Sign Language Translation Mobile Application and Open Communications Framework

Deliverable 3.4: Automatic Speech Recognition Component and Models

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101017255

Project Information
Project Number: 101017255
Project Title: SignON: Sign Language Translation Mobile Application and Open Communications Framework
Funding Scheme: H2020 ICT-57-2020
Project Start Date: January 1st 2021

Deliverable Information
Title: Automatic speech recognition component and models
Work Package: WP 3 - Source Message Recognition, Analysis and Understanding
Lead beneficiary: Radboud University
Due Date: 30/06/2023
Revision Number: V1.2
Authors: Aditya Parikh, Louis ten Bosch, Henk van den Heuvel
Dissemination Level: Public
Deliverable Type: Report

Overview: This deliverable provides a comprehensive overview of the automatic speech recognition (ASR) component developed within the SignON project. It focuses on the development and implementation of ASR models for the supported languages, namely English, Spanish, Dutch, and Irish,

using a modular/hybrid approach with Kaldi ASR, as well as an end-to-end deep learning approach using wav2vec 2.0 and Whisper ASR. The document offers detailed insights into the audio, text, and pronunciation lexicon datasets utilized during the training of the ASR models. Additionally, it presents a comprehensive outline of the training procedures with Kaldi ASR, including the fine-tuning process with wav2vec 2.0. The deliverable evaluates the performance of the ASR model developed through the modular Kaldi approach and compares its Word Error Rate (WER) with both Kaldi and end-to-end systems i.e. wav2vec 2.0 and Whisper ASR. Finally, it explores the deployment of the ASR models and their integration with the SignON orchestrator system.

Revision History

Version #	Implemented by	Revision Date	Description of changes
V0.1	Aditya Parikh	30/05/2023	Initial Draft.
V1.0	Aditya Parikh, Henk van den Heuvel, Louis ten Bosch	19/06/2023	Finished the first draft.
V1.2	Aditya Parikh, Henk van den Heuvel, Louis ten Bosch	26/06/2023	Implemented feedback from the consortium members.

The SignON project has received funding from the European Union's Horizon 2020 Programme under Grant Agreement No. 101017255. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the SignON project or the European Commission. The European Commission is not liable for any use that may be made of the information contained therein.

Approval Procedure

Version #	Deliverable Name	Approved by	Institution	Approval Date
V1.0	D3.4	Aoife Brady	DCU	23/06/2023
V1.0	D3.4	Marco Giovanelli	FINCONS	21/06/2023
V1.0	D3.4	Adrián Núñez-Marcos	UPV/EHU	22/06/2023
V1.0	D3.4	John O'Flaherty	MAC	20/06/2023
V1.0	D3.4	Euan McGill	UPF	21/06/2023
V1.0	D3.4	Irene Murtagh	TU Dublin	21/06/202x
V1.0	D3.4	Ellen Rushe	TCD	23/06/202x
V1.0	D3.4	Jorn Rijckaert	VGTC	21/06/2023
V1.2	D3.4	Henk van den Heuvel	RU	20/06/2023
V1.0	D3.4	Bram Vanroy	KU Leuven	23/06/2023
V1.0	D3.4	Rehana Omardeen	EUD	22/06/2023
V1.0 V1.1	D3.4	Mirella De Sisto Dimitar Shterionov	TiU	20/06/2023 26/06/2023

Acronyms

The following table provides definitions for acronyms and terms relevant to this document.

Acronym	Definition
ASR	Automatic Speech Recognition
DHH	Deaf or Hard of Hearing
AM	Acoustic Models
LM	Language Models
CGN	Corpus Gesproken Nederlands
G2P	Grapheme-to-Phoneme (conversion)
SAMPA	Speech Assessment Methods Phonetic Alphabet
WER	Word Error Rate
DNN	Deep Neural Network
LVCSR	Large Vocabulary Continuous Speech Recognition
TDNN	Time Delayed Neural Network
CNN	Convolutional Neural Network
MFCC	Mel Frequency Cepstral Coefficient
HMM	Hidden Markov Model
GMM	Gaussian Mixture Model
LDA	Linear Discriminative Analysis
MLLT	Maximum Likelihood Linear Transformation
fMLLR	FeatureSpace Maximum Likelihood Linear Regression

SAT	Speaker Adaptive Training
MMI	Maximal Mutual Information
LF-MMI	Lattice-Free Maximal Mutual Information
CTC	Connectionist Temporal Classification
WFST	Weighted Finite-state Transducer
Ho-Re-Ca	Hotels-Restaurants-Cafe
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
API	Application Programming Interface
REST	Representational State Transfer
JWT	JSON Web Tokens
OAuth2	Open Authorization 2.0
UNIX	Uniplexed Information and Computing System

Table of Contents

1. Introduction

Deliverable D3.4 presents a comprehensive overview of the Automatic Speech Recognition (ASR) models and their integrated components within the SignON project. Over the past two years, we have made significant advancements in developing and enhancing the SignON ASR component. In this deliverable, we will explain the hybrid ASR models that we have currently developed with the Kaldi ASR toolkit[1] and deployed as a web service. Additionally, we will discuss the new developments and improvements in ASR models on the basis of end-to-end deep learning models using wav2vec 2.0[2] and Whisper ASR[3] models. We will also discuss the audio and text data that we have used in training and fine-tuning our ASR models. Furthermore, we will provide a future plan for developing special versions of speech recognizers for the hospitality use case, based on fine-tuned language models, and for Deaf or Hard of Hearing (DHH) speakers based on the ongoing recordings specific to their use case. Next, we will present the performance of the developed ASRs on the benchmarking test sets. Moreover, we will describe the ASR web service that is currently serving the ASR models in the SignON project, as well as the newly developed web service that will serve the end-to-end wav2vec 2.0 and Whisper models. Finally, we provide a brief overview of the latency of the ASR models and discuss our future plan to make the web service leaner and faster.

2. Overview of Automatic Speech Recognition Models

2.1 Hybrid/Modular ASR Models with Kaldi

In the SignON application, the ASR models used for English, Spanish Dutch, and Irish languages are created using a classical modular/hybrid ASR approach using the Kaldi toolkit[4]. This approach breaks down the overall ASR architecture into smaller functional components: acoustic model (AM), language model (LM), and pronunciation lexicon. The modular approach allows us to train our own AM, and LM with a task-dependent customized vocabulary. The use of this modular approach has its advantages, especially for specific domains like different languages, dialects or speakers. We can customize the ASR

[1] https://kaldi-asr.org/

[2]Baevski, Alexei, et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations."Advances in neural information processing systems 33 (2020): 12449-12460.

[3] "Introducing Whisper - OpenAI." 21 Sep. 2022, https://openai.com/research/Whisper.

[4] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Vesely, K. (2011). The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding* (No. CONF). IEEE Signal Processing Society.

models to match the specific domain or task, which improves the system's performance. However, traditional modular ASR methods require a lot of transcribed speech recordings for training, extensive text resources, and detailed grapheme-to-phoneme (G2P) mappings or dictionaries. This can be challenging for low-resource languages that have limited labeled data and a small digital presence.

In the SignON project we use the modular/hybrid ASR approach to benefit from customization and enhanced performance in specific domains. However, we also faced challenges when working with Irish, a low-resource language, due to high data requirements of traditional modular ASR methods.

2.2 End-to-End Deep Learning approach with wav2vec 2.0

Wav2vec 2.0 is a state-of-the-art self-supervised learning framework for speech processing, specifically designed for ASR. Developed by researchers at Facebook AI, it has demonstrated an impressive performance in various speech-related tasks[5]. The core idea behind wav2vec 2.0 is to learn powerful speech representations from large amounts of *unlabeled* speech data. By training on massive audio datasets without explicit transcription, wav2vec 2.0 can effectively capture rich contextual information and acoustic patterns present in speech signals.

Wav2vec 2.0 utilizes a two-step training process. In the first step, a "contrastive" self-supervised learning objective is employed, where the model learns to differentiate between true quantized latent speech representation for a masked time step within a set of distractors. Quantized refers to the process of selecting the best matching code-word from a predefined codebook to represent a specific latent speech representation, enabling the conversion of continuous speech data into a discrete representation for further processing and analysis. This step helps the model learn meaningful representations that capture the important aspects of speech. The result of this step is a so-called pretrained model.

In the second step, the pretrained model is fine-tuned on labeled transcribed data from specific downstream tasks, such as ASR. This transfer learning approach allows the model to leverage the learned representations to improve performance on the target task with limited labeled data. The result is a finetuned model.

The wav2vec 2.0 framework has shown remarkable results in ASR tasks, even in low-resource languages or settings with limited annotated data. It has proven to be a valuable tool for developing high-performance ASR systems and has contributed to enormous advancements in end-to-end speech processing.

[5]Baevski, Alexei, et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations."Advances in neural information processing systems 33 (2020): 12449-12460. https://doi.org/10.48550/arXiv.2006.11477

We utilized a publicly released pre-trained wav2vec 2.0 model, XLS-R[6] , which was trained on 436k hours of publicly available speech audio. This model is available on Hugging Face. The largest variant has up to 2 billion parameters[7]; intermediate checkpoints having 300 million parameters[8] and 1 billion parameters[9].

2.3 Whisper ASR

Whisper[10] is an alternative to the Kaldi and wav2vec 2.0 models. It is an ASR system trained on 680,000 hours of multilingual and multitask[11] supervised data collected from the web. This model has shown the use of such a large and diverse dataset leads to improved robustness to accents, background noise and technical language. Moreover, it enables transcription in multiple languages including English, Spanish and Dutch (but not Irish), which are target languages in the SignON project. The Whisper ASR models[12] are available in five different variants, each varying in size. These variants range from tiny (39M parameters) and base (74M parameters) to small (244M parameters), medium (769M parameters), and large (1550M parameters). The accuracy of the models increases as the number of parameters grows, enabling improved performance. In our experiments, we have observed that using Whisper models can achieve nearly real-time speech recognition using their different size of models and that can be very beneficial for the hearing users' ASR system. In our evaluation, we utilize the Whisper models without conducting any finetuning. We directly employ the available pretrained models, utilizing them as they are, without making any modifications to their architecture or parameters. Before the end of the SignON project, we will also evaluate the performance of Whisper ASR on the use case recordings from DHH people.

[6] Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., Auli, M. (2022) XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. Proc. Interspeech 2022, 2278-2282, doi: 10.21437/Interspeech.2022-143

[7] https://huggingface.co/facebook/wav2vec2-xls-r-2b

[8] https://huggingface.co/facebook/wav2vec2-xls-r-300m

[9] https://huggingface.co/facebook/wav2vec2-xls-r-1b

[10] Radford, Alec, et al. "Robust speech recognition via large-scale weak supervision." arXiv preprint arXiv:2212.04356 (2022).

[11] In whisper models, multitask refers to its ability to perform tasks such as automatic speech recognition, voice activity detection, and any-to-English speech translation.

[12] https://github.com/openai/whisper

3. Data

In this section, we provide an overview of the datasets that were used in creating the ASR models for typical speech in SignON. For the ASR models, the speech corpora, text corpora and pronunciation lexicons were mainly selected from open-source datasets.

3.1 English ASR

The Kaldi AM for English ASR are trained using the GigaSpeech corpus[13], which is a collection of labeled audio data. It provides 10,000 hours of high-quality transcribed audio from various sources like audiobooks, podcasts, and YouTube. The corpus has different subsets defined on the number of hours of audio data they contain, such as XL (10,000 hours), L (2,500 hours), M (1,000 hours) , S (250 hours), and XS (10 hours). For the AM, we mainly used the 'S' subset with 250 hours of audio data for training. This contains 230,068 utterances. We also experimented with the larger subsets ('M' and 'L'), but they didn't show significant improvements in word error rate (WER) compared to the 'S' subset. For LM, we used the Google One Billion Word Benchmark text corpus[14]. It is a dataset with almost one billion word tokens and contains 400k word types with a frequency of more than 10. The pronunciation lexicons for English were trained based on CMUdict[15] which has 135k entries of the English pronunciation lexicon. For the testing of ASR, we used Gigaspeech "dev" and "test" corpus which consist of 12.5 hours and 40 hours of audio data respectively.

3.2 Spanish ASR

For the Spanish ASR, we utilized the Common Voice Spanish[16] dataset for acoustic modeling. The Common voice dataset includes rich metadata such as speaker age, accent and gender, and consists of 213,244 utterances for training, equating to 313.56 hours of speech material. For building the language model, we utilized the Spanish Billion Words Corpus[17] which has nearly 1.5 billion Spanish tokens and 0.54 million types with a frequency higher than 10. For testing, we used the Common Voice Spanish Dev

[13]Chen, Guoguo, et al. "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio." arXiv preprint arXiv:2106.06909 (2021).
[14] Chelba, Ciprian, et al. "One billion word benchmark for measuring progress in statistical language modeling." arXiv preprint arXiv:1312.3005 (2013).
[15] Weide, Robert. "The Carnegie Mellon pronouncing dictionary." release 0.6, www. cs. cmu. edu (1998).
[16] https://commonvoice.mozilla.org/es
[17]Cristian Cardellino: Spanish Billion Words Corpus and Embeddings (March 2016), https://crscardellino.github.io/SBWCE/

and Test sets, which consist of 26.1 and 25.9 hours of speech, respectively. For pronunciation lexicons we used a dedicated G2P tool based on SAMPA (Speech Assessment Methods Phonetic Alphabet)[18].

3.3 Irish ASR

Acquiring speech data for the Irish language was a significant challenge due to the scarcity of open-source resources available for this language. To tackle this problem, we combined multiple open-source Irish datasets. For the AM training we utilized the Common Voice Irish[19] dataset. We used only the validated utterances from this dataset and excluded those that were part of the test set. Additionally, we used the Living Audio[20] dataset which contributed an additional hour of Irish speech data. We also incorporated all Irish utterances from the Google Fleurs[21] dataset. By combining all three datasets, we were able to train on a total of 9,274 utterances equating to 13.5 hours of speech. For testing, we used the Common Voice Irish Test set, containing 513 utterances (0.5 hours of speech), and a set of 'Invalidated' Common Voice Irish utterances, with 282 utterances (0.3 hours of speech, after removing speech samples with background noise or no speech). The invalidated clips in the Common Voice dataset are the clips that have received more downvotes than upvotes.

For LM, we used the CC-100: Monolingual datasets from Web Crawl Data[22], which includes data for over 100 languages including Irish, with 84 million word tokens and 0.12 million word types having frequency higher than 10. Lastly, for the experiments with Kaldi ASR, we trained a G2P model using 13,300 seed Irish pronunciations acquired from Wikipron[23].

[18]Raškinis, Arimantas-Juvencijus, Gailius Raškinis, and Asta Kazlauskienė. "SAMPA (speech assessment methods phonetic alphabet) for encoding transcriptions of Lithuanian speech corpora." Informacinės technologijos ir valdymas= Information technology and control. Kaunas: Technologija., 2003, T. 29, nr. 4 (2003).

[19] https://commonvoice.mozilla.org/ga-IE

[20] https://github.com/Idlak/Living-Audio-Dataset

[21]Conneau, Alexis, et al. "Fleurs: Few-shot learning evaluation of universal representations of speech." 2022 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2023.

[22]CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data, Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, Edouard Grave, Proceedings of the 12th Language Resources and Evaluation Conference (LREC), p. 4003-4012, May 2020

[23]Lee, Jackson L., et al. "Massively multilingual pronunciation modeling with WikiPron." Proceedings of the 12th language resources and evaluation conference. 2020.

3.4 Dutch ASR

For Dutch ASR, the AM audio data were taken from the Spoken Dutch Corpus[24] (Corpus Gesproken Nederlands, CGN). For AM training of Dutch ASR, we used all the components of CGN corpus spoken by Northern Dutch speakers (the Netherlands) which was around 483 hours of speech. Apart from that for the LM, text corpus from CGN and Twents Nieuws Corpus[25] (530 million words) was used. For testing of Dutch ASR, we considered some speech from CGN corpus spontaneous speech (DevS) and telephonic conversations (DevT) which is around 0.5 hours and 1.97 hours of speech. Apart from that, we also considered the Common Voice Dutch[26] test set in which the speech was specifically tagged as Netherlands Dutch for testing. Those were 3512 utterances which account for 4.4 hours of speech data. The pronunciation lexicons of Dutch were acquired from the CGN lexicon. The CGN lexicon comprises almost all types (unique word forms) that occur in the CGN corpus. The CGN lexicons provide pronunciation for 181k Dutch words.

For fine-tuning the Dutch ASR model with wav2vec 2.0 XLS-R, we used a combined dataset from the CGN corpus, which includes both Northern Dutch (the Netherlands) and Southern Dutch (Flanders, Belgium) languages, along with the Common Voice Dutch dataset.

A similar Kaldi based system for southern Dutch speakers (Flanders, Belgium), i.e. Flemish ASR, was kindly provided by our colleagues at KU Leuven. Since the latter is not an official deliverable for the SignON project, we describe it in an appendix (A2).

It is also important to note that the currently deployed ASR web service with Kaldi models is serving Northern Dutch and Southern Dutch ASR separately, and that is why there can be a regional bias towards Northern Dutch in Dutch ASR.

4. ASR Model Training and FineTuning Procedure

In this section, we provide an overview of the ASR model training procedure using both the modular and end-to-end approaches. As described earlier the modular training approach involves separate component training such as the AM, LM, and pronunciation lexicons. Additionally, we describe the

[24]Oostdijk, N.H.J. 2002, Article in monograph or in proceedings (Peters, P.; Collins, P.; Smith, A. (ed.), New frontiers of corpus research. Papers from the twenty first international conference on English language research on computerized corpora, Sydney 2000, pp. 105-112)

[25]Ordelman, Roeland JF. "Twente nieuws corpus (TwNC)." Parlevink Language Technology Group. University of Twente (2002).

[26] https://commonvoice.mozilla.org/nl

fine-tuning process specifically for end-to-end wav2vec 2.0 models. Moreover, we explain the steps to incorporate an LM head on the pre-trained wav2vec 2.0 model that has undergone fine-tuning.

4.1 Modular ASR training

For English, Spanish, Dutch and Irish ASRs a deep neural network (DNN) based large vocabulary continuous speech recognition (LVCSR) system was developed using the open-source Kaldi speech recognition toolkit.

4.1.1 Acoustic Modelling

For the AM, a time-delayed neural network[27] (TDNN) with the front end of the Convolutional Neural Network (CNN) model was trained on the training datasets. The acoustic model training is done in the Kaldi toolkit using the NNET3 library for CNN-TDNN. In English, Spanish and Dutch ASR the same Kaldi Librispeech recipe[28] is used. As Irish is an under-resourced language, the acoustic model training script is adapted from Kaldi Mini Librispeech recipe[29]. Both recipes follow a similar training pattern, but the hyperparameters such as the number of leaves, number of Gaussians, neural network size, L2 regularization, learning rate, and the number of epochs were optimized to fit the smaller volume of data available.

Prior to training the DNNs, the data underwent initial alignment using the triphone GMM-HMM method and were trained using Mel Frequency Cepstral Coefficient (MFCC) features. Additionally, various techniques such as Linear Discriminative Analysis (LDA), Maximum Likelihood Linear Transformation (MLLT), feature space maximum likelihood linear regression (fMLLR), and speaker adaptive training (SAT) were applied to enhance the data quality. The resulting feature vectors comprised a concatenation of a 40-dimensional high-resolution MFCC feature vector and a 100-dimensional extracted i-vector.

In the Kaldi speech recognition toolkit, there are two commonly used data augmentation techniques for speech data i.e. speed perturbation[30] and spectral augmentation (SpecAugment)[31]. Both of these on-the-fly data augmentation techniques work during the training improving the flexibility of training

[27] Peddinti et al. "A time delay neural network architecture for efficient modeling of long temporal contexts." In the Sixteenth Annual Conference of the International Speech Communication Association, 2015.
[28] https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech/s5
[29] https://github.com/kaldi-asr/kaldi/tree/master/egs/mini_librispeech/s5
[30]Ko, Tom, et al. "Audio augmentation for speech recognition." Sixteenth annual conference of the international speech communication association. 2015.
[31] Park, Daniel S., et al. "Specaugment: A simple data augmentation method for automatic speech recognition." arXiv preprint arXiv:1904.08779 (2019).

and helping in reducing the required disk space. The speed perturbation technique creates three copies of training data with the speed warping factor of 0.9, 1.0, and 1.1. By using the SpecAugment tool, the time and frequency domain bands of log mel-spectrogram of the single training speech data are randomly masked up to certain extents. These methods are known to yield an improvement in ASR scores. The TDNN model is trained with the Lattice-Free Maximal Mutual Information (LF-MMI) criterion. This MMI function, simply put, increases the probability of the reference transcript of the particular audio file while decreasing the probability of all other transcripts. This method, like Connectionist Temporal Classification (CTC), uses a sentence level posterior for training the neural network but unlike end-to-end approaches, still, loosely relies on alignments from a pretrained HMM-GMM model. In this TDNN training, a regularization technique 'leaky HMM' is used to deal with overfitting due to sequential training. A 'leaky HMM' allows the transition probability from each stage to every other stage to ensure gradual forgetting of context. It is equivalent to stopping and restarting the HMM with some probability on each frame. The leaky-hmm-factor is set to 0.1. The dropout is different at various points of training. From the beginning of training to 20% of training the dropout ratio is 0 and increases linearly to 50% at the halfway point of training and again decreases to 0 linearly at the end of training.

4.1.2 Language Modelling

The LM used in the Kaldi hybrid system is a statistical n-gram model that has been trained on the text data described in section 3. This training process was carried out using the SRILM toolkit[32]. Generally, employing a larger LM helps generate more accurate search lattices. However, due to memory limitations on the SignON platform, we performed LM pruning. This involved using the prune option in SRILM to remove n-gram probabilities that had minimal impact on the model's perplexity, relative to a specified threshold. To construct the language models, words were selected based on their frequency in the raw text materials. Words below a certain threshold frequency were replaced with '<UNK>' tokens, representing unknown words.

Before using the text for LM, the below operations were performed on the text to standardize:

1. Split sentences on sentence-ending punctuation symbols.
2. Removed all types of special symbols and punctuations present in the text.
3. Removed all white spaces except for in-between-word spaces.
4. Converted all numbers to words using the num2words[33] library in English, Spanish and Dutch.

[32] Stolcke, Andreas. "SRILM-an extensible language modeling toolkit." Seventh international conference on spoken language processing. 2002.
[33] https://pypi.org/project/num2words/

5. Removed all numbers present in the Irish text.
6. Removed whitespace characters that come before an apostrophe.
7. Replaced any successive whitespace characters with a space.
8. Converted all text to lowercase.

4.1.3 Pronunciation Lexicons

For English and Irish language, we used a Sequence-to-Sequence G2P toolkit from CMU sphinx[34], on the basis of the pronunciation lexicon entries mentioned in Section 3. The G2P tool performs G2P conversion using a transformer model from a tensor2tensor toolkit[35]. The transformer model deliberately avoids using recurrence and depends completely on an attention mechanism to create global dependencies between input and output. For Spanish ASR, we used another dedicated G2P tool[36] based on SAMPA phonetic transcriptions of tokens.

4.2 Fine Tuning End-to-End Models with XLS-R

We utilized a publicly released pre-trained wav2vec 2.0 model, XLS-R[37], which was trained on 436k hours of publicly available speech audio and is available on Hugging Face. During its self-supervised pre-training, XLS-R learned contextualized speech representations by randomly masking feature vectors and passing them through a transformer network. For fine-tuning on our speech recognition task, we added a single linear layer on top of the pre-trained network and fine tuned the model on our labeled speech data. We used the 300 million-parameter version of XLS-R, which is among the smaller versions (currently, model sizes range from 300 million to 2 billion parameters). The fine-tuning was performed on an NVIDIA Tesla T4 GPU using the Adam optimizer, with a learning rate starting with a warm-up for 500 steps, peaked at $3e^{-4}$ for all global steps, and then decayed exponentially. The fine tuning script for the end-to-end models can be found in Github repositories[38] [39]. Apart from that the whole fine-tuning process was logged into the Weights and Biases tool[40].

[34] https://github.com/cmusphinx/g2p-seq2seq
[35] https://github.com/tensorflow/tensor2tensor
[36] https://github.com/cristiantg/g2p_spanish
[37] https://huggingface.co/facebook/wav2vec2-xls-r-300m
[38] https://github.com/Aditya3107/wav2vec2-basics/blob/main/run_speech_recognition_ctc.py
[39] https://github.com/Aditya3107/wav2vec2-basics/blob/main/run_w2v2.sh
[40] https://wandb.ai/site

4.2.1 Integrating the Language Model with the Fine-Tuned model

We used the same LM that was used earlier on the modular Kaldi ASR and on an end-to-end approach. These LMs were initially created in ARPA format but were transformed into a binary format using KENLM[41] to decrease the time required to load the models. The integration of the LM with the fine tuned model was performed using shallow fusion through the CTC decoder library pyctcdecode[42]. While most of the hyperparameters were left at their default settings, we did experiments with the *weight* of the LM during shallow fusion by adjusting it between 0 to 1, with intermediate values of 0.1, 0.3, 0.5, 0.7, 0.8, 0.9 and 1.0 to assess its impact on the overall performance. In wav2vec 2.0, the output of the model is represented as the probability distribution of the predicted phonemes (or graphemes) at each time frame (each 20ms) of the input signal. While the model can generate both lexical and non-lexical words through its sequence of phonemes, the use of an LM helps to refine non-lexical predictions by incorporating information about the likelihood of different sequences of phonemes forming words in the language. From our LM weight experiments, the weight of LM was selected at 0.7 at which we could find the least amount of non-lexical words and lowest WER.

5. Performance

The accuracy of an ASR system depends on how well the training data for the AM and LM match the test conditions. That is why it is important to evaluate the ASR system's accuracy in a specific target environment.

The WER is generally used as a standard measure to assess the effectiveness of a LVCSR system. It compares the ASR system's predicted word sequence with a reference transcription, considering errors such as substitutions (S), insertions (I), and deletions (D). For the total number of words (N) in the reference transcription, WER is defined as a percentage (in %; lower is better)

$$WER = \frac{100 * (I+D+S)}{N}$$

In the table below, we present the performance of currently deployed ASRs with the modular Kaldi approach.

[41] https://github.com/kpu/kenlm
[42] https://github.com/kensho-technologies/pyctcdecode

Table 1: Performance of modular ASR trained with Kaldi ASR

Language	Test Set	Hours	Word Error Rate
English	GigaSpeech Dev	12.5	23.69%
	GigaSpeech Test	40	22.67%
Spanish	Common Voice Test	26.1	15.69%
	Common Voice Dev	25.9	13.68%
Dutch	CGN Dev (Telephonic Conversation)	2	28.60%
	Common Voice Test(NL Dutch)	4.4	24.78%
Irish	Common Voice Test	0.5	22.69%
	Common Voice Invalidated	0.3	43.06%

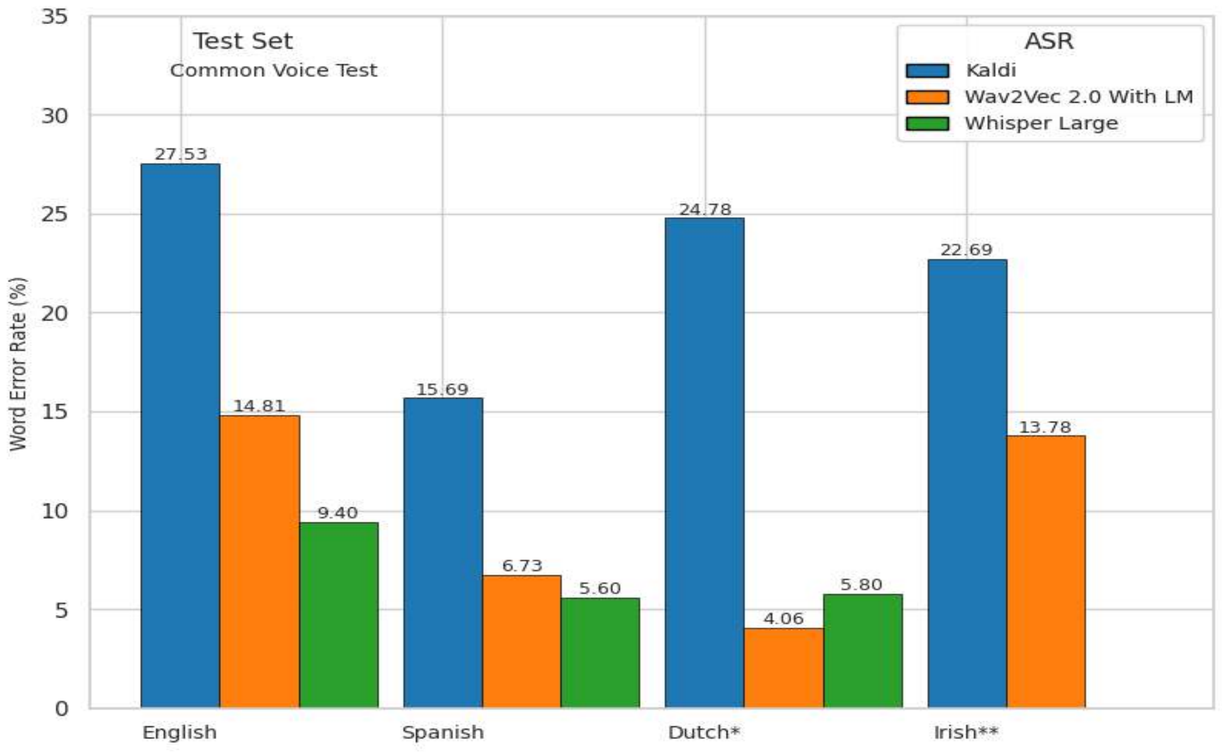
Below, figure 1 compares the performance of the modular ASR (Kaldi) with end-to-end ASR models i.e. wav2vec 2.0 and Whisper ASR. The graph shows that the fine-tuned model using wav2vec 2.0 significantly improves the WER compared to traditional modular Kaldi models. Additionally, the Whisper large model also has a lower WER compared to the wav2vec 2.0 models, except for Dutch. This graph represents a comparison of the performance on a standard test set from the Common Voice datasets, so the numbers could be best compared within each language.

In the performance comparison, we have not compared the commercially available solutions like Google Cloud Speech-to-Text with ASRs created or fine-tuned by us inhouse. Commercial ASR systems like Google's Speech-to-Text have the advantage of being built on extensive speech datasets comprising millions of hours of diverse audio sources. However, the specific details about the training data and models used by commercial ASR providers are often not openly disclosed. On the other hand, open-source ASR models like Whisper and wav2vec 2.0 provide transparency by openly sharing their data, code, and models[43] [44]. Also, commercial speech-to-text services do not publish the benchmark on a

[43] https://github.com/openai/whisper/
[44] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec

common benchmarking corpus. Such benchmarks are available but they are not official. Such performance of commercially available ASR web services is shown in Picovoice's Github repository[45].

Figure 1: Performance of end-to-end deep learning ASR models compared to modular Kaldi models[46] **[47]



6. Fine-Tuning with the Use Case Recordings

In the remaining time of the SignON project, specialized speech recognition models will be created by fine-tuning the pretrained wav2vec 2.0 XLS-R model on the basis of in-domain recordings from the hospitality use case for both typical and atypical speakers (from the DHH population).

Our motivation for using wav2vec 2.0 (instead of Whisper) is that wav2vec 2.0 (currently) is better suited for being fine-tuned to down streaming tasks whereas Whisper is better suitable to use on an as-is basis. The set of models to be eventually employed in the SignOn Mobile application will be decided upon on consortium level during the final six months of the project. This is a consortium-wide decision since the

[45] https://github.com/Picovoice/speech-to-text-benchmark

[46] Dutch wav2vec 2.0 model is finetuned with XLS-R 2 billion model whereas others are finetuned with XLS-R 300 million.

[47] Whisper models do not support Irish language by default.

ASR model will impact the flexibility and usability of the SignOn application for all spoken input. In this process, latency and performance profiles will be taken into account.

This process will involve incorporating audio recordings from DHH individuals gathered via a dedicated SignON data collection application (SignON ML). Furthermore, for the hospitality domain, specific LMs will be developed using text collected from open-source reviews in the hotel, restaurant, and cafe (Ho-Re-Ca) domain. These LMs will be added at the head of the fine-tuned wav2vec 2.0 models.

Additionally, the performance of using Whisper models will be evaluated by analyzing test recordings, which will also be obtained through use case recordings.

7. ASR Web Service

In the SignON application, the ASR component's functionality is implemented through a web service called audioserver (https://restasr.cls.ru.nl/api-docs/). So the ASR is not built within the SignON framework (for further details about the SignON framework architecture, please refer to D2.4[48]). The audioserver is currently serving modular ASR models created with Kaldi ASR, and it acts as a state-of-the-art backend web service for real-time transcription (decoding) of audio files via standard HTTP requests. It offers easy deployment and development by combining Docker, UNIX/Linux, Node.JS Express, MongoDB, and the Kaldi ASR.

In the SignON application, ASR is called through SignON Orchestrator, which then passes the audio data to the SignON WP3 Dispatcher. The SignON WP3 Dispatcher is responsible for making a POST request to the audioserver ASR web service. The SignON application, through the SignON framework, acts as a client to the audioserver web service, enabling it to send and receive information through a JSON REST API. This ensures a flexible and scalable ASR infrastructure, allowing dynamic selection of LM and AM for decoding. The web server maintains a non-relational database to keep track of user interactions with the system.

To ensure data security, the interaction protocol is based on HTTPS, providing encrypted and secure client-server data transmission. Login authentication and secure requests utilize JSON Web Tokens (JWT). Audio files sent to the server are checked for media type and size before upload. The currently allowed audio formats are mp3, wav and m4a, with a maximum file size per audio file of 6MB. Privacy

[48] D2.4 - Intermediate release of the Open SignON Framework: https://signon-project.eu/publications/public-deliverables/

considerations are addressed by completely removing the audio data from temporary storage at the server side after the completion of the transcription.

Next to the aforementioned Kaldi based web service, we have built a second web service for ASR that also includes end-to-end solutions (https://signon-wav2vec2.cls.ru.nl/docs). It is our plan to integrate this new web service into the SignON project. Currently, with wav2vec 2.0, three ASR models are deployed in web service i.e. Spanish, Dutch and Irish, while Whisper models (including Tiny, Base, Small, Medium and Large) are supported for English, Spanish, and Dutch.

The new ASR web service is being developed using the FastAPI framework and includes functionalities such as user registration with email and password, email verification, and OAuth2 authentication for login. Similar to the currently deployed ASR web service, this new development also ensures audio file size and format validation before upload and addresses privacy concerns by removing audio files from the server side from temporary folder after transcription completion.

One final comment relates to latency. Latency in an ASR web service refers to the time delay or response time between sending a request to the ASR web service and receiving the corresponding recognition results. It is essentially the time taken for the ASR system to process the input audio and provide a transcription of the speech. Higher latency means there is a longer delay between sending the audio and getting the results, which can impact real-time applications or user experience. Latency can be influenced by various factors, including the number of parameters and size of the ASR model, the processing power and resources available on the server where the web service is hosted, network conditions, beam size, and the size and duration of the input audio. A low latency is highly desirable for the SignON project.

The graphs provided below show the latency in various cases, and compares the inference times between Kaldi, wav2vec 2.0-based models and different Whisper ASR models. Along the horizontal axis audio file duration is displayed and on the vertical axis the time taken to provide transcription is displayed. From the first figure, it is evident that end-to-end models perform well in terms of low latency for utterances up to 20 seconds. Larger durations, however, are prohibitive in terms of latency. The second plot below shows a specific latency profile for the various Whisper model sizes.

Figure 2: Latency comparison between Kaldi models and end-to-end wav2vec 2.0 (XLS-R) models

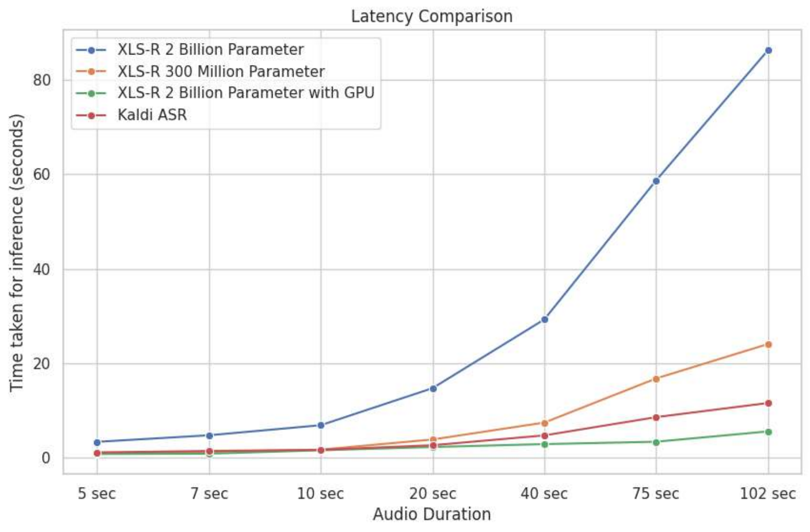
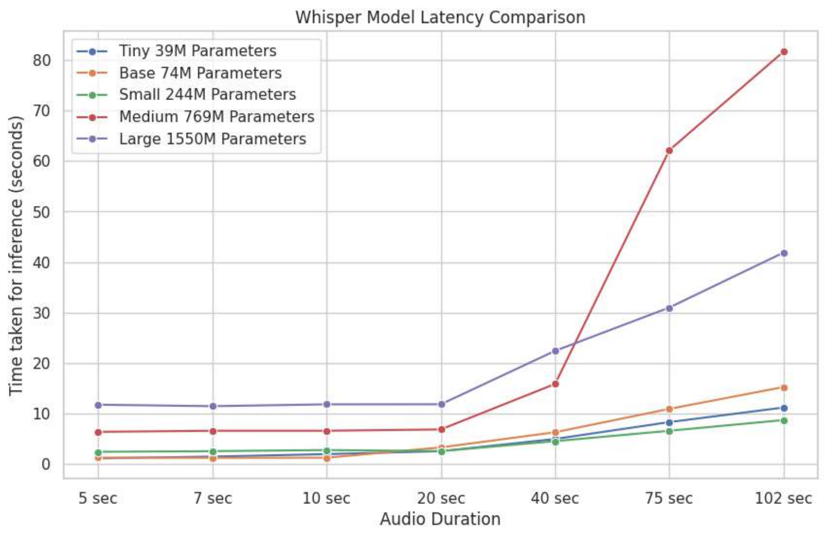


Figure 3: Latency comparison between different Whisper ASR models (Tiny, Base, Small, Medium and Large)



The appendix section of this document contains reference scripts and its corresponding output, which demonstrate how the web services establish connections on the client side of the SignON platform.

8. Conclusion

In this report, we provide a comprehensive overview of the ASR models and their components used in the SignON project. We cover the entire process involving audio, text, and pronunciation datasets, as well as the training of AM, LM, and G2P conversion models for modular/hybrid ASR models.

Furthermore, we highlight the fine-tuning process of ASR models using the wav2vec 2.0 pretrained models, specifically XLS-R. By evaluating the performance of our current ASR models across various benchmarking test sets, we demonstrate the notable advancements achieved compared to traditional methods. Finally, we outline the integration of ASR web services within the SignON application framework, enabling seamless communication.

Appendices

A.1 Client-Side Connection Code and Functionality

The following Github Gist[49] shows the code at the client side to make the connection to the Audioserver web service that currently communicates with the SignON framework. Its basic functionality consists in three steps: (1) login, (2) download/upload file or link and retrieve audio ticket, and (3) decoding and downloading results, in a specific (CTM) format.

The CTM formatted output as shown below, consists of transcribed words, the confidence score for the specific word and the duration at which a particular word was spoken.

```
HELLO NICE TO ME TO
0: {'channel': 1, 'tbeg': 0, 'dur': 0.81, 'word': 'HELLO', 'score': 0.89}
1: {'channel': 1, 'tbeg': 0.95, 'dur': 0.76, 'word': 'NICE', 'score': 1}
2: {'channel': 1, 'tbeg': 1.71, 'dur': 0.15, 'word': 'TO', 'score': 1}
3: {'channel': 1, 'tbeg': 1.86, 'dur': 0.09, 'word': 'ME', 'score': 0.73}
4: {'channel': 1, 'tbeg': 1.95, 'dur': 0.06, 'word': 'TO', 'score': 0.37}
```

The Github Gist[50] presents a new ASR web service designed to communicate with the SignON framework. It provides similar functionalities to the audioserver web service, including login, audio file upload, ASR model selection for transcription, language choice, decoding, and result downloading with word offsets. The word offset output includes transcribed words, confidence scores, word duration at which a particular word is spoken with start time and end time, and the transcript itself.

```
[
  "Words Offsets: [{'word': 'Ik', 'start_time': 0.0, 'end_time': 0.74,
'confidence': 0.732}, {'word': 'probeer', 'start_time': 0.74, 'end_time': 1.22,
'confidence': 0.768}, {'word': 'met', 'start_time': 1.22, 'end_time': 1.44,
'confidence': 0.86}, {'word': 'jou', 'start_time': 1.44, 'end_time': 1.62,
'confidence': 0.944}, {'word': 'te', 'start_time': 1.62, 'end_time': 1.82,
'confidence': 0.895}, {'word': 'communiceren', 'start_time': 1.82, 'end_time':
2.36, 'confidence': 0.949}, {'word': 'via', 'start_time': 2.36, 'end_time': 2.6,
'confidence': 0.892}, {'word': 'deze', 'start_time': 2.6, 'end_time': 2.92,
'confidence': 0.892}, {'word': 'app.', 'start_time': 2.92, 'end_time': 3.2,
'confidence': 0.874}] Transcript:  Ik probeer met jou te communiceren via deze
app."
]
```

[49] https://gist.github.com/Aditya3107/c1ad68600c5738b08845a089a9644928
[50] https://gist.github.com/Aditya3107/ad295fbc2afa757ddc26700a9c6bda11

A.2 Flemish ASR

The Netherlands and the part in Belgium called Vlaanderen (Flanders) have officially the same standard language Dutch. However there are notable differences especially in the pronunciation of phonemes. In the project proposal this distinction between Northern Dutch and Southern Dutch was not made since the Northern Dutch variant was implied. However, we deployed Southern Dutch (Flanders, Belgium) ASR with modular Kaldi ASR. This Kaldi based ASR system for Southern Dutch speakers (Flanders, Belgium) was kindly provided by our colleagues at KU Leuven. Below, we have described the datasets used to train AM, LM and pronunciation lexicons for Flemish ASR.

CGN[51] (Corpus Gesproken Nederlands) is used for Flemish ASR. It is a manually orthographically annotated speech corpus of around 900 hours of contemporary Dutch speech, originating from Flemish and Dutch people. Out of 900 hours, 270 hours correspond to Southern Dutch (Flemish). For AM training, all narrow-band telephone speech and spontaneous conversational speech, which correspond respectively to components C, D, and component A of CGN were excluded. For Flemish ASR, the remaining 155 hours of audio are randomly divided into two sets: CGN-train, which contains 90% of the audio, and CGN-dev, which contains 10% of the audio. For LM, the ESAT 2008[52] N-best n-gram language model was used without further adaptations. The training material for the LM was obtained from two sources: the Dutch publisher PCM (360 million words, media and newspaper texts) and the Flemish Mediargus[53] (1,436 million words).

On the testing datasets mentioned above, the performance of Flemish ASR is as below.

[51] Oostdijk, N.H.J. 2002, Article in monograph or in proceedings (Peters, P.; Collins, P.; Smith, A. (ed.), New frontiers of corpus research. Papers from the twenty first international conference on English language research on computerized corpora, Sydney 2000, pp. 105-112)

[52] Demuynck, Kris, et al. "The ESAT 2008 system for N-Best Dutch speech recognition benchmark." 2009 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, 2009.

[53] Kessens, Judith, and David A. van Leeuwen. "N-best: The northern-and southern-Dutch benchmark evaluation of speech recognition technology." Eighth Annual Conference of the International Speech Communication Association. 2007.

Table2: Performance of Flemish (Southern Dutch) ASR

Language	Test Set	Hours	Word Error Rate
Flemish (Southern Dutch (Flanders, Belgium)	CGN Dev	15.5	14.02%
	N Best 2008 Evaluation[54]	2	10.99%

However, later in the SignON project, the possibility of incorporating Flemish audio and text inputs was dismissed, since as explained above this was not part of the project objectives, but our web services in principle provide the capability to seamlessly integrate a Flemish ASR model into the SignON project. Moreover, the ASR models of wav2vec 2.0 and Whisper deal better with both variants than Kaldi does. Finally, the SignON ML recording app has the extra option to mark and record Dutch from speakers from Flanders, Belgium.

[54] van Leeuwen, David A. "N-Best 2008: a benchmark evaluation for large vocabulary speech recognition in Dutch." Essential Speech and Language Technology for Dutch: Results by the STEVIN programme (2013): 271-288.