# Comparing Modular and End-To-End Approaches in ASR for Well-Resourced and Low-Resourced Languages

**Aditya Parikh Louis ten Bosch Henk van den Heuvel Cristian Tejedor-García**
Centre for Language and Speech Technology,
Radboud University, Nijmegen, The Netherlands
{aditya.parikh,louis.tenbosch,henk.vandenheuvel,cristian.tejedorgarcia}@ru.nl

## Abstract

We present a comparative study of a state-of-the-art traditional modular Automatic Speech Recognition (Kaldi ASR) and an end-to-end ASR (wav2vec 2.0) for a well-resourced language (Spanish) and a low-resourced language (Irish). We created ASRs for both languages and evaluated their performance under different update regimes. Our results show that the end-to-end wav2vec 2.0 outperforms the modular ASR for both languages in terms of Word Error Rate (WER) but performs worst in terms of real-time decoding. We also addressed the issue of non-lexical words in wav2vec 2.0's output. We found that in wav2vec 2.0 by LM integration with shallow fusion and increasing LM weight to 0.7 and 0.8 respectively for the Spanish and Irish provided the optimum ASR performance by reducing non-lexical words. However, this does not eliminate all non-lexical words. Finally, our study found that Kaldi ASR would perform best for real-time decoding for longer audio inputs compared to wav2vec 2.0 model trained on the same dataset on the minimal infrastructure, although wav2vec 2.0's performance can be improved with a GPU acceleration in backend. These results may have significant implications for creating real-time ASR services, especially for low-resourced languages.

## 1 Introduction

Traditional modular ASR frameworks decompose the ASR task into acoustic, pronunciation, and language modeling e.g. Povey et al. (2011). The modular approach of ASR is knowledge-based and provides flexibility in training one's own acoustic model (AM) and language model (LM), in combination with a dedicated customised vocabulary. The knowledge-based modular approach allows adequate performance in specific domains like specific languages, dialects or speakers. A modular ASR can be tailored to the specific domain or task, which can lead to further improvement of the performance of the system (Roy et al., EasyChair, 2022). However, the traditional modular approach of ASR requires a significant amount of transcribed speech recording for training, large text resources, and explicit grapheme-to-phoneme (G2P) mappings or complete dictionaries as basic requirements. This poses a significant challenge for low-resourced languages that do not have a significant digital footprint with a limited amount of labeled data available (Srivastava et al., 2018).

Self-supervised learning (SSL) has emerged as a powerful technique for settings where annotated audio data is scarce. The key idea behind this approach is to learn (pretrained) general representations from substantial amounts of unlabeled source data, and subsequently leverage them to improve the performance (finetuning) on downstream target tasks with a very limited amount of transcribed data. This is particularly useful for tasks such as speech recognition, where obtaining labeled data can be a time-consuming and costly process. Models based on SSL, e.g. wav2vec 2.0 (Baevski et al., 2020), have shown their powerful representation ability and feasibility for ultra-low-resourced speech recognition, making self-supervised end-to-end models a desirable alternative to the flexible and useful modular infrastructure.

This paper aims to evaluate and compare the performance of two different approaches for developing ASR systems: modular Kaldi ASR e.g., Povey et al., 2011 and end-to-end ASR based on wav2vec 2.0 (Baevski et al., 2020), for two languages: Spanish (well-resourced) and Irish (low-resourced). The study not only assesses the performances of both approaches in terms of WER but also addresses challenges with wav2vec 2.0 such as generating non-lexical word forms (such as 'weekent', 'halloo') and the impact of LM weights. Additionally, we examine the latencies and real-time factor (RTF) while deploying both ASRs under the same client-server network environment. In this way, we aim to determine which approach is more effective for developing ASR systems for different languages and resource levels, specifically with minimal infrastructure.

## 2 Related Work

Since the emergence of self-supervised learning methods, various studies showcased the potential of self-supervised end-to-end approaches in speech technology across different languages and modalities (Zuluaga-Gomez et al., 2023; Coto-Solano et al., 2022; Al-Ghezi et al., 2021; Yi et al., 2021). One such study by Zuluaga-Gomez et al. (2023) examines the robustness of two end-to-end models wav2vec 2.0 and XLS-R trained in a new domain, air traffic control (ATC) communications. Their findings show significant reductions in relative WER ranging from 20% to 40% compared to the hybrid-based ASR baseline, indicating the effectiveness of self-supervised end-to-end approaches in this domain. Another study by Coto-Solano et al. (2022) was conducted on Cook Islands Maori (CIM), a low-resourced indigenous language, to compare the performance of three ASR models: A traditional modular system (Kaldi; Povey et al., 2011) and two deep learning-based systems (DeepSpeech (Hannun et al., 2014) and XLSR-wav2vec 2.0 (Conneau et al., 2021)) and their results also indicated that Deep Learning ASR systems XLSR-wav2vec 2.0 are performing at the level of modular ASR methods on small datasets, and they are also effective in dealing with extremely low-resourced Indigenous languages like CIM. A study on Swedish L2 learners by Al-Ghezi et al. (2021) found that models pre-trained on large size of untranscribed L1 Swedish speech data give a competitive performance to that of modular ASR without the need for customized language and pronunciation models. Their best model managed to correctly decode words that do not appear in the training dataset whereas the modular ASR failed to do so. In Enzell (2022), domain adaptation with an N-gram LM is shown for Swedish, where the effects of LM weights on end-to-end models are briefly discussed.

## 3 Data

In our research, we utilized various open-source datasets and public speech corpora. For the Spanish ASR, we utilized the Common Voice (CV) Spanish (Ardila et al., 2020) dataset for the AM. The CV dataset includes rich metadata such as speaker age, accent, and gender, and consists of 213244 utterances for training, equating to 313.56 hours of speech material. For building the LM, we utilized the Spanish Billion Words Corpus (Cardellino, 2019) which has nearly 1.5 billion Spanish tokens and 0.54 million types with a frequency higher than 10. For testing, we used the CV Spanish Dev 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) (Wells et al., 1997). It's worth noting that obtaining datasets for Spanish was relatively easy as it is a well-resourced language with a substantial digital footprint. See the Table 1.

Table 1: Overview of Common voice Spanish Datasets

| Dataset | #Utterances | Duration | #Word Token | #Word Type |
|---|---|---|---|---|
| Train | 213244 | 313.56h | 2124011 | 83604 |
| Test | 15440 | 26.1h | 151681 | 23314 |
| Dev | 15440 | 25.9h | 151819 | 23602 |

For Irish, the situation is essentially different. Acquiring speech data for this language is a significant challenge due to the scarcity of open-source resources available for this language. To tackle this scarcity problem, we combined multiple small open-source Irish datasets. For the AM training we utilized the CV Irish dataset (Ardila et al., 2020). 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" dataset (Braude et al., 2019) which contributed an additional hour of Irish speech data. We also incorporated all Irish utterances from the "Google Fleurs" dataset (Conneau et al., 2023). After combining these 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 CV Irish Test set, containing 513 utterances (0.5 hours of speech), in combination with a set of 'Invalidated' CV 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 CV dataset are the clips that have received more downvotes than upvotes. In Table 2, the overview of Irish datasets is provided.

For the LM, we used the CC-100: Monolingual datasets from Web Crawl Data (Conneau et al., 2020), which includes data for over 100 languages including Irish, with in total 84 million word tokens and 0.12 million word types having frequency higher than 10. Lastly, for permitting the experiments with Kaldi ASR, we trained a G2P model based on Joint-sequence models (Bisani and Ney, 2008) using 13300 seed Irish pronunciations acquired from Wikipron (Lee et al., 2020).

Table 2: Overview of Irish Datasets. **CV**, **LA** and **GF** abbreviated for Common Voice, Living Audio and Google Fleurs respectively.

| Dataset | #Utterances | Duration | #Word Tokens | #Word Types |
|---|---|---|---|---|
| **CV** Train | 4097 | 4.1h | 27880 | 2341 |
| **LA** Irish | 1122 | 1h | 11360 | 3542 |
| **GF** Irish | 1947 | 8.4h | 48929 | 9866 |
| **CV** Test | 513 | 0.5h | 3423 | 1109 |
| **CV** Invalidated | 282 | 0.3h | 2230 | 707 |

# 4 Experiments

Our experiment setup is composed of four objectives: 1. Evaluate the performance of both the modular and end-to-end ASR approaches in terms of WER and Character Error Rate (CER), 2. Examine the influence of LM weights when integrating with fine-tuned wav2vec 2.0 models 3. Evaluate the presence of non-lexical words in the generated transcriptions and 4. Measure the latency of the ASR systems when deployed using both methods.

## 4.1 Modular ASR Training

We established the first baseline modular ASR for Spanish and Irish languages, using the dataset specified in Section 3. For the Spanish ASR, the baseline was built using a Kaldi (Povey et al., 2011) chain model adapted from the Librispeech recipe[1], while for the Irish ASR, it was adapted from the mini-Librispeech[2] recipe. 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 data size. The AM in both recipes is a combination of a Time-Delayed Neural Network (TDNN) and a Convolutional Neural Network (CNN). Additionally, 4-gram statistical LMs for Spanish and Irish were generated using the SRILM tool (Stolcke, 2002), based on the text resources for these languages. Finally, the pronunciation lexicons were created using a rule-based approach for Spanish and a data-driven approach for Irish as explain in the section 3.

## 4.2 Finetuning with End-to-End Approach

We utilized a publicly released pre-trained wav2vec 2.0 model (Baevski et al., 2020), XLS-R (Babu et al., 2022), which was trained on 436K hours of publicly available speech audio and is available on HuggingFace[3]. 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 finetuned the model on our labeled speech data for both Spanish and Irish. We used the 300 million-parameter version of XLS-R[4], which is among the smaller versions (mid 2023, models range from 300 million to two 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 total number of global steps for fine-tuning to Spanish and Irish was 44415 and 7180, respectively. In our research, the same language-dependent statistical LM was used for the modular and on end-to-end approach, for both Spanish and Irish. These LMs were initially created in ARPA format but were transformed into binary using KENLM (Heafield et al., 2013) to decrease the time required to load the models. The integration of the LM with the AM was performed using shallow fusion through the CTC decoder library `pyctcdecode`[5].

## 4.3 Non-lexical Words

An ASR system based on CTC may produce non-lexical word forms. In wav2vec 2.0, the output of the model is represented as the probability distribution of the predicted phonemes at each time frame (each 20ms) of the input signal. While the model can generate both lexical and non-lexical word forms through its sequence of phonemes, the use of a (word-based) LM helps to refine non-lexical predictions by incorporating information about the likelihood of different sequences of phonemes that form words (legal grapheme sequences or legal phone sequences) in the language.

[1] github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/

[2] github.com/kaldi-asr/kaldi/blob/master/egs/mini_librispeech/

[3] huggingface.co/docs/transformers/model_doc/wav2vec2

[4] huggingface.co/facebook/wav2vec2-xls-r-300m

[5] github.com/kensho-technologies/pyctcdecode/

In contrast, Kaldi's lexicon search space is limited to the pronunciation lexicons. The HCLG graph in Kaldi uses the lexicon FST (Povey et al., 2011; Mohri et al., 2007) to determine the possible words based on the AM's predictions, effectively restricting the search space to the words defined in the lexicon. This ensures that Kaldi only produces words that we provide, rather than generating non-existing words, leading to more accurate results.

In wav2vec 2.0, we investigated the effect of varying the weight of the LM during the shallow fusion process, by calculating the number of unique words (word types) in each experiment for different values of LM weights ranging from 0 to 1, with intermediate values of 0.1, 0.3, 0.5, 0.7, 0.8, 0.9, and 1.0. The results of these experiments allowed us to observe the effects of non-lexical words in hypothesis transcripts generated by wav2vec 2.0.

### 4.4 ASR Usability in Deployment

The ASR created using both approaches was deployed as a web service. The Kaldi-based ASR pipeline is capable of processing most speech files faster than real-time using only CPUs (Parikh et al., 2022).

However, decoding with large wav2vec 2.0 models with an integrated LM is prohibitively slow on a CPU and therefore requires the availability of at least one GPU for real-time decoding. Additionally, the wav2vec 2.0 models needed to be manually loaded for the first time setup. The latency of the ASR web service is an important feature for the usability of the entire system and user satisfaction. We calculated the latency results in terms of RTF for audio files of varying durations for both Kaldi ASR and wav2vec 2.0 models while maintaining a consistent connection environment. Linear regression was used to obtain equations. The linear trendlines were obtained by fitting linear models to each dataset using the Ordinary Least Squares (OLS) method. The slope and intercept coefficients of each line were calculated using the linear regression model.

## 5 Results

The initial evaluation of both systems is based on WER. For the modular approach, we conducted online decoding with Kaldi ASR, and for the end-to-end wav2vec 2.0 approach, we computed the WERs for the finetuned model and for the shallow-fused model with various weights of LM.

Table 3: Experimental Results of Kaldi ASR using a CNN-TDNN Architecture for AM: Testing Datasets and Corresponding WER and CER

| **Spanish ASR** | | |
|---|---|---|
| **Dataset** | **WER** | **CER** |
| CV Test | 15.69% | 5.89% |
| CV Dev | 13.68% | 4.90% |
| **Irish ASR** | | |
| CV Test | 22.69% | 11.54% |
| CV Invalidated | 43.06% | 24.44% |

As shown in Table 3 and 4, the end-to-end wav2vec 2.0 method outperformed the modular Kaldi ASR approach. In Spanish ASR, with Kaldi, we obtained WERs of 15.69% and 13.68% on the CV Test and CV Dev sets, respectively, which were improved to 10.63% and 9.38% by wav2vec 2.0 without an LM. Similarly, in Irish ASR, we obtained WERs of 19.98% and 39.19% using the wav2vec 2.0 without an LM on the CV Test and Invalidated sets, compared to the Kaldi ASR with WERs of 22.69% and 43.06%.

We also determined the impact of an LM on the finetuned model with wav2vec 2.0. As described in section 4.3, we computed the WER and CER for a number of LM weight values. For the Spanish ASR, the lowest WER of 6.73% and 5.92% on the CV Test and CV Dev sets, respectively, was achieved with an LM weight of 0.7. In the Irish ASR, the lowest WER was obtained at an LM weight of 0.8, with WERs of 13.78% and 30.85% on the CV Test and CV Invalidated sets, respectively. These results demonstrate a significant improvement in WER compared to the baseline modular ASR, using the same training data.

We evaluated the impact of the LM weight on the non-lexical words in the hypothesis transcripts generated by Spanish and Irish wav2vec 2.0 models. The non-lexical words were defined as words that were not present in the unigrams of the LM shallow-fused with the wav2vec 2.0 model. As seen in Table 5 initially, without using an LM, there were 6220 and 5770 non-lexical words in the Spanish CV Test and Dev hypothesis transcripts, respectively. By integrating an LM and increasing the weight of LM to 0.5, the non-lexical words were reduced to a minimum of 1317 and 1235 in CV Test and Dev transcripts, respectively corresponding to a reduction of approximately 79% of the total non-lexical words. Similarly in Irish ASR, without using an

Table 4: WER and CER of two test sets for Spanish and Irish ASR by wav2vec 2.0. We recorded WER and CER for fine-tuned model integrated with different LM weights.

| Dataset | Evaluation Matrix | No LM | LM Weights | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0 | 0.1 | 0.3 | 0.5 | 0.7 | 0.8 | 0.9 | 1 |
| **Spanish ASR** | | | | | | | | | | |
| CV Test | WER | 10.63% | 10.44% | 9.03% | 7.43% | 6.85% | **6.73%** | 6.82% | 6.98% | 7.20% |
| | CER | 3.09% | 2.95% | 2.73% | 2.44% | 2.34% | **2.35%** | 2.39% | 2.44% | 2.49% |
| CV Dev | WER | 9.38% | 9.06% | 7.86% | 6.53% | 6.03% | **5.92%** | 6.01% | 6.15% | 6.39% |
| | CER | 2.59% | 2.47% | 2.28% | 2.03% | 1.94% | **1.93%** | 1.97% | 2.01% | 2.06% |
| **Irish ASR** | | | | | | | | | | |
| CV Test | WER | 19.98% | 19.07% | 17.23% | 14.95% | 13.96% | 13.87% | **13.78%** | 13.78% | 13.87% |
| | CER | 7.24% | 6.91% | 6.52% | 6.03% | 5.79% | 5.84% | **5.85%** | 5.88% | 5.89% |
| CV Invalidated | WER | 39.19% | 39.95% | 37.62% | 33.45% | 31.88% | 31.07% | **30.85%** | 31.07% | 31.39% |
| | CER | 16.81% | 16.54% | 16.11% | 15.39% | 15.16% | 15.20% | **15.15%** | 15.23% | 15.28% |

Table 5: Count of non-lexical words in transcripts generated by wav2vec 2.0

| LM Weight | Test Dataset | | | |
|---|---|---|---|---|
| | Spanish | | Irish | |
| | CV Test | CV Dev | CV Test | CV Invalidated |
| **No LM** | 6220 | 5770 | 339 | 357 |
| **0** | 3408 | 3046 | 257 | 238 |
| **0.1** | 2440 | 2184 | 207 | 197 |
| **0.3** | 1538 | 1404 | 149 | 150 |
| **0.5** | 1317 | 1235 | 124 | 125 |
| **0.7** | 1404 | 1324 | 119 | 122 |
| **0.8** | 1555 | 1438 | 122 | 128 |
| **0.9** | 1769 | 1622 | 124 | 132 |
| **1** | 1999 | 1820 | 127 | 141 |

LM, in CV Test and Invalidated, there were 339 and 357 non-lexical words which were reduced to 119 and 122 using an LM with 0.7 weight corresponding to a reduction of approximately 65% of the total non-lexical words. Although the optimal WER and CER were achieved with only marginal differences at LM weights of 0.7 for Spanish and 0.8 for Irish, it can be said that there is still a presence of a small number of non-lexical homophones in hypothesis transcripts. However, even with a high LM weight, not all non-lexical words were removed. A slight increase in the number of non-lexical words was observed as the weight of the LM was increased from 0.7. This highlights the fact that even with low CER produced by wav2vec 2.0 models, there can still be a significant number of non-lexical words present in the generated transcripts.

In Figure 1, we present a comparative analysis of latency times and Real-Time Factors (RTF) for two ASR systems, Kaldi and wav2vec 2.0. This analysis covers audio files ranging from 5 to 102 seconds in duration, all processed under identical testing conditions, including network settings and beam size. Additionally, we consider a scenario where the wav2vec 2.0 model is utilized with a NVIDIA Tesla T4 GPU with 15.36GB of memory. The key observation is that both Kaldi and wav2vec 2.0 exhibit linearly increasing latency times as audio duration extends. For Kaldi, the latency equation is given by $y = 0.1074x + 0.50190$ ($y$: latency; $x$: duration in seconds), while for wav2vec 2.0 on the identical testing condition as Kaldi ASR, it is $y = 0.2380x - 0.8749$. When using wav2vec 2.0 with GPU backend, the latency equation becomes $y = 0.0426x + 0.8357$. These equations describes how the latency of ASR system increases as the duration of the audio input increases. In summary, the latency is same in all the ASRs for audio utterances up to 10 seconds. It is also evident that all three systems experience increased latency with longer audio segments. Wav2vec 2.0 displays a higher linear increase, while Kaldi exhibits a slower rate of increase, indicating greater efficiency with longer audio. Notably, wav2vec 2.0 with GPU acceleration demonstrates significantly reduced latency, underscoring the advantages of GPU processing for longer audio tasks. These insights are invaluable for selecting the ideal system for real-time or near-real-time audio processing, considering expected processing times based on varying audio durations.

The RTF values for both systems show an inverse relationship with audio file length. For Kaldi, the estimated RTF is $y = -0.0008x + 0.1791$ ($y$: RTF; $x$: duration in seconds). Kaldi's performance is only 0.0161 times better (RTF = 0.129 for 20 seconds of audio to RTF = 0.113 for 102 seconds of audio) for files from 20 to 102 seconds long. In contrast, wav2vec 2.0 model when same system as Kaldi ASR in backend gives an RTF estimated

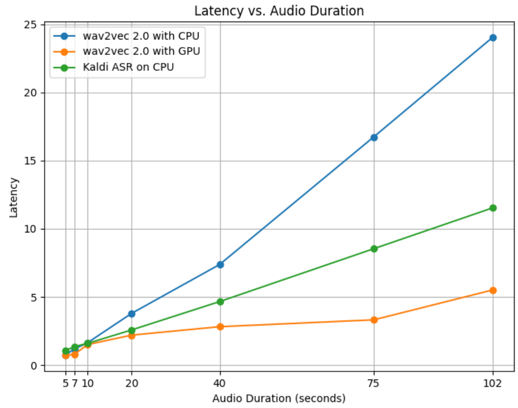

Figure 1: Latency measured in terms of system time for wav2vec 2.0 models and Kaldi vs. Audio Duration

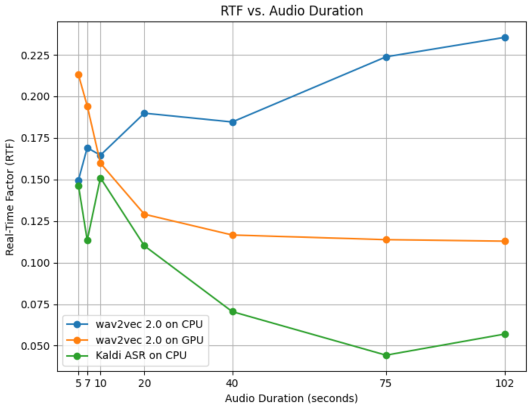

Figure 2: Real-time Factors (system time/length of audio) for wav2vec 2.0 models and Kaldi vs. Audio Duration

by $y = 0.00079x + 0.1588$. In this case, the RTF value is *increasing* with the audio duration, i.e. the system's processing time becomes relatively slower as the audio duration becomes longer. This suggests that the system might not be able to keep up with the real-time demands of longer audio segments, and it could experience prohibitive delays in processing or decoding longer audio. While this issue can be solved with an acceleration at backend as GPU and with GPU, wav2vec 2.0's performance is 3.5 times better for files from 20 to 102 seconds long, with an RTF of $y = -0.0009x + 0.1351$. The initial loading time of the wav2vec 2.0 model (which takes around 10 to 20 seconds) is not taken into consideration in the charts.

## 6 Discussion

From our experimental results, it is evident that the end-to-end wav2vec 2.0 approach outperforms the modular Kaldi ASR for both well-resourced and low-resourced languages. In particular, we found that in end-to-end wav2vec 2.0 during shallow fusion increasing the LM weight from 0.0 to 0.7 and 0.8 for Spanish and Irish, respectively, led to a decrease in non-lexical words, WER, and CER, resulting in optimum performance. Interestingly, beyond a certain threshold, further increasing an LM weight led to an *increase* in non-lexical words, WER, and a decrease in performance. The wav2vec 2.0 model outputs a sequence of token probabilities represented in an alphabet set and an arg-max followed by a tokenizer provides sufficiently good accuracy but when an LM is integrated on top of it, words with lower probability and poor acoustic support are more likely to be overruled by the LM. Hence a reduction in WER and non-lexical words is found but after a certain limit for the LM-weight, the LM starts replacing correctly identified words resulting in an increase in WER. The default weight of LM in `pyctcdecode` is 0.5, but finding the optimum weight for the combination of LM and AM is crucial for achieving the best performance. In the modular ASR, after the decoding process performing lattice rescoring with recurrent neural LMs (Xu et al., 2018) can also further improve the ASR performance.

The knowledge-based hybrid system and end-to-end systems that we have compared here differ in terms of WER. This does not at all imply that the classical approach can defaultly be replaced by an SSL end-to-end approach. In the experiments reported on above, we observed that both systems often (but not always) make *different* errors, which opens the possibility to consider them as first-level audio-to-text transformations after which both 'streams' could be merged on a second level, based on considering confidence measures associated to each word in the hypothesized outputs in each stream. This stream-based merging of multiple different hypotheses is topic for a follow-up investigation.

In terms of time performance, the fitted latency and RTF lines are reliable indicators of trends in the data and can be used for predictions and insights. For real-time decoding, wav2vec 2.0 takes considerably more time to decode the longer than 10 seconds audio, compare to Kaldi ASR in the same network connection and server infrastructure but with a GPU acceleration, wav2vec 2.0 decoding times outperform Kaldi in all cases, but a first model loading time must be taken into account in the case of wav2vec 2.0.

## 7 Conclusion

We compared the performance of modular and end-to-end approaches for creating ASR on a low and well-resourced language and results showed that the end-to-end wav2vec 2.0 ASR outperforms the modular Kaldi ASR even without an LM. Incorporating an LM with weights of 0.7 and 0.8 for Spanish and Irish languages, respectively, further improves the performance of the end-to-end approach. However, we observed that the end-to-end approach generates non-lexical words, which can be partially resolved but not entirely eliminated by integrating an LM. Also, a dedicated GPU is required to achieve the best time performance for end-to-end ASR, which is 3.5 times faster than modular ASR. Therefore, modular ASR can still be a relevant option for in-domain tasks with lower CPU/GPU requirements.

## Limitations

There are mainly three limitations with our study. 1. The main limitation of this study concerns the data preparation phase, especially for low-resource languages. Conducting experiments, as presented in this paper, requires adequate linguistic resources. It includes not only audio material but also essential components such as lexicons or a grapheme to phoneme conversion system. The scarcity of such linguistic resources for minority languages can pose a significant challenge so the availability of such an ASR system remains crucial for this comparison. 2. Another significant limitation relates to the availability of suitable large models, such as Whisper, for the purpose of comparison. Not all pre-trained end-to-end ASR systems encompass support for every minority or low-resourced language. So the availability of such an ASR system remains crucial for this comparison. 3. Third limitation is the hardware dependent performance. In our case, AMD 32-Core Processor with a total of 64 CPUs, which is also quite capable. However, the performance of ASR systems can be impacted by factors at server side such as CPU load, available memory, and system usage by other processes and at network side such as bandwidth, processing speed, and transmission protocol. This variability can affect the latency and RTF of the ASR system, meaning that the time it takes to process and transcribe speech can vary under different system conditions.

## Acknowledgements

This work has been conducted in the SignON project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017255.

## References

Ragheb Al-Ghezi, Yaroslav Getman, Aku Rouhe, Raili Hildén, and Mikko Kurimo. 2021. Self-Supervised End-to-End ASR for Low Resource L2 Swedish. In *Proc. Interspeech 2021*, pages 1429–1433.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. In *Proc. Interspeech 2022*, pages 2278–2282.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

David A. Braude, Matthew P. Aylett, Caoimhín Laoide-Kemp, Simone Ashby, Kristen M. Scott, Brian Ó Raghallaigh, Anna Braudo, Alex Brouwer, and Adriana Stan. 2019. All Together Now: The Living Audio Dataset. In *Proc. Interspeech 2019*, pages 1521–1525.

Cristian Cardellino. 2019. Spanish Billion Words Corpus and Embeddings.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised Cross-Lingual Representation Learning for Speech Recognition. In *Proc. Interspeech 2021*, pages 2426–2430.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

Rolando Coto-Solano, Sally Akevai Nicholas, Samiha Datta, Victoria Quint, Piripi Wills, Emma Ngakuravaru Powell, Liam Koka'ua, Syed Tanveer, and Isaac Feldman. 2022. Development of automatic speech recognition for the documentation of Cook Islands Māori. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3872–3882, Marseille, France. European Language Resources Association.

Viktor Enzell. 2022. Domain adaptation with n-gram language models for swedish automatic speech recognition : Using text data augmentation to create domain-specific n-gram models for a swedish open-source wav2vec 2.0 model. Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS).

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep speech: Scaling up end-to-end speech recognition.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria. Association for Computational Linguistics.

Jackson L. Lee, Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. Massively multilingual pronunciation modeling with WikiPron. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4223–4228, Marseille, France. European Language Resources Association.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 2007. Speech Recognition with Weighted Finite-State Transducers. In *Handbook on Speech Processing and Speech Communication, Part E: Speech recognition*.

Aditya Parikh, Louis ten Bosch, Henk van den Heuvel, and Cristian Tejedor-García. 2022. Design principles of an automatic speech recognition functionality in a user-centric signed and spoken language translation system. *Computational Linguistics in the Netherlands Journal*, 12:19–32.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.

Mukund K Roy, Sunita Arora, Karunesh Arora, and Shyam S Agarwal. EasyChair, 2022. Building speech corpus in rapid manner to adapt a general purpose asr system to specific domain. EasyChair Preprint no. 7181.

Brij Mohan Lal Srivastava, Sunayana Sitaram, Rupesh Kumar Mehta, Krishna Doss Mohan, Pallavi Matani, Sandeepkumar Satpal, Kalika Bali, Radhakrishnan Srikanth, and Niranjan Nayak. 2018. Interspeech 2018 Low Resource Automatic Speech Recognition Challenge for Indian Languages. In *Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018)*, pages 11–14.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proc. 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 901–904.

John C Wells et al. 1997. Sampa computer readable phonetic alphabet. *Handbook of standards and resources for spoken language systems*, 4:684–732.

Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. 2018. A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5929–5933.

Cheng Yi, Jianzong Wang, Ning Cheng, Shiyu Zhou, and Bo Xu. 2021. Transfer ability of monolingual wav2vec2.0 for low-resource speech recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6.

Juan Zuluaga-Gomez, Amrutha Prasad, Iuliia Nigmatulina, Seyyed Saeed Sarfjoo, Petr Motlicek, Matthias Kleinert, Hartmut Helmke, Oliver Ohneiser, and Qingran Zhan. 2023. How does pre-trained wav2vec 2.0 perform on domain-shifted asr? an extensive benchmark on air traffic control communications. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 205–212.