted words are mostly not in Nynorsk but in Bokmål instead and in some cases even in Swedish (see Table 5.3). For instance, words such as *eg* and *dei*, were transcribed to *jeg* and *de*. The fine-tuned model works much better on Nynorsk and correctly transcribes most words to Nynorsk. However, in some cases, the gender of the word is predicted incorrectly as shown in the first example of Table 5.3. Instead of *den beste skulen* the model predicted *det beste skulen*, even though all training sentences containing the word *skulen* use the correct gender.

It is important to note, however, that the baseline Whisper models were originally not trained on Nynorsk for multilingual speech recognition. As noted in Chapter 3.6, Whisper is a multi-task model and can either transcribe a spoken language to its respective written form or it can translate the language to English. Whisper was only trained on Nynorsk for translation purposes and not for multilingual speech recognition, which might explain the substantially higher WERs and variations in transcription performance [31]. Since Whisper was trained on 266 hours of data in Norwegian Bokmål for multilingual speech recognition, its performance on Bokmål is noticeably better [31].

5.1.2 Dialect Performance

One of the major challenges of the speech recognition system used in Furhat is to correctly transcribe Norwegian dialects, which can vary substantially from region to region. Norwegian speakers that have tested the system often had to switch to a neutral dialect instead of using their native dialect due to the system not being able to recognise what was being said. Ideally, speakers should be able to use their native dialect to communicate with the system to make the interaction as natural as

Table 5.2: Example predictions in Bokmål by the baseline and fine-tuned models. In comparison to running the model in Nynorsk, the baseline model works pretty well on Bokmål. The model often transcribes words incorrectly although the pronunciation would be similar, e.g., *da for* instead of *derfor* or *alvore* instead of *alvoret*. Fine-tuning the model resolved most of the issues. However, some words that were correctly classified by the baseline model are no longer correct when using the fine-tuned model, such as *altså* instead of *ansvar* in the fourth example.

Ground Truth	Baseline model	Fine-tuned
derfor trenger vi verdens beste skole	da fortrenger vi verdens beste skole	derfor trenger vi verdens beste skole
de er formannsland neste år	de er formandsland neste år	de er formålslått neste år
spørsmålet er hvordan kan vi få dette til	spørsmålet er hvordan vi kan få dette til	spørsmålet er hvordan kan vi få dette til
president regjeringen har tatt ansvar for det	resultatet har regjeringen tatt ansvar for det	president regjeringen har tatt altså for det
da undertegnede	da unutegner jeg	da undertegnede
vi tar det alvoret på arbeid på alvor	de tar det alvore på arbeid på alvore	vi tar det alvoret på arbeid på alvor
jeg mener at minimumsbemanningen er fornuftig å å gjennomføre	jeg mener at minimumsbevandlingen er fornuftig å gjennomføre	jeg mener at minimumsbemanningen er fornuftig å å gjennomføre

Table 5.3: Example predictions in Nynorsk by the baseline and fine-tuned models. The baseline model predictions are often in Bokmål and in some cases even in Swedish (fifth and last example). Words such as *eg*, *dei*, and *korleis* are often transcribed to the Bokmål equivalent, i.e., *jeg*, *de* and *hvordan*. The fine-tuned model, on the other hand, transcribes many words correctly to Nynorsk, although it uses the wrong gender in some cases as shown in the first example. The correct transcription would be *den beste skulen* instead of *det beste skulen*.

Ground Truth	Baseline model	Fine-tuned
derfor treng vi den beste skulen i verda	derfor trenger vi verdens beste skole	derfor treng vi det beste skule i verda
dei er formannsland neste år	de er formandsland nestår	dei er formålstenleg neste år
spørsmålet er korleis kan vi få dette til	hvordan kan vi få dette til	spørsmålet er korleis kan vi få dette til
president regjeringa har teke ansvar for det	president regjeringen har trådt anså for det	president regjeringa har teke ansvar for det
då underteikna	det är en uttäckning	då underteikna
vi tek det alvoret på arbeid på alvor	de tar det allvore på arbej på allvår	vi tek det alvoret på arbeid på alvor
eg meiner at minimumsbemanninga er fornuftig å å gjennomføre	jeg mener at minimumsbevandlingen er fornuftig att genomföra	eg meiner at minimumsbemanninga er fornuftig å å gjennomføre

possible. Hence, the purpose of this section is to investigate how well the fine-tuned Whisper models perform across the different dialects.

To analyse the dialect performance of the models, the predictions were grouped by the respective dialect of the speaker and the mean WER was computed. The WER of the baseline Whisper models based on the dialect categories used in the NPSC dataset (see Chapter 4) are plotted in Figure 5.1(a) for Bokmål and Figure 5.1(b) for Nynorsk. Figure 5.2 shows the WERs of the fine-tuned models.

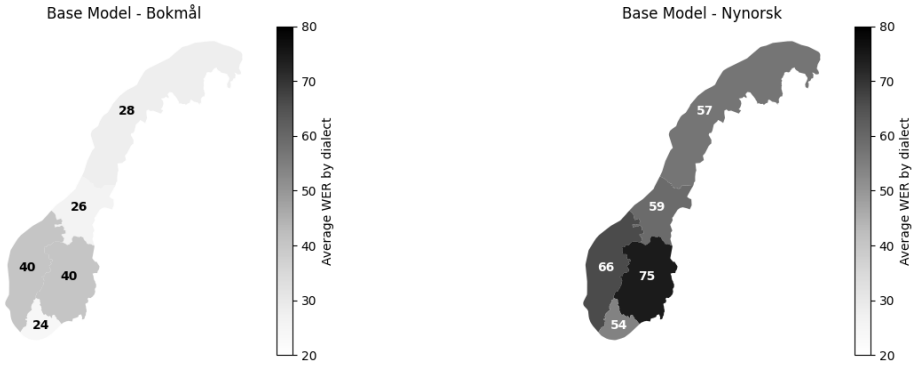
Compared to the baseline model, the fine-tuned models achieved a considerably lower and more stable WER across all dialects. The WER of the Bokmål model is relatively low for most dialects, ranging from 7 to 10. However, the model struggled more with the Western dialect, which had an error rate of 13, even though the Western dialect constitutes 26% of the training data as shown by Figure 5.3. This also holds true for the Nynorsk model, which also achieved slightly higher error rates on the other dialects ranging from 9 to 12. Interestingly, both models scored best on the Trønderlag dialect although the model was trained on relatively few speeches with speakers from that region compared to the other dialects. This might be due to the training data used for the baseline model, as the dialect-based WER for the Trøndelag dialect of the baseline models were rather low in comparison with the remaining dialects.

To get a better understanding of the dialect performance, the fine-tuned models have been studied with regard to the 19 official electoral districts in Norway. An overview of both the electoral districts as well as the WERs is given in Figure 5.4 and 5.5 for the fine-tuned Bokmål and Nynorsk model respectively. In addition, Figure 5.6 shows the distribution of electoral districts in the training dataset. The districts of Sogn and Fjordane and Hordaland were particularly challenging for the Bokmål model, which obtained a WER of 17 and 14 accordingly. However, while 11% of the speeches in the training dataset are from speakers from the region of Hordaland, the district of Sogn og Fjordane only constitutes approximately 2% of the data. Thus, more training data might have resulted in a lower WER for that specific region. The WER for the region of Oppland was at 13 also rather high compared to the remaining districts that are considered to be part of the Eastern dialect despite a high training data percentage of 7%. In addition to Sogn and Fjordana, Hordaland and Oppland, the Nynorsk model also had difficulties transcribing speakers from Finnmark and Østfold, with the latter being quite similar to the Oslo dialect. However, the error rate of 16 is considerably higher than the error rate of the district of Oslo, which is 11. Interestingly, both models achieved the lowest score in the district of Hedmark with a WER of just 4 and 6, meaning the transcriptions of speakers from that area were for the most part correct.

5.1.3 Sentences with Names and Abbreviations

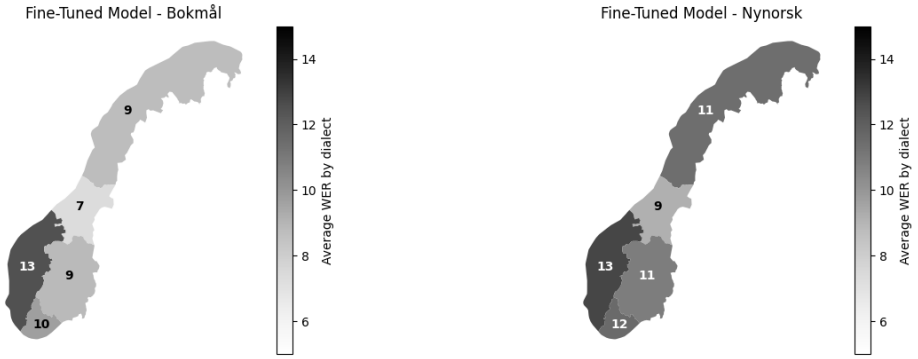
Names and abbreviations are frequently used in sentences and proved to be a major challenge of the speech recognition system of the Furhat robot used at NorwAI, whether in Norwegian or English. The purpose of this section is to analyse the performance of the baseline and fine-tuned Whisper models in more detail with respect to names and abbreviations.

All words in the ground-truth annotations of the NPSC dataset use small letters even if the word is at the start of a sentence, except for when the word is a name



(a) Norwegian Bokmål (b) Norwegian Nynorsk

Figure 5.1: Performance of the Whisper baseline models with respect to the dialect categories used in the NPSC dataset. The performance varies quite a bit with both models achieving the highest WER on the Western and Eastern dialects. The Southern dialect has the lowest WER.



(a) Norwegian Bokmål (b) Norwegian Nynorsk

Figure 5.2: Performance of the fine-tuned Whisper models based on the NPSC dialect categories. Dialect performance has been considerably improved for both models and the variations in WER are not as high as the baseline models. The Bokmål model performs slightly better than the Nynorsk model. The Western dialect has the highest WER for both models while the lowest WER was achieved on the Trøndelag dialect.

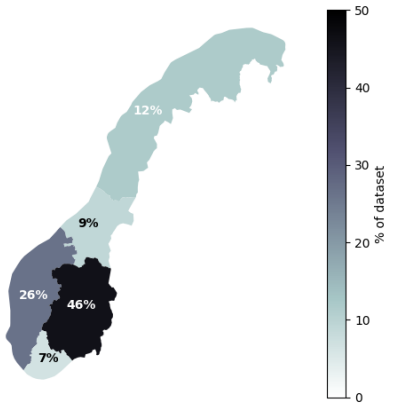


Figure 5.3: Dialect distribution of the training dataset.

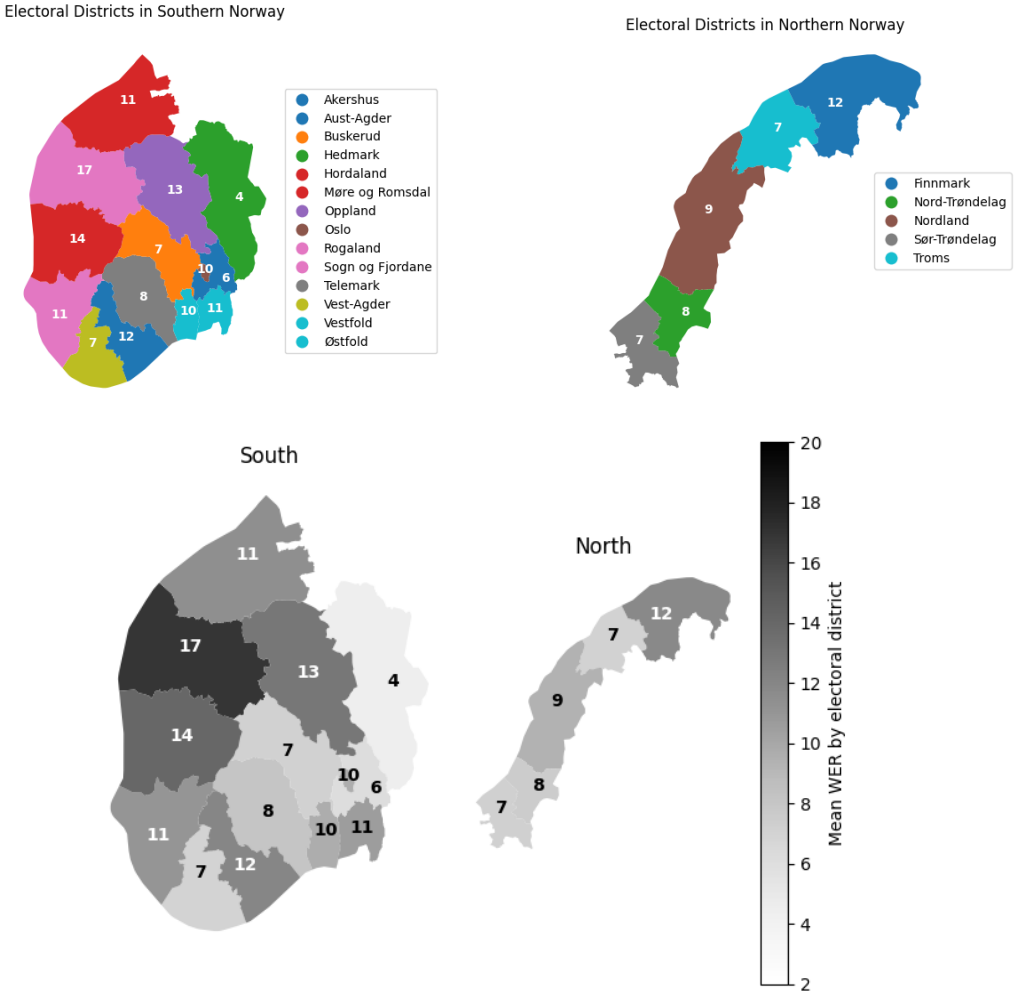


Figure 5.4: Overview of the 19 electoral districts in Norway with the corresponding WERs scored by the fine-tuned Whisper model on Bokmål. 2 speakers, corresponding to 287 sentences or 4.52% of the test split, are not assigned to any district and were excluded from this plot. The model struggled the most with speakers from the districts of Sogn og Fjordane and Hordaland.

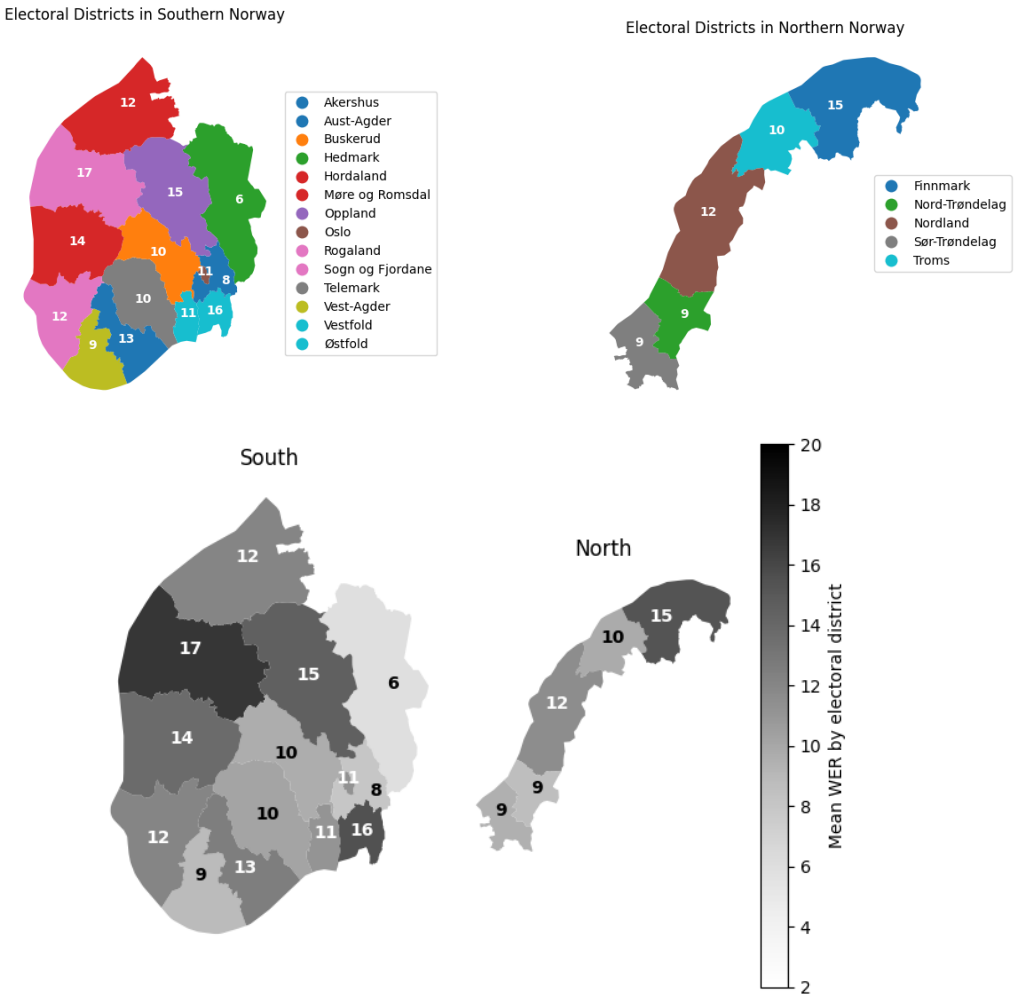


Figure 5.5: Overview of the 19 electoral districts in Norway with the corresponding WERs scored by the fine-tuned Whisper model on Nynorsk. 2 speakers, corresponding to 287 sentences or 4.52% of the test split, are not assigned to any district and were excluded from this plot. While the districts of Sogn and Fjordane and Hordaland were among the most challenging districts as well, the Nynorsk model had also difficulties transcribing speakers from Østfold, Oppland, and Finnmark.

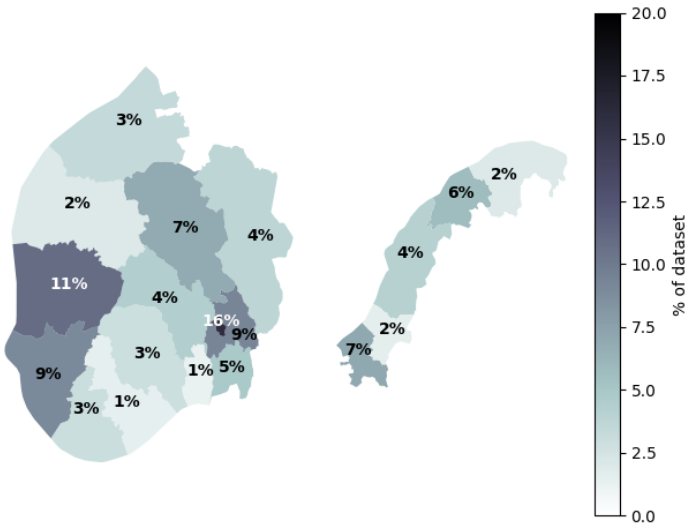


Figure 5.6: Distribution of electoral districts in the training dataset. The colours represent the percentage of speeches given by speakers assigned to the respective electoral district. 8 speakers, corresponding to 3545 sentences or 6.9% of the dataset, are not assigned to any electoral district and were excluded from this plot.

or an abbreviation. Hence, in order to identify sentences that contain names or abbreviations, the sentences were filtered based on whether or not a capital letter was present. If it did, the sentence comprised a name or abbreviation. This was done before the sentences were converted to lowercase for the computation of the WER as mentioned in the introduction of this section. The predictions of both the baseline and fine-tuned models were then grouped by whether or not a name or abbreviation was present and the mean WER was computed. The results are given in Table 5.4.

A clear improvement in terms of WER can be seen when running both the baseline and fine-tuned models on sentences without any names or abbreviations. The difference is particularly strong for the baseline model on Bokmål. The predictions of the baseline models are often completely unrelated to the actual ground truth if a name or abbreviation is present as shown in Table 5.5 and Table 5.6. In contrast, the predictions of the fine-tuned model are in many cases closer to the actual ground truth, even if the predicted word is not entirely correct. The overall error rates were drastically reduced by fine-tuning the models, resulting in 8.84 and 10.35 for Bokmål and Nynorsk for sentences without any names or abbreviations as well as 11.95 and 13.36 for sentences with names or abbreviations. Nevertheless, while the WER gap between sentences with and without names or abbreviations has been reduced quite a bit, it still prevails. Even though fine-tuning does help reduce the error rate, which makes sense since the model has an opportunity to learn the names and abbreviations introduced by the dataset, the performance might deteriorate once the model is exposed to unseen data.

5.1.4 Performance by Sentence Length

Conversations with the Furhat robot used at NorwAI tend to vary in length. While the maximum query length supported by the robot is currently limited, the speech recognition model should be capable of handling sentences of varying lengths without

Table 5.4: WER of the baseline and fine-tuned Whisper models based on sentences with and without any names or abbreviations. Both the baseline and fine-tuned models achieve a notably lower WER on sentences without any names or abbreviations. The difference is particularly strong for the baseline model on Bokmål.

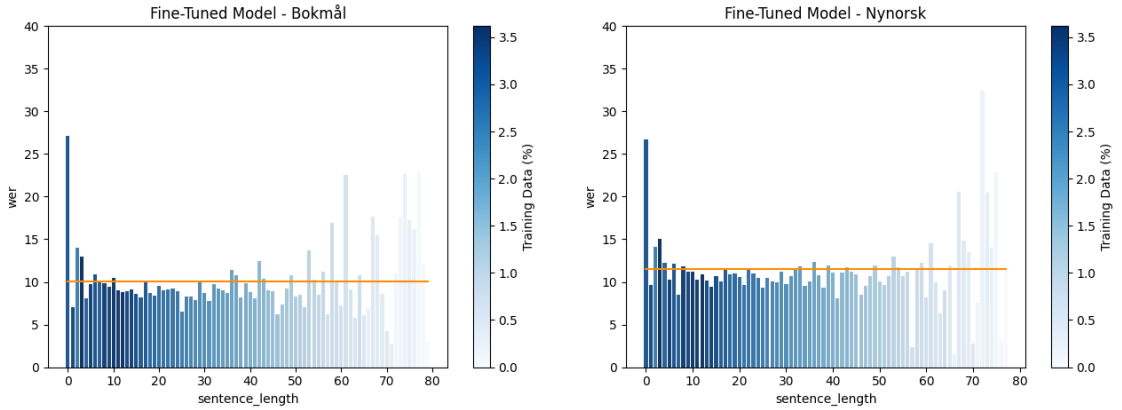
Names	Without names/abbreviations	With names/abbreviations
Whisper-Medium (Bokmål)	25.48	56.20
Whisper-Medium (Nynorsk)	65.19	72.31
Fine-tuned model (Bokmål)	8.84	11.95
Fine-tuned model (Nynorsk)	10.35	13.36

Table 5.5: Examples of incorrect predictions of sentences containing names or abbreviations in Bokmål. The predictions of the fine-tuned model are often closer to the ground truth than the baseline model.

Ground Truth	Baseline model	Fine-tuned
ja fremskrittspart nei president	jeg har frimskitt for meg of jeg har visst better	ja president
skei	skjønner	skeie
steffensen	det var sånn	fepersen
hoksrud	ok sælut	moxnes
stensland	det er sant	steens land

Table 5.6: Examples of incorrect predictions of sentences containing names or abbreviations in Nynorsk. The baseline model switches to English in some cases, which it does not do when running it on Bokmål.

Ground Truth	Baseline model	Fine-tuned
statsminister solberg	just a little bit more back	statsråd soldberg
steffensen	that was it	elvestuen
hoksrud	boxing is over	vågslid
gjelsvik	yeah see	gjertsen
neste talar tellef inge mørland	næste tal er telefingemøller	neste talar telef inge mørland



(a) Norwegian Bokmål

(b) Norwegian Nynorsk

Figure 5.7: Performance of the fine-tuned Bokmål model (a) and Nynorsk model (b) with respect to the length of the sentences. The hue of the bars denotes the percentage of sentences with the respective number of words in the training dataset. The orange line represents the mean WER achieved by the model. WER is relatively stable across most sentence lengths for both models. The frequency of spikes with greater than average WER tends to increase for very long sentences, which is presumably due to comparatively low training data. Single words sentences are problematic for both models despite sufficient training data due to the more frequent occurrence of names or abbreviations.

affecting its performance. This section analyses the WER of the fine-tuned Whisper models with respect to different sentence lengths. The results are given in Figure 5.7(a) for Bokmål and Figure 5.7(b) for Nynorsk. The hue of the bars indicates the total percentage of training data with the corresponding sentence length.

All in all, the WER of both models is below the mean WER for most sentence lengths with only slight variations. The occurrence of spikes with a noticeably higher WER tends to increase for very long sentences exceeding 60 words. However, it is important to note that the percentage of training data decreases with increasing sentence length. As a consequence, it cannot be concluded that the model performs worse on long sentences as the WER might decrease and stabilise with more training data.

Sentences containing only a single word cause the WER to increase substantially to 27.11 for the fine-tuned Bokmål and 26.76 for the Nynorsk model despite sufficient training data. A closer analysis revealed that this stark increase was caused by the wrong classification of names. The test dataset contains a total of 284 sentences with a single word, out of which 196 are names or abbreviations. The mean for a sentence with a single word containing a name or abbreviation is 35.71 compared to a WER of 7.96 for a single-word sentence without a name. This also applies to the Nynorsk model with a corresponding WER of 36.22 with names and 5.68 without.

5.1.5 Performance by Speaker

Since the Furhat robot is supposed to be able to interact with different people, another important feature of the speech recognition model used in the system is the ability

to handle different speakers, regardless of gender, age, and dialect. This section analyses how well Whisper handles the different speakers of the NPSC dataset.

Table 5.7 contains the mean WER of the fine-tuned models based on the gender of the speaker. Although both models achieve a slightly higher WER for male speakers, the differences are with 1.47 for Bokmål and 1.20 relatively low. The cause of the gap may be the uneven distribution of male and female speakers in the training data where the 62.27% are male and only 37.73% female. Still, since the NPSC dataset is relatively small compared to the amount of data Whisper was trained on, more data will presumably result in an even smaller gap.

Figure 5.8 plot the WER against the individual speakers and their respective dialects. Only speakers with a WER that exceeded the mean WER of the model are included. In total, 39 speakers in the case of Bokmål and 37 speakers in the case of Nynorsk scored a WER that exceeded the average WER of the corresponding models. As the NPSC dataset comprises a total of 253 different speakers, the models performed worse on 15.41% and 14.63% of the speakers. As shown by Figure 5.8, the majority of the speakers that the Bokmål model struggled with had a Western dialect, which is also the dialect with the highest WER as discussed in Section 5.1.2. The speakers with the speaker id 32 and 40 stand out the most as the WER was relatively high even though the speakers accounted for more than 2% of the training data. Speaker 32 also accounts for 3.95% of the data in the test split while the majority of the speakers have a test split percentage of less than 2% as shown in Figure 5.11. As a result, speaker 32 affected the overall WER of the models slightly more than other speakers. A closer analysis reveals that the speakers 32 and 40 are assigned to the electoral districts of Oppland and Rogaland as shown by Figure 5.9, both of which have a higher than average WER despite being relatively well-represented in the training data (see Figure 5.6). The same observations hold true for the Nynorsk model, although speakers with a WER exceeding 20 occur more frequently. However, as mentioned in previous sections, this is probably due to the fact that Whisper was not trained on Nynorsk for multi-language speech recognition [31].

Lastly, model performance was also analysed with regard to the age of the speakers. Figure 5.10(a) plots the WER based on the age of the speakers. The hue of the bars indicates the percentage of speeches with speakers at the given age in the training split. Figure 5.10(b) shows the WER based on the binned age of the speaker using a bin size of 5. Figure 5.10(c) and (d) contain the same plots for the Nynorsk model. Although it appears that the performance of both models appears to deteriorate slightly with an increase in age, no clear pattern can be observed. While spikes in WER seem to occur more frequently if the speaker is older than 50, especially for the Nynorsk model, this does not apply to all ages above 50 despite a large amount of training data. For example, speakers at the age of 62 have a mean WER of 9.97 while speakers at the age of 55 have a much higher WER at 13.22. Nonetheless, both are well represented in the training data with 6.05% and 4.99%. Although the binned plot might also give the impression of deteriorating performance with age, bins $(69, 73]$ and $(73, 79]$ have a lower than average WER despite a relatively small amount of training data. Thus, it cannot be concluded that the model performs worse in older age groups. While the WER varies across different ages, the differences are most likely due to the dialect of the speaker and not because of the age of the speaker.

Table 5.7: Mean WER based on the gender of the speakers. Overall, the gap in terms of WER between male and female speakers is relatively low. The slight difference in WER may be a result of the gender distribution in the training data where 62.27% of the speakers are male and only 37.73% are female.

Model	WER - male	WER - female
Fine-tuned model (Bokmål)	10.65	9.18
Fine-tuned model (Nynorsk)	12.01	10.81

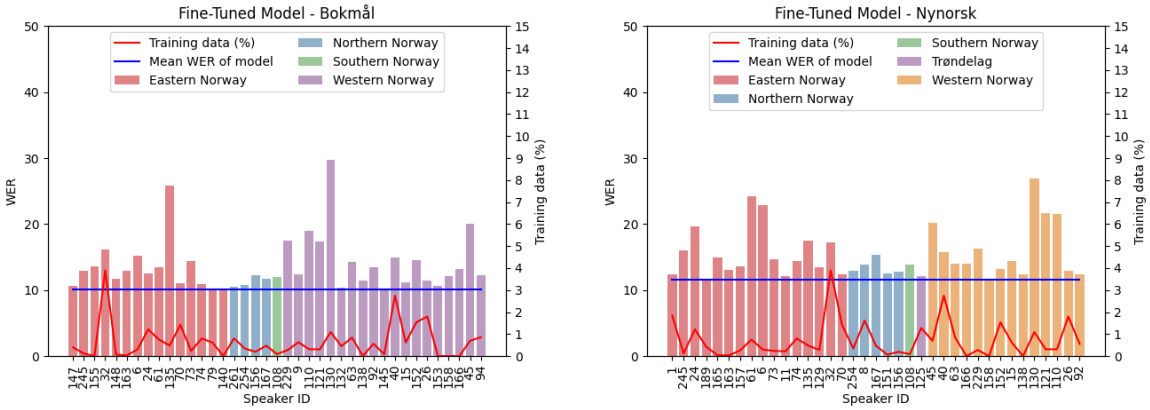


Figure 5.8: WER of the fine-tuned Bokmål and Nynorsk model with regard to the dialect of the speaker. Only speakers for which the WER was higher than the overall mean WER were included. The red line shows the amount of training data in % with the respective speaker id. All in all, 39 (Bokmål) and 37 (Nynorsk) speakers had a WER that exceeded the mean WER of the respective models. Speakers 32 and 40 achieved a greater than average WER even though each account for more than 2% of the training data.

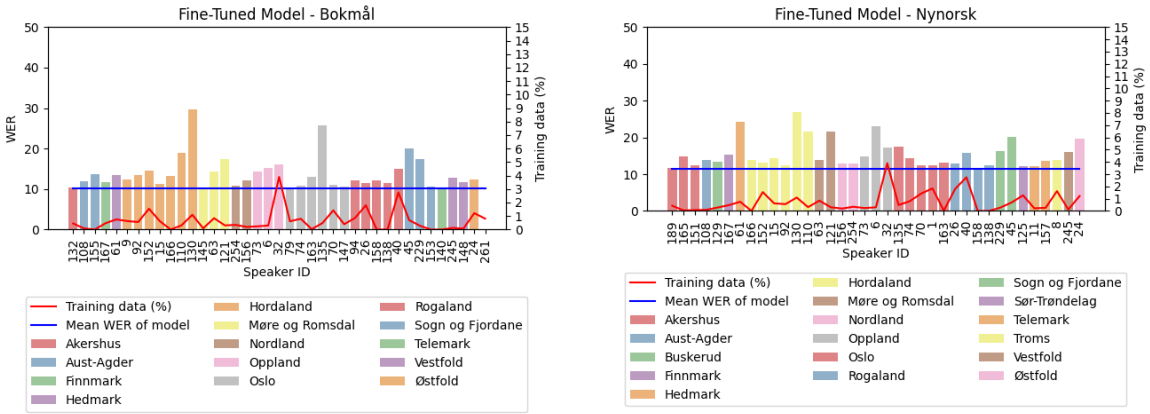
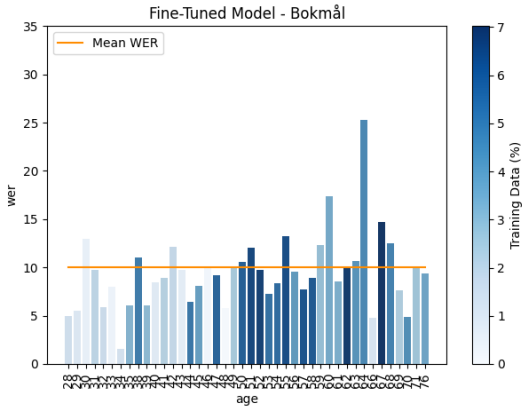
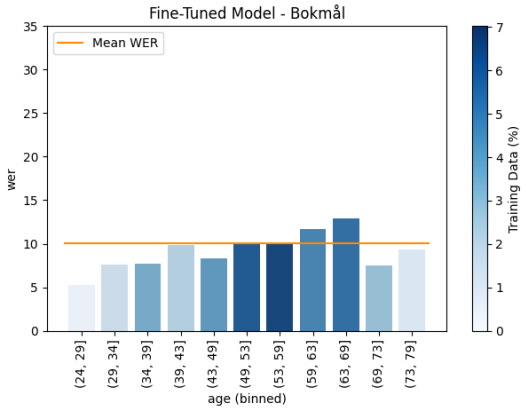


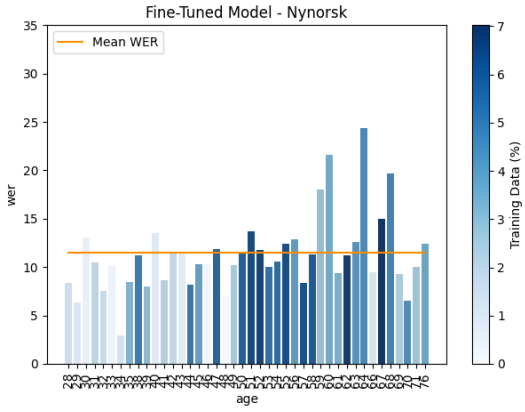
Figure 5.9: WER of the fine-tuned Bokmål and Nynorsk model with regard to the electoral district of the speaker. Only speakers for which the WER was higher than the overall mean WER were included. The red line shows the amount of training data in % with the respective speaker id.



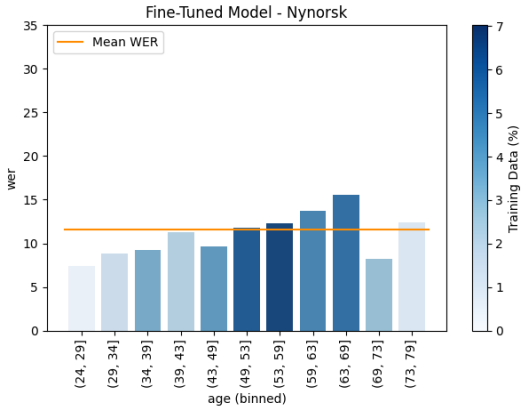
(a) WER by speaker age - Bokmål



(b) WER by binned speaker age - Bokmål



(c) WER by speaker age - Nynorsk



(d) WER by binned speaker age - Nynorsk

Figure 5.10: WER of the fine-tuned Bokmål model with respect to the age of the speaker (a). Figure (b) plots the WER based on the binned age of the speakers using a bin size of 5. The hue indicates the percentage of speakers with the respective age or age bin in the training dataset.

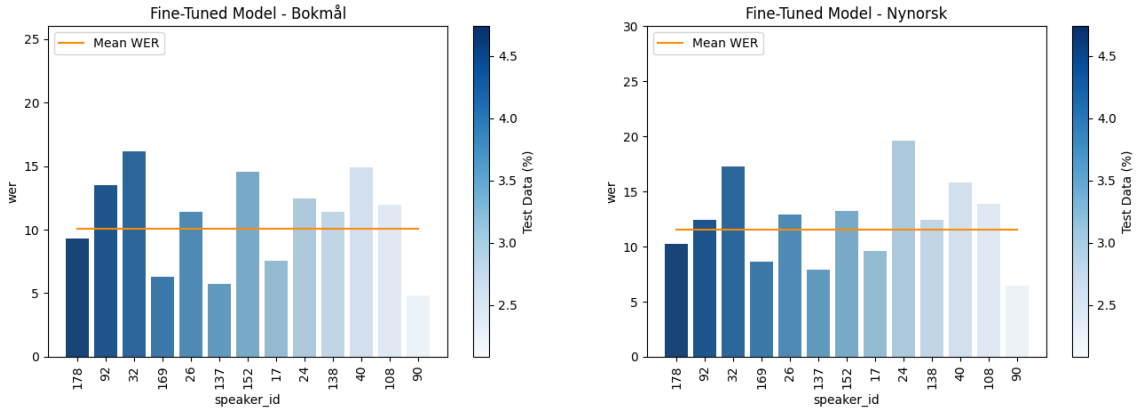


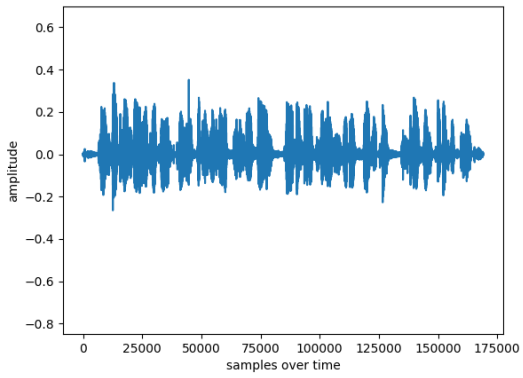
Figure 5.11: WER of the fine-tuned model in Bokmål and Nynorsk based on the speaker id. The hue denotes the percentage of the test data split the speaker accounts for. Only speakers with a test split percentage of more than 2% are included. The majority of the speakers account for less than 2% of the test data. However, a few speakers account for more than 2% of the test data while having a noticeably greater WER at the same time, such as speakers 32 and 24, and thereby affected the overall WER slightly more than other speakers.

5.1.6 Performance in the Presence of Real-World Noise

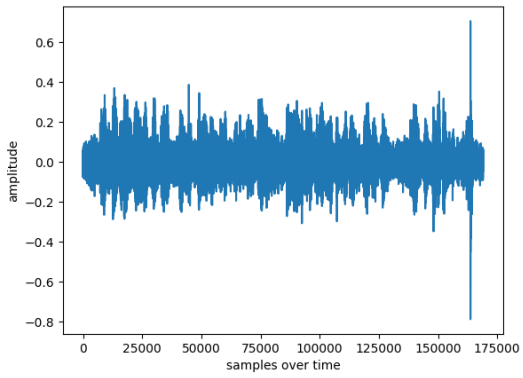
One of the main issues with the current speech recognition system used in the Furhat robot is the poor performance in the presence of noise, such as people chatting in the background. Noise can lead to significant distortions of the signal, making it more difficult to read and interpret. An example of how real-world noise affects the Mel spectrogram of an audio signal is given in Figure 5.12. The presence of background noise often results in queries that diverge greatly from what the person interacting with the robot has said, leading to answers that are not expected by the user. The problem even persists if the user speaks louder and directly into the microphone, indicating that the system only works with a very high signal-to-noise ratio. The ability to withstand some degree of background noise is paramount as the robot is often presented at conferences and keynotes. Hence, the purpose of this section is to analyse the susceptibility of the fine-tuned Whisper models to real-world noise.

The data of the test split was augmented with real-world noise as described in Chapter 4.3.4 and both models were run on the augmented data. Each model was tested using 6 different signal-to-noise ratios, that is, -10, -5, 0, 5, 10, and 20 dB. The results are presented in Table 5.8.

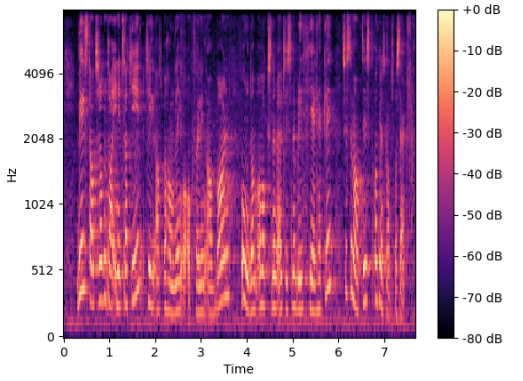
The WERs obtained using different SNRs show that Whisper is robust to low-level noise with an SNR of 20 dB or greater. The WER only increases marginally from 10.06 to 10.58 on Bokmål and from 11.53 to 12.08 on Nynorsk. Exposing the model to high levels of noise at 10 dB results in a slightly stronger increase to 13.59 and 16.32, while any SNR beyond that causes the WER to increase rapidly. These observations match well with the findings by Radford, Kim, Xu, *et al.* [31], which are shown in Figure 3.4. The WER of the fine-tuned Whisper models is stable at low levels of noise and only increases slightly when exposed to high levels of noise up to 10 dB. However, while the SNR should in an ideal case stay above 10 dB to guarantee a sufficiently correct transcription, the speech recognition model can potentially still



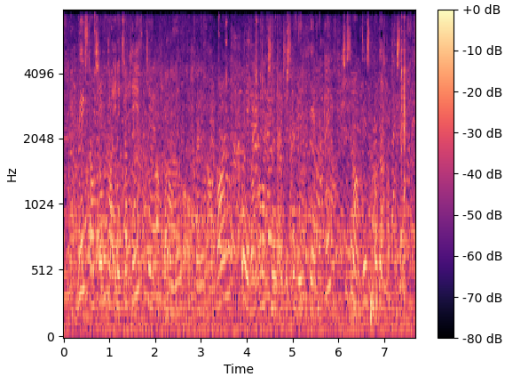
(a) Digitised signal without noise



(b) Digitised signal with noise



(c) Mel spectrogram without noise



(d) Mel spectrogram with noise

Figure 5.12: Plot (a) and (c) show the digitised signal and Mel spectrogram of a speech given at the Norwegian parliament without noise. Plot (b) and (d) are based on the same signal combined with additive real-world noise using an SNR of 0 dB. Adding real-world makes it harder to read and interpret the signal. For instance, pauses made by the speaker are no longer as easily identifiable as in the plots of the signal without noise.

be used in areas with a lower SNR if a high-quality microphone with the ability to cancel some degree of background noise is used.

5.2 Comparison with Other Models

This section compares the results of the fine-tuned Whisper model presented in the previous section with other models run on the same data. To allow for a fair comparison, punctuation is removed and all words in the ground truth sentences and predictions are converted to lowercase prior to computing the WER. The models Whisper is compared with were not modified in any way.

Table 5.8: WER based on various signal-to-noise ratios using the fine-tuned Whisper models for Bokmål and Nynorsk. Whisper shows some degree of noise robustness, even though the WER increased slightly even for SNRs of 10 and 20 dB. The WER starts to deteriorate substantially at an SNR of 5 dB.

SNR (dB)	WER (Bokmål)	WER (Nynorsk)
−10	44.80	48.96
−5	40.45	46.08
0	37.64	43.37
5	20.99	24.26
10	13.59	16.32
20	10.58	12.08

5.2.1 Whisper Small

Inference time and model size are important factors that need to be considered when choosing a speech recognition model for the Furhat robot. Compared to the medium-sized model, the small Whisper model requires less GPU memory and has on average a lower inference time on both CPU and CPU as shown in Figure 5.13. This section compares the fine-tuned medium-sized model with a small Whisper model that has been fine-tuned by the national library on multiple datasets[1]. However, the model is still being trained and is not finished yet. The results presented in this section are based on the latest model published on January 8th 2023. The model only supports Bokmål and an equivalent Nynorsk model has not been released yet.

The fine-tuned, small-sized Whisper model achieved an overall WER of 37.36 on the test split of the NPSC dataset, which is remarkably higher than the WER of 10.06 obtained by the fine-tuned, medium-sized model. The overall performance of the small model on different dialects was quite similar to that of the baseline medium-sized model as shown in Figure 5.14(a), with the highest error rates achieved on the Eastern and Western dialects and the lowest on the Southern dialect. Interestingly, the model struggled particularly with speakers from the district of Østfold, which has a mean WER of 103 (see Figure 5.14(b)). The poor performance on the Østfold district is primarily caused by speaker 24, for whom the model has a mean WER of 163.51. What is more, although the district of Hordaland with a mean WER of 44 proves to be difficult for the small model as well, Sogn and Fjordane have a WER of 34, which is lower than the mean WER of the model and similar to the performance on other districts.

Names and abbreviations were generally an issue for the small model as well. Sentences without names or abbreviations have a mean WER of 34.58. The WER increases to 41.65 if the sentence contains a name or abbreviation. Still, the gap between the two WERs is considerably smaller than with the base model, indicating that fine-tuning generally improves model performance on names and abbreviations. The performance with regard to age and gender is quite similar to the medium-sized model, that is, the small model does not perform considerably worse on certain age groups or genders, although the gap in terms of WER between female and male speakers was with 2.43 slightly higher than the fine-tuned medium-sized model.

In terms of sentence length, the small model performs similarly to the fine-tuned,

[1]https://huggingface.co/NbAiLab/whisper-small-nob

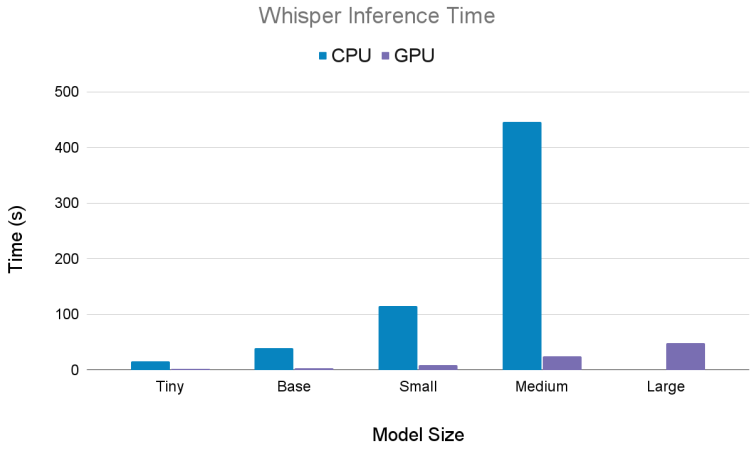


Figure 5.13: Inference time of Whisper using different model sizes. (Source:[2])

medium-sized model (see Figure 5.15). Spikes in WER that are exceeding the mean WER tend to increase with longer sentences. While single-word sentences result in a larger WER much like the medium-sized model, the small model also struggles with two-word sentences that are causing an average WER of 103.35. However, while the high WER of single-word sentences is induced by names or abbreviations, which causes the WER to increase from 46.59 without names/abbreviations to 131.63 with, the opposite is true for two-word sentences. The WER for two-word sentences without names/abbreviations is 174.79 and 38.35 with.

5.2.2 Wav2Vec 2.0

This section compares the fine-tuned, medium-sized Whisper model to Wav2Vec 2.0 (see Chapter 3.5) using the Bokmål and Nynorsk model with 300 million parameters trained by the National Library[3,4] using the NPSC dataset. Since the Wav2Vec 2.0 models were trained on the same dataset as the fine-tuned models discussed in this thesis, the performance is remarkably better than the small Whisper model discussed in the previous section.

On Bokmål, the Wav2Vec 2.0 model achieved a WER of 12.37, which is slightly higher than the error rate of the fine-tuned, medium-sized Whisper model. The Nynorsk model, on the other hand, has a much higher WER of 27.70. However, the WERs reported by the National Library on Huggingface are much lower. Based on the NPSC test split, the Wav2Vec 2.0 Bokmål model achieved an error rate of 7.03 and the Nynorsk model a WER of 12.22. It is not clear why the difference in WERs occurs despite following the evaluation guide that was referred to in the Huggingface repository of the corresponding models [5]. One reason might be that the data pre-processing and post-processing steps used in this thesis diverge from the pipeline used by the National Library, which might have affected the WERs negatively. For instance, it was observed that the Wav2Vec 2.0 model predictions are

[2]https://www.assemblyai.com/blog/how-to-run-openais-whisper-speech-recognition-model/
[3]https://huggingface.co/NbAiLab/nb-wav2vec2-300m-bokmaal
[4]https://huggingface.co/NbAiLab/nb-wav2vec2-300m-nynorsk
[5]https://github.com/huggingface/transformers/tree/main/examples/research_projects/robust-speech-event

(a) WER by dialect

(b) WER by sentence length

Figure 5.14: WER of the small Whisper model fine-tuned by the national library on the NPSC dialect categories (a) and the electoral districts of the respective speakers (b). While the dialect performance is quite similar to the baseline Whipser model, the small model struggles particularly with the district of Østfold, which has a mean WER of 104.

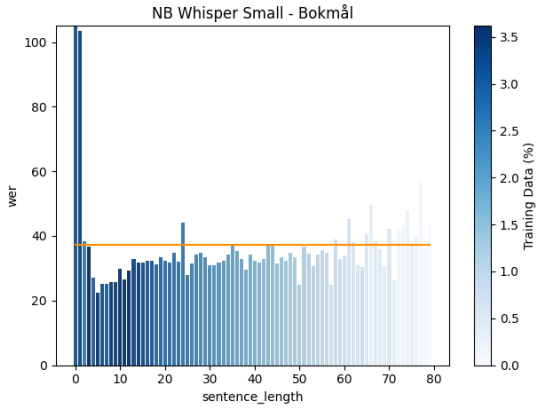


Figure 5.15: WER achieved by the small Whisper model fine-tuned by the national library based on the sentence length. Overall, the performance is quite similar to the medium-sized model. However, two-word sentences also prove to be an issue for the small model, resulting in a WER exceeding 100.

Table 5.9: Wav2Vec 2.0 predicted sentences using the non-normalised form, resulting in numbers and dates being expressed using letters instead of numbers. The Whisper models that were fine-tuned as part of this thesis, however, use the normalised form, resulting in a higher WER for the Wav2Vec 2.0 model.

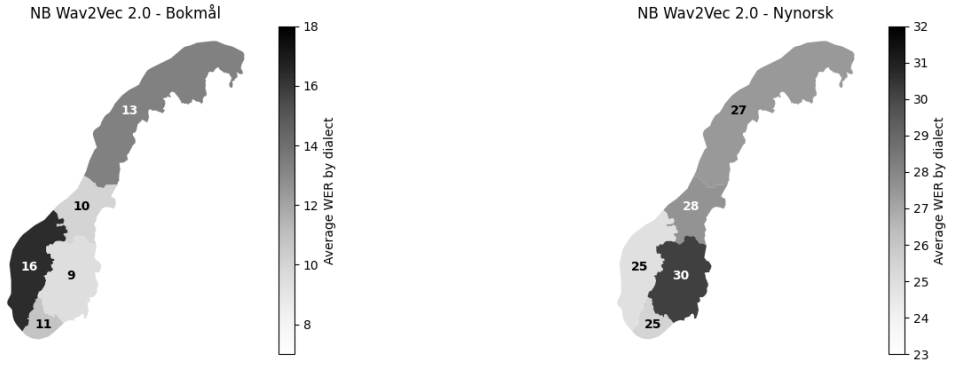
Ground Truth	Prediction
det var 19.4. i år	det var det nittende april i år
dette tilsvarer 300000 plasser	dette tilsvarer tre hundre tusen plasser

Table 5.10: Example predictions by the Wav2Vec 2.0 model in Bokmål. The model often predicts the special tokens mentioned in Chapter 4.2.3, such as *<ee>* and *<qq>*, which were removed from the ground truth prior to fine-tuning the Whisper models.

GT (with tokens)	GT (without tokens)	Prediction
<ee> de er formannsland neste år	de er formannsland neste år	eee de er formannsland neste år
<ee> da undertegnede	da undertegnede	eee da undertegnede
<*ee> og i innstillingen så er det to lyspunkt	og i innstillingen så er det to lyspunkt	eee og i innstillingen så er det to lyspunkt
ja president <qq> utvinning av olje og gass er den største utslippskilden vi har i norge med nesten 15 millioner tonn i utslipp i fjor	ja president utvinning av olje og gass er den største utslippskilden vi har i norge med nesten 15 millioner tonn i utslipp i fjor	ja president eee qqq utvinning av olje og gass er den største utslippskilden vi har i norge med nesten femten millioner tonn i utslipp i fjord

in non-normalised form, i.e., dates and numbers are expressed using letters instead of actual numbers. Still, as mentioned in Chapter 4.5, the models in this thesis are trained and evaluated using the normalised form. Consequently, dates and numbers were categorised as incorrect, leading to an increase in WER (see Table 5.9). Apart from that, the predictions by the Wav2Vec 2.0 model often included sequences of unrelated letters that are not present in the ground truth (see Table 5.10). As mentioned in Chapter 4.2.3, the ground truth transcriptions occasionally contained special tokens which were removed prior to training the Whisper models. It appears that the Wav2Vec 2.0 model attempts to predict the occurrence of these special tokens, which causes the WER to increase.

Contrary to the Whisper models, both Wav2Vec 2.0 models do not appear to have any issues with names or abbreviations appearing in sentences. For the Bokmål model, the WER of sentences without names/abbreviations is 12.43 while sentences with names/abbreviations have a mean WER of 12.28. The Nynorsk model achieved a WER of 27.84 and 27.49 correspondingly. Still, the models were trained on the same NPSC dataset and the error rates might increase when exposing the model to names it has not been trained on.



(a) WER by dialect (Bokmål)

(b) WER by dialect (Nynorsk)

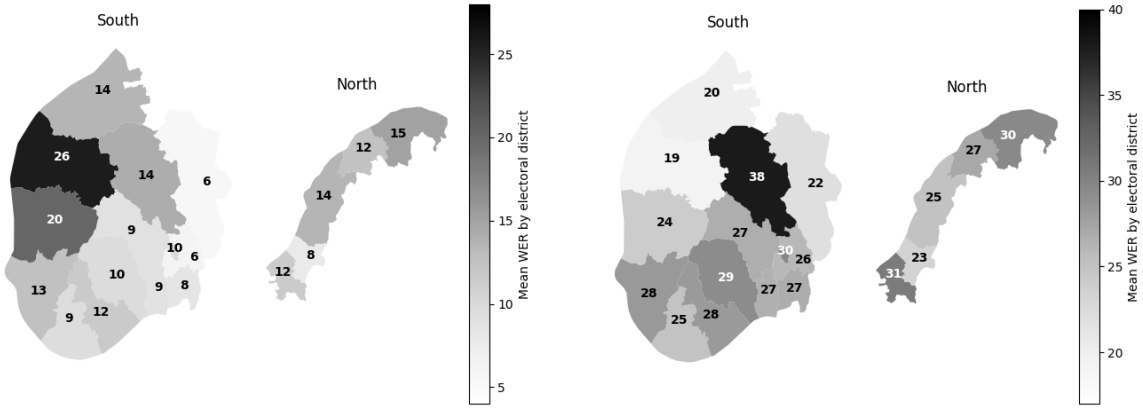
Figure 5.16: WER achieved by the Wav2Vec 2.0 models based on the dialects. The models perform quite differently on the dialects. The Bokmål model performs best on the Eastern and worst on the Western dialect, while the opposite is true for the Nynorsk model.

The dialect performance was different for the Wav2Vec 2.0 models. While the Bokmål model struggled most with the Western dialect, the Nynorsk model performed worst on the Eastern dialect (see Figure 5.16). This suggests that the training data of the models was probably focused on the regions where the respective written forms are mostly used instead of creating models that are generally capable of mapping every dialect to either Nynorsk or Bokmål, regardless of the geographical distribution. This is also reflected in the district-based performance of the models (see Figure 5.17). The Bokmål model performs best on the districts surrounding Oslo and worst on the districts of Sogn and Fjordane and Hordaland. On the other side, the Nynorsk model has its lowest WER for Hordaland and Sogn and Fjordane, while the Eastern and Northern districts generally have a mean WER greater than 25, with the exception of Hedmark and Nord-Trøndelag.

In terms of speaker performance, none of the Wav2Vec 2.0 models performs worse on certain age groups and the difference in WER between female and male speakers is with 1.91 and 2.48 low. Regarding the WER based on the sentence length, the observations that were made for the Whisper models also hold true for the Wav2Vec 2.0 models. Single-word sentences cause the WER to spike and the frequency of WERs exceeding the mean increases with sentence length.

5.3 Experiments

In order to find the best-performing model, several experiments with different hyperparameters were conducted. Overall, the best model was found using a learning rate of $1e-4$, no regularisation, and a gradient accumulation step size of 2. The model was fine-tuned on the NPSC using 8000 steps. This section provides an overview of the experimental results based on the mean WER obtained on the validation split of the NPSC dataset. The WERs presented in this section are slightly higher than the actual WERs as the error rate was computed in a slightly different way during training. While the punctuation was removed, the sentences were not converted



(a) WER by district (Bokmål)

(b) WER by district (Nynorsk)

Figure 5.17: WER of the Wav2Vec 2.0 models on the electoral districts of Norway. In general, the Bokmål model performs best on the electoral districts surrounding Oslo and worst on the districts of Sogn and Fjordane as well as Hordaland. The Nynorsk model, on the other hand, performs best on the latter two and worst on the Eastern and Northern districts, with the exception of Hedmark and Nord-Trøndelag.

Table 5.11: WER obtained on the validation set of the NPSC dataset using different learning rates.

Learning rate	WER (Bokmål)	WER (Nynorsk)
0.001	106.40	122.20
1e−4	10.02	11.34
1e−5	22.38	13.73
1e−6	91.81	75.67

to lowercase, which may cause identical words with different capitalisations to be regarded as incorrect, causing an increase in WER.

5.3.1 Learning Rate

An overview of the WER achieved on the validation set using different learning rates is given in Table 5.11. The best WERs of 10.02 for Bokmål and 11.34 for Nynorsk were obtained using a learning rate of 1e−4. Increasing the learning rate to 0.001 caused the WER to increase noticeably, while a learning rate of 1e−6 proved to be too low, resulting in a WER of 91.81 and 75.67.

5.3.2 Step Size

Apart from the learning rate, the step size of the model was varied as well based on the assumption that a higher number of training steps will result in a lower WER. The initial models were trained using a step size of 8000. Increasing the step size to 24000 did not improve the model in a noticeable way. Hence, further increases in step size were disregarded due to the low improvement in mean WER.

5.3.3 Regularisation

In addition to the learning rate and step size, $L2$ regularisation (see Chapter 2.3.5) was added to the model as the authors of Whisper mentioned that no regularisation was used while training the baseline models. It was assumed that regularisation would increase the generalisation capabilities of the model, improving its overall WER on the validation set. Adding a weight decay of 1e−5 did not result in any improvement of the WER, while a stronger decay rate of 1e−4 caused the WER for both models to increase. Consequently, regularisation was not used for the final models.

5.4 Summary

This chapter presented and analysed the results obtained from running the fine-tuned, medium-sized Whisper models on the NPSC test dataset and how they perform in comparison with other speech recognition models. Section 5.1 provided an overview of the results that were achieved with the fine-tuned models and analysed each model in detail with regard to their performance on dialects, names and abbreviations, varying sentence lengths, and speaker characteristics. In addition, the speech recognition performance was analysed in the presence of real word noise using different signal-to-noise ratios. Overall, fine-tuning Whisper not only improved the general performance of the model but also strengthened its ability to transcribe various Norwegian dialects. By fine-tuning the model on the Nynorsk transcriptions of the NPSC audio, the ability of Whisper to transcribe any Norwegian dialect to Nynorsk was improved remarkably. Lastly, Whisper also proves to be robust against low levels of noise with an SNR of 20 dB or greater. It also performs adequately at high levels of noise with an SNR of 10 dB. The ability to transcribe speech in the presence of noise is of great importance to the Furhat robot, which the models should be used in.

Section 5.2 continued by comparing the fine-tuned models with other models. The models were first compared with a small Whisper model that was fine-tuned on Norwegian by the National Library. Even though the small model has a shorter inference time, the overall performance was inferior to the fine-tuned, medium-sized models based on the WER obtained on the test split of the NPSC dataset in Bokmål. The models were further compared with two Wav2Vec 2.0 models that were fine-tuned on the NPSC dataset by the National Library on both Bokmål and Nynorsk. The WERs obtained on the test split by the Wav2Vec 2.0 models are higher than the error rates obtained by Whisper and are worse than the WERs reported by the authors. However, this is most certainly due to the pre- and post-processing steps being different from the steps used in this thesis.

Lastly, Section 5.3 provided a brief overview of the experiments that were conducted in order to find the best-performing models. In the end, the baseline models were fine-tuned using a learning rate of 1e−4 and a step size of 8000. Regularisation did not improve the WER obtained on the validation split and was therefore disregarded.

Chapter 6

Discussion and Conclusion

The purpose of this chapter is to provide an overview of the thesis and to discuss the main findings. Section 6.1 summarises the main goal of the thesis and briefly reviews the main chapters. Section 6.2 discusses the findings based on the results presented in Chapter 5 and the research questions defined in Chapter 1. Lastly, Section 6.3 concludes by providing an overview of possible research topics related to this thesis.

6.1 Summary

This thesis investigated if the Transformer-based [38] speech recognition system Whisper [31] is a suitable candidate for replacing the default ASR system used in the Furhat robot at NorwAI. The research conducted in this thesis was motivated by the prevalent issues with the current systems in terms of bad dialect performance, high noise susceptibility, and the inability to transcribe directly to Nynorsk. Due to the importance of an adequate inference time as well as a high transcription quality, the large and small Whisper models were disregarded. Instead, the medium-sized model was picked and fine-tuned on a high-quality dataset, which was then analysed concerning the performance on Nynorsk and Bokmål, noise robustness, and various speaker characteristics, including age, gender, and dialect.

The first chapter provided an overview of the requirements and challenges of the current speech recognition system used in the Furhat robot, which is the primary motivation for this thesis. It also introduced Whisper as a potential solution to the shortcomings of the current system. The chapter concluded by listing the research questions guiding this thesis.

Chapter 2 focused on the core concepts relevant to understanding the underlying architecture of Whisper, including the digitisation of analogue acoustic signals and spectrograms, early approaches to speech recognition, as well as the fundamentals of neural networks. Moreover, it introduced a range of neural network architectures, including RNNs, autoencoders, sequence-to-sequence models, as well as transformers, and discussed the concept of attention in neural networks.

Building on the fundamentals of Chapter 2, Chapter 3 presented a range of end-to-end speech recognition architectures. It discussed early approaches using a combination of HMMs and neural networks, early CTC-based architectures, as well as modern, Transformer-based models with a particular emphasis on Wav2Vec 2.0 [4] and Whisper [31].

Chapter 4 delineated the methodology used for fine-tuning the medium-sized

Whisper model. It discussed the NPSC [35] dataset and how the data was processed before training. In addition, the chapter also introduced the word error rate, which is the primary evaluation metric.

Finally, Chapter 5 presented the results of running the fine-tuned models on the NPSC test dataset. The results were analysed in detail concerning the length of the sentence, dialect, age, and gender of the speaker, the performance of the model in the presence of noise, and how well the model handles names and abbreviations. Chapter 5 also compared the model to a small Whisper model and a Wav2Vec 2.0 model, which were both fine-tuned by the National Library in Norway. The chapter concluded by providing an overview of the experiments conducted to find the best-performing model.

6.2 Discussion of the Findings

Chapter 5.1 started by comparing the WER of the medium-sized, baseline Whisper models with the fine-tuned models. The WER of the baseline models was quite high, in particular on Nynorsk. The model achieved a mean WER of 67.77 with a standard deviation of 292.58, indicating a substantial fluctuation in WER across the test data. In comparison, the same model has a mean WER of 37.55 on Bokmål, which is 30.22 lower than the WER on Nynorsk. Further analysis of the predicted transcriptions by the baseline model revealed that the speeches were mostly transcribed to Bokmål and in some cases even Swedish or English (see Table 5.3 and 5.6). This shows that the baseline model was not trained on Nynorsk for multilingual speech recognition. After fine-tuning the model, the WER on Nynorsk improved immensely from 67.77 to 11.53, which is a difference of -56.24. What is more, the standard deviation was also reduced to 18.09, indicating a more stable model performance on the test set. Table 5.3 also showed that the transcriptions by the model are generally in Nynorsk and the model no longer switches to English in the presence of names as shown in Table 5.6, even if the names are not transcribed correctly. Consequently, with regard to the first research question **RQ1**, it can be concluded that the WER can be reduced considerably by fine-tuning the model on a high-quality dataset with annotations in Nynorsk and the transcriptions generated by the model are generally in Nynorsk. Nonetheless, the Nynorsk model was tested only on the test split of the NPSC dataset and further testing is required to see if the model generalises to other datasets as well.

Apart from Nynorsk, the fine-tuned model also achieved a mean WER of 10.06 on the test split of the NPSC dataset, which is much lower than the WER of 67.77 by the baseline model. Fine-tuning the baseline model decreased the WER by -27.49. Moreover, the WER is also more stable with a standard deviation of 16.58 compared to 222.01 of the baseline model. Thus, to answer the second research question **RQ2**, the WER of the baseline model on Norwegian Bokmål was improved by fine-tuning the model on a high-quality dataset. What is more, the authors of Whisper [31] correctly assumed that the performance of the model on low-resource languages, such as Norwegian, can be improved by fine-tuning it on a high-quality dataset.

The third research question **RQ3** aimed at analysing if the WER of the fine-tuned Whisper model is in any way increased by the dialect, age, or gender of the speaker. To begin with, fine-tuning the model greatly enhanced the performance of Whisper in the different Norwegian dialects. The WER of the baseline model in Bokmål

varied from about 24 to 40 and 54 to 75 in Nynorsk and the models performed worst on the Western and Eastern dialects. Fine-tuning the models reduced the performance gap between the dialects, resulting in a WER of 7 to 13 on Bokmål and 9 to 13 on Nynorsk. Even though the overall WER of the fine-tuned models is lower for all dialects, a small gap of 6 and 4 remains. However, the gap increases in size when computing the WER based on the electoral district the speakers are assigned to, which is a more granular analysis of the dialect performance. In Bokmål, the fine-tuned model achieves WERs ranging from 4 to 17 and 6 to 17 in Nynorsk. Both models struggle most with the district of Sogn og Fjordane and work best with speakers from the district of Hedmark. While the overall dialect performance has been significantly improved by fine-tuning the model, differences still exist and Whisper does not work independently of the dialect. It is essential to know that some districts are poorly represented in the NPSC dataset and a larger dataset may result in a much smaller gap in WER. With regard to the age of the speakers, the WER varies quite a bit on both models. As shown in Figure 5.10(a) for Bokmål and Figure 5.10(c) for Nynorsk, the models have quite different WERs across all age groups. For instance, while speakers in the age group between 55 and 64 have a mean WER that exceeds the mean WER, the age groups 57, 58, and 62 have a lower-than-average WER although all of the groups have a high training data percentage. What is more, while the age bins $(59, 63]$ and $(63, 69]$ in Figure 5.10(b) and (d) have a higher than average WER, the age bins of $(69, 73]$ and $(73, 79]$ have a lower than average WER. Hence, it cannot be concluded that the models generally perform worse for older groups. While the WER varies across different age groups, the differences are most likely due to the dialect of the speaker and not because of their age. Lastly, the models were analysed regarding their performance on the gender of the speakers. The models have a WER of 10.65 in Bokmål and 12.01 in Nynorsk for male speakers and a respective of 9.18 and 10.81 for female speakers. The models have thus a slightly better WER on female speakers. Nevertheless, the differences are with 1.47 and 1.20 relatively small. All in all, while the performance of Whisper is affected by the dialect of the speaker, it does not appear to be affected by the age or gender of the speaker judged by the age- and gender-specific WERs.

The objective of the fourth research question **RQ4** was to determine if the WER of the fine-tuned Whisper model is increased by real-world noise. Table 5.8 showed that an SNR of 20 and 10 dB only slightly increased the WER. For Bokmål, the WER increased from 10.06 to 10.58 with an SNR of 20 dB and 13.59 using an SNR of 10. For Nynorsk, the WER increased from 11.53 to 12.08 and 16.32 respectively. SNRs smaller than 10 dB cause the WER to increase rapidly. This fits well with the observations made by the authors of Whisper. Whisper was analysed concerning the exposure to white noise and pub noise using different SNRs. It was observed that the WER remained stable for SNRs greater than 20 dB, while the WER started to increase slightly at an SNR of 10 dB and rapidly at SNRs smaller than 10 dB [31]. An SNR of 10 dB is considered high noise according to the authors of Whisper [31]. To conclude, the WER of the fine-tuned Whisper model increases if exposed to high noise with an SNR of 10 dB or lower, while the WER remains stable in case of low-noise exposure with an SNR of 20 dB or greater.

Finally, the objective of research question **RQ5** was to investigate if names and abbreviations cause the WER of the fine-tuned Whisper model to rise. As shown in Table 5.4, the fine-tuned models have in general a much lower WER on both

sentences with and without names/abbreviations than the baseline models. However, the WER on sentences without names/abbreviations is still 3.11 and 3.01 lower in Bokmål and Nynorsk. Even though the gap is quite small for the fine-tuned models, the larger gap of 30.72 in the case of the baseline model in Bokmål indicates that the fine-tuned models most certainly will drop in performance when exposed to names/abbreviations that were not present in the training split of the NPSC dataset. While the gap was much smaller when running the baseline model on Nynorsk, it is essential to note that the baseline model was not trained on Nynorsk for multilingual speech recognition, which also explains the significantly higher WERs of the model. To sum up, names and abbreviations cause the WER of the fine-tuned Whisper model to increase.

6.3 Future Work

All in all, the fine-tuned, medium-sized Whisper models are promising, in particular with the overall WER on both Bokmål and Nynorsk and the performance in the presence of real-world noise. What is more, even though some dialects cause the model performance to drop, the fine-tuned models generally work well across the different dialects. Nonetheless, the discussion in the previous section shows that there is still room for improvement. To begin with, the fine-tuned models were only tested on the test split of the NPSC dataset. Still, to test the generalisation capability of the models, they should also be tested on other datasets from other domains, such as day-to-day conversations, radio and television broadcasts, or podcasts. This applies particularly to the Nynorsk model as the baseline model originally does not support Nynorsk for multilingual speech recognition. The speeches given at the parliament often discuss topics and use words that may not be common in day-to-day interactions. Hence, while the model achieves a relatively low WER on the NPSC data, it might not perform as well once deployed to the Furhat robot because the way the robot is interacted with differs from what the model was trained on. What is more, while different age groups were represented in the NPSC dataset, it was not tested on younger speakers, such as children and teenagers.

Aside from that, the NPSC dataset is in general relatively small compared to the massive amount of data the baseline Whisper models were trained on in English. The models should ideally be fine-tuned on a dataset that not only is larger in size but also covers as many parts of the country as possible to have a strong representation of the majority of the Norwegian dialects to obtain a more even performance across the electoral districts and dialects. What is more, a larger dataset might decrease the overall WER even further as the model is exposed to a broader set of words during training.

Thirdly, the fine-tuned models need to be deployed and tested on the Furhat robot to determine if the issues discussed in Chapter 1.1 have been resolved without causing a substantial increase in query time. The fine-tuned models should be compared to the default ASR model of the robot using a set of pre-defined queries and different levels of noise.

Finally, although the word error rate is the default metric for comparing speech recognition models, it is far from ideal as it often classifies transcriptions as incorrect even though the transcription would have been understood by a human reader. This issue was also pointed out by the authors of Whisper [31]. The sentences often need

to be pre-processed extensively to avoid predicted words being classified as incorrect due to the inclusion of a dot, comma, or capital letter. Moreover, the WER is also relatively strict if the words are not identical even though a human reader would have no issues understanding it. An example is given in Table 5.2. The ground truth sentence is ”derfor trenger vi verdens beste skole” and the prediction by the baseline model is ”da fortrenger vi verdens beste skole”. While the words ”derfor trenger” were wrongly predicted as ”da fortrenger”, a human reader would still be capable of understanding the sentence.

Bibliography

[1] A. Ali and S. Renals, "Word error rate estimation for speech recognition: E-WER," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 20–24. DOI: 10.18653/v1/P18-2004. [Online]. Available: https://aclanthology.org/P18-2004.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, *et al.*, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16, New York, NY, USA: JMLR.org, 2016, pp. 173–182.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016. DOI: 10.48550/arXiv.1607.06450. [Online]. Available: http://arxiv.org/abs/1607.06450.

[4] A. Baevski, Y. Zhou, A. Mohamed, *et al.*, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, *et al.*, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.0473.

[6] C. Bartneck and J. Forlizzi, "A Design-Centred Framework for Social Human-Robot Interaction," in *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759)*, 2004, pp. 591–594. DOI: 10.1109/ROMAN.2004.1374827.

[7] T. Belpaeme, J. Kennedy, A. Ramachandran, *et al.*, "Social Robots for Education: A Review," *Science Robotics*, vol. 3, no. 21, eaat5954, 2018. DOI: 10.1126/scirobotics.aat5954. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.aat5954. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aat5954.

[8] R. van den Berghe, J. Verhagen, O. Oudgenoeg-Paz, *et al.*, "Social Robots for Language Learning: A Review," *Review of Educational Research*, vol. 89, no. 2, pp. 259–295, 2019. DOI: 10.3102/0034654318821286. eprint: https://doi.org/10.3102/0034654318821286. [Online]. Available: https://doi.org/10.3102/0034654318821286.

[9] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach* (The Springer International Series in Engineering and Computer Science), en, 1994th ed. Dordrecht, Netherlands: Springer, Oct. 1993.

[10] S. Bruch, X. Wang, M. Bendersky, *et al.*, "An Analysis of the Softmax Cross Entropy Loss for Learning-to-Rank with Binary Relevance," in *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, ser. ICTIR '19, Santa Clara, CA, USA: Association for Computing Machinery, 2019, pp. 75–78, ISBN: 9781450368810. DOI: 10.1145/3341981.3344221. [Online]. Available: https://doi.org/10.1145/3341981.3344221.

[11] J.-J. Cabibihan, H. Javed, M. Ang, *et al.*, "Why Robots? A Survey on the Roles and Benefits of Social Robots in the Therapy of Children with Autism," *International Journal of Social Robotics*, vol. 5, no. 4, pp. 593–618, Nov. 1, 2013. DOI: 10.1007/s12369-013-0202-2. [Online]. Available: https://doi.org/10.1007/s12369-013-0202-2.

[12] K. Cho, B. van Merriënboer, C. Gulcehre, *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. [Online]. Available: https://aclanthology.org/D14-1179.

[13] A. Conneau, M. Ma, S. Khanuja, *et al.*, "FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech," *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 798–805, 2022.

[14] G. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973. DOI: 10.1109/PROC.1973.9030.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[16] E. Gordon-Rodriguez, G. Loaiza-Ganem, G. Pleiss, *et al.*, "Uses and Abuses of the Cross-Entropy Loss: Case Studies in Modern Deep Learning," in *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, J. Zosa Forde, F. Ruiz, M. F. Pradier, *et al.*, Eds., ser. Proceedings of Machine Learning Research, vol. 137, PMLR, Dec. 2020, pp. 1–10. [Online]. Available: https://proceedings.mlr.press/v137/gordon-rodriguez20a.html.

[17] A. Graves, S. Fernández, F. Gomez, *et al.*, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 369–376, ISBN: 1595933832. DOI: 10.1145/1143844.1143891.

[18] A. Gulati, J. Qin, C.-C. Chiu, *et al.*, *Conformer: Convolution-Augmented Transformer for Speech Recognition*, 2020. arXiv: 2005.08100 [eess.AS].

[19] A. Y. Hannun, C. Case, J. Casper, *et al.*, "Deep Speech: Scaling Up End-to-End Speech Recognition," *CoRR*, vol. abs/1412.5567, 2014. arXiv: 1412.5567. [Online]. Available: http://arxiv.org/abs/1412.5567.

[20] K. He, X. Zhang, S. Ren, *et al.*, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[21] D. Hendrycks and K. Gimpel, "Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units," *CoRR*, vol. abs/1606.08415, 2016. arXiv: 1606.08415. [Online]. Available: http://arxiv.org/abs/1606.08415.

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735.

[23] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge, England: Cambridge University Press, Mar. 2015.

[24] U. Kamath, J. Liu, and J. Whitaker, *Deep learning for NLP and Speech Recognition*, en. Berlin, Germany: Springer, Jun. 2019.

[25] O. Kuchaiev, J. Li, H. Nguyen, *et al.*, *NeMo: A Toolkit for Building AI Applications using Neural Modules*, 2019. arXiv: 1909.09577 [cs.LG].

[26] Z. Li, F. Liu, W. Yang, *et al.*, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, Jun. 2021, ISSN: 2162-237X. DOI: 10.1109/tnnls.2021.3084827.

[27] Z. Lin, M. Feng, C. N. d. Santos, *et al.*, *A Structured Self-attentive Sentence Embedding*, 2017. DOI: 10.48550/ARXIV.1703.03130. [Online]. Available: https://arxiv.org/abs/1703.03130.

[28] Linguistic Data Consortium, *2000 HUB5 English Evaluation Transcripts*, 2002. DOI: 10.35111/8C95-0C55. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2002T43.

[29] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1412–1421. DOI: 10.18653/v1/D15-1166. [Online]. Available: https://aclanthology.org/D15-1166.

[30] V. Panayotov, G. Chen, D. Povey, *et al.*, "Librispeech: An ASR Corpus Based On Public Domain Audio Books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210. DOI: 10.1109/ICASSP.2015.7178964.

[31] A. Radford, J. W. Kim, T. Xu, *et al.*, "Robust Speech Recognition via Large-Scale Weak Supervision," Dec. 2022. [Online]. Available: http://arxiv.org/abs/2212.04356.

[32] M. Schuster and K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: 10.1109/78.650093.

[33] M. Själander, M. Jahre, G. Tufte, *et al.*, *EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure*, 2019. arXiv: 1912.05848 [cs.DC].

[34] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub, 1997.

[35] P. E. Solberg and P. Ortiz, "The Norwegian Parliamentary Speech Corpus," in *Proceedings of the 13th Language Resources and Evaluation Conference*, 2022. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2022/pdf/2022.lrec-1.106.pdf.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, *et al.*, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[37] E. Trentin and M. Gori, "A Survey of Hybrid ANN/HMM Models for Automatic Speech Recognition," *Neurocomputing*, vol. 37, no. 1, pp. 91–126, 2001, ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(00)00308-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231200003088.

[38] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[39] K. Venås and M. Skjekkeland. "Dialekter i Noreg." (2023), [Online]. Available: https://snl.no/dialekter_i_Noreg (visited on 04/14/2023).

Appendices

Appendix A

Conventions and Notations

A.1 Vectors

Vectors are denoted using bold, small letters.

$$\mathbf{v} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{A.1}$$

A.2 Matrices

Matrices are defined using bold, capital letters.

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \tag{A.2}$$

Appendix B

NPSC Dataset

Table B.1: Technical specifications of the NPSC audio files.

Format	Format settings	Bit rate mode	Bit rate	Channels	Sampling rate	Bit depth
PCM	Little / Signed	Constant	1.536 kb/s	2	48 kHz	16 bits

Table B.2: Overview of the features in the *sentence_ data.json* file.

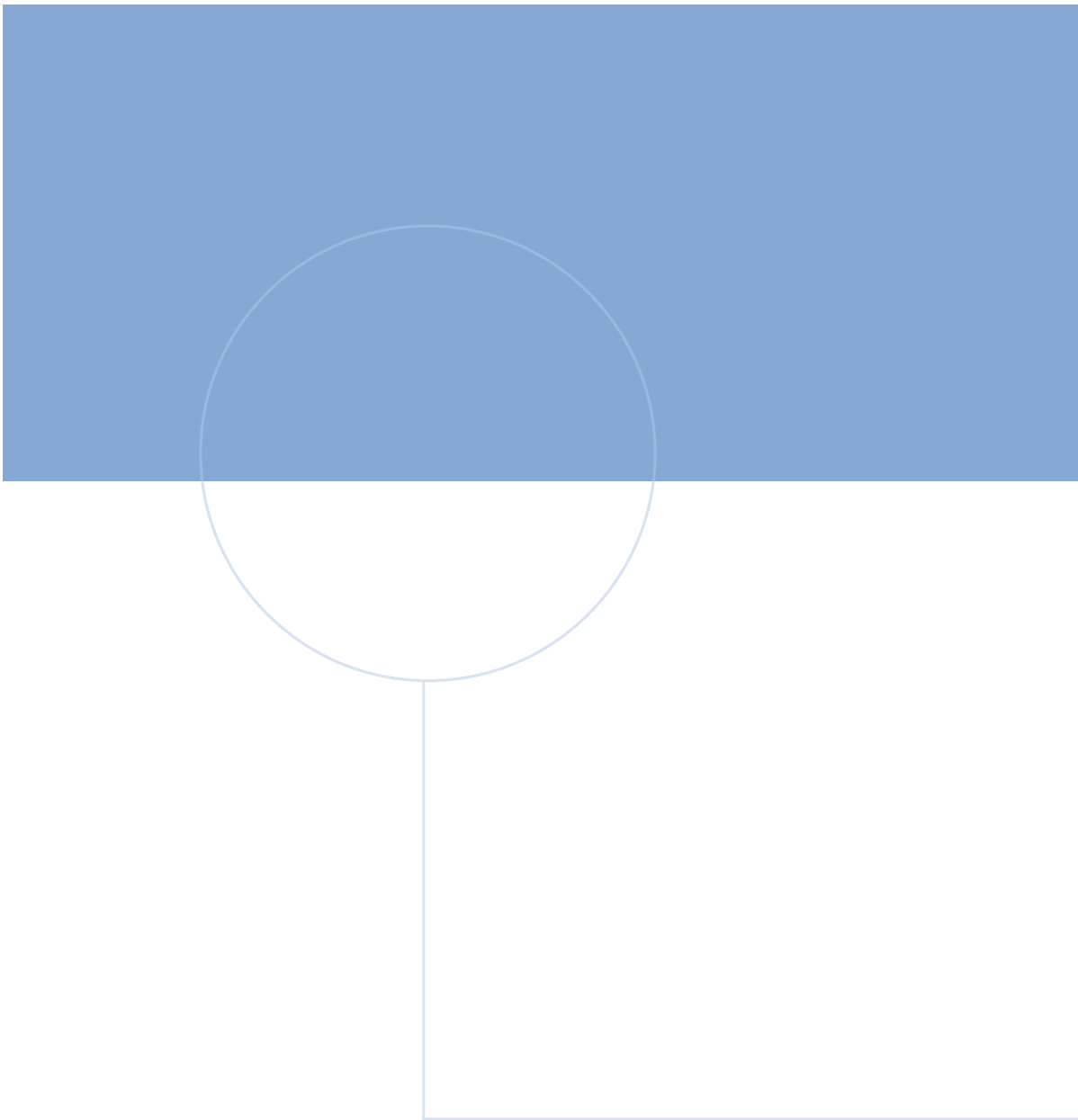
Feature	Description
meeting_date	Date of the meeting (yyyymmdd).
full_audio_file	Name of the audio file covering the entire meeting.
proceedings_file	Name of the official proceedings file.
duration	Duration of the full audio file in milliseconds.
transcriber_id	Id of the transcriber.
reviewer_id	Id of the reviewer.
data_split	Specifies which dataset split the sub-folder belongs to (train, eval or test).
sentences	List containing a set of dictionaries for each sentence. Each dictionary comprises a set of features listed in Table B.3.

Table B.3: All sentence-specific features of the NPSC dataset.

Feature	Description
speaker_name	Name of the speaker.
speaker_id	ID of the speaker.
sentence_id	ID of the sentence.
sentence_language_code	Defines the language of the sentence. Possible values are nb-NO (Bokmål), nn-NO (Nynorsk), and en-US (English).
sentence_text	Non-normalised text transcription of the sentence.
sentence_order	Number indicating the order of the sentences part of the meeting.
audio_file	Name of the audio file in the accompanying audio folder belonging to the sentence.
start_time	Start time of the sentence in milliseconds.
end_time	End time of the sentence in milliseconds.
normsentence_text	Normalised text transcription of the sentence.
transsentence_text	Normalised machine translation of the transcribed sentence. If the manual transcription is in Bokmål, the machine-translated sentence is in Nynorsk. If the manual transcription is in Nynorsk, the machine-translated sentence is in Bokmål.
translated	True if the sentence has been machine-translated, otherwise false.

Table B.4: Overview of all features from the *NPSC_speaker_data.json* file accompanying the NPSC dataset.

Feature	Description
speaker_id	ID of the speaker.
speaker_name	Name of the speaker.
speaker_URI	Wikidata URI of the speaker.
date_of_birth	Date of birth of the speaker.
place_of_birth	Name of the birthplace of the speaker. Null if not found in Wikidata.
pob_URI	Wikidata URI of the speaker's birthplace. Null if unavailable.
pob_country	Country where the speaker was born. Null if unknown.
electoral_district	Electoral district the speaker is assigned to. Null if unknown.
ed_URI	Wikidata URI of the electoral district. Null if the electoral district is unknown.
gender	Gender of the speaker. Male or Female.
chosen_language	The chosen written language of the speaker. Possible values include nb-NO or nn-NO.
dialect	Dialect region of the speaker.



NTNU
Norwegian University of
Science and Technology

NorwAI