Leveraging AV-HuBERT clusters for ASR and pre-trained model evaluation

Marek Sarvaš
101183486

June 1, 2023

Abstract

Automatic Speech Recognition (ASR) is a task of transcribing human speech into text. Nowadays performed mainly using large deep neural network models. To achieve desired performance, these models are trained on large quantities of data and in today's very popular self-supervised fashion. This way, models learn to predict discrete cluster IDs during training and to learn complex features during the process. To increase the performance and robustness of these models, previous works [13, 4] introduced another modality into the ASR task, video. These works showed that models pre-trained in multi-modal settings have state-of-the-art performance and can beat single-modality models in the ASR downstream task. Usually, these pre-trained models are used as initialized encoders and are fine-tuned with added decoder model for specific downstream task. Fine-tuning such large models can be a costly task when trying to select the best fitting pre-trained model for our specific downstream task. This work investigates clusters extracted from pre-trained AV-HuBERT and, if or how can they be used to determine best pre-trained models for a downstream task. The experiments showed that small neural networks (LSTM in this case) could be trained only on cluster IDs extracted from big pre-trained models, and during the inference, network's mappings between characters and clusters can be obtained.

1 Introduction

Automatic speech recognition system (ASR) is a system that is capable of transcribing human speech into text. Speech is usually a preferable way of communication between humans, and ASR systems can be a means to achieve this level of communication also between humans and machines.

ASR has been the focus of research for many years now and is used in a wide range of applications, such as speech assistants, dictation, speech translation, or just transcribing the audio input in various scenarios that help humans. This is still a challenging task, especially if we want to use ASR systems in a real-world environment settings where the input audio is usually very noise with different sounds, not just a speech. This has a large impact on the performance of these models, as it is highly dependent on the data used for training and training techniques.

The use of deep neural networks, such as Recurrent Neural Networks and nowadays very popular transformers, has greatly improved the performance of ASR in the last few years. The aim is to train robust models that perform well in different available evaluation datasets. However, the data and their quality differ across the datasets[9]. In most cases in machine learning, the more data available for training, the better a model tends to be. Because of this, nowadays unsupervised learning is more and more popular, as it can leverage unlabeled data and therefore provide more data options for training.

The effective use of unsupervised learning for ASR lies in pre-training large models for the prediction of discrete units. Such pre-trained models and learned complex feature representations can then be fine-tuned and used for different downstream tasks, not only ASR. As some previous works show[6, 13, 2] shows, very good results in ASR can be achieved by leveraging such models pre-trained in self-supervised or unsupervised fashion, not only for the ASR task. Some previous work[13, 4] showed that training models in multi-modal settings for ASR can boost performance even more. In these cases, it is by adding visual features to the input. These features are extracted from video frames of lips while speaking (lip reading task) and combined with audio features for models to learn to predict multi-modal units.

These discrete units or feature cluster IDs are used in pre-training; however, after that, they are usually not used anymore in any downstream task. In these scenarios, higher-dimensional embeddings are used rather than cluster IDs. In this work, we try to leverage these learned clusters to train small model for ”translation” from these IDs, extracted from the last layer of pre-trained encoder, to characters. The idea behind training such models is to see if there is any correlation between quality of these clusters or ability of some models to learn this mapping and performance of the fine-tuned versions of big pre-trained models. This knowledge could be used in scenarios where we want to use pre-trained models, such as AV-HuBERT or HuBERT, for a fine-tuning on a downstream task, but fine-tuning all the available pre-trained models is costly in terms of computational hardware and time. Instead we could train a much smaller model on features from these models and pick the best fitting pre-trained model for wanted downstream task.

2 Background

Automatic Speech Recognition (ASR) is a task of transcribing spoken language into text. Multiple different approaches have been used over the years to develop better ASR systems. Nowadays, a very popular approach is to use deep neural networks based on transformer architecture. The input of such models is often transformed into some form of intermediate features extracted from the audio waveform[11].

The audio data are usually not ideal in terms of how good the quality of the audio is, how much noise there is, either created by the hardware used to record the audio or other factors. Due to this, some form of data pre-processing is used before feature extraction, that is, minimizing SNR (signal-to-noise ratio), removing silence at the begging and end of the audio, or trimming the audio to some maximum length[11, 13]. Depending on the model, different methods of feature extraction are used. Popular and frequently used methods are extracting MFCC features that are used as input to the model or some small neural network model used as a modality specific (audio in this case) encoder projecting waveform into intermediate features[6, 11, 13].

In addition to ASR, Audio-Visual ASR models take as input not only audio features, but also video data, in the form of a lip reading task using video frames with lips corresponding to the audio part. Same as for audio, there is usually a data pre-processing part and extracting the features from visual data that are then used in combination with audio data. The intention behind such models is to enhance the performance and robustness of ASR models, for example, against noisy data [13, 4]. One of the use cases for adding the lip reading task as additional visual information to traditional ASR is presented by Chan et al.[4].

Today, a very popular approach is to use unsupervised learning with discrete unit discovery as a pre-training method for the ASR models. However, the discrete clusters are often used just for pre-training the models, which are used as encoder or initialization for fine-tuning for some downstream task the encoders, and these clusters are not used further.

2.1 Unsupervised learning

Unsupervised learning has been increasingly used in automatic speech recognition (ASR) in recent years, offering numerous benefits over traditional supervised learning methods. One of the benefits is the ability to learn from unannotated speech data or to discover hidden patterns in speech. This provides a great advantage because obtaining labeled data is often a challenging task. Typically, speech recognition systems require a large volume, thousands of hours, of transcribed speech data to achieve the desired performance. Several works have shown that the use of unsupervised learning can outperform supervised learning[6, 2] and also in combination with supervised learning, as shown by Bai et al.[3]. Unsupervised learning was also successfully used in other domains, such as natural language processing [5].

Clustering methods[13, 6] and variations of methods for predicting some kind of labels[2, 6] are widely used for models trained in an unsupervised fashion. Clustering methods such as k-means were used to discover discrete latent acoustic units[6] or, in multimodal settings, to create discrete cluster labels used as an initial target for unsupervised learning[13]. This is often used with masked prediction of discrete cluster labels, such as masked language modeling[6, 2], where some of the cluster labels in the input are masked and the model is trained to predict these labels. Several works showed[2, 6, 4], that pre-training models in this way as an encoder and then fine-tuning them for some downstream task can greatly improve performance.

2.2 AV-HuBERT

Paper by Shi et al.[13], describes several methods of speech recognition on video domain. Two preliminary methods are proposed, beside the main AV-HuBERT. The first is Visual HuBERT, which is HuBERT[6] adapted to the video domain instead of the audio. This model learns to predict targets using only visual features. Second, Cross-model Visual HuBERT which leverages audio, by distilling knowledge from audio data, to model visual inputs.

Audio-Visual HuBERT[13] is an transformer encoder model trained to predict multimodal clusters in a fashion similar to visual HuBERT[13], which is HuBERT adapted for the visual domain, with improvements. First, this model consists of two separate encoders, ResNet-18 and linear projection, for the visual and audio domains, respectively. The intermediate features extracted from these encoders are then fused together with channel-wise concatenation and used as input to the common transformer encoder to predict the cluster assignments for each frame. Modality dropout is used to prevent audio features from dominating. Similarly to other previous works[2, 6], a masking strategy was also used to train AV-HuBERT. However, Shi et al.[13] proposed a new strategy in which masked segments of visual input are replaced by random segments of video.

Pre-training is done in 5 iterations and on both modalities. This offers the advantage of producing multimodal cluster assignments. These cluster assignments are then utilized as target labels for the masked prediction task for the next iteration. Using this method, after the initial iteration, AV-HuBERT targets become inherently multimodal, and as shown by Shi et al.[13], this combination of modality provides better quality.

2.3 LRS3 dataset

LRS3-TED[1] is a multimodal dataset created for speech recognition. This dataset consists of video, audio and corresponding transcription of over 400 hours extracted from 5594 TED and TEDx talks. The video part of this dataset is provided as .mp4 files consisting of cropped faces with resolution 224 × 224 and a frame rate of 25 fps. The audio part is provided as single channel 16kHz tracks. Text transcriptions and alignment boundaries are provided in plain text.

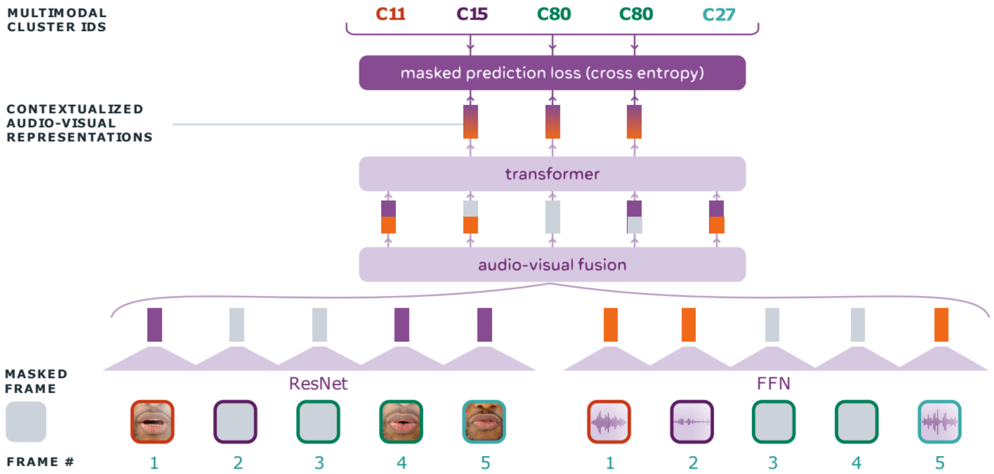

Figure 1: AV-HuBERT model with multi-modality input. (Taken from Shi et al.[13])

This dataset is divided into 3 subsets: `pre-train, train-val, and test` set, providing 407, 30, and 1 hour of data, respectively. The number of utterances corresponding to this is 119k, 32k, and 1452, respectively.

AV-HuBERT models used in the experiments described below were pre-trained on `pre-train` subset, and all the experiments were done only on `train-val` and `test` sets.

3 Data and Models

As described in the next chapter, two main experiments were carried out. To replicate the fine-tuning experiments from the AV-HuBERT paper[13] and to be able to compare the results between our experiments, the only data used for the following experiments in this report was 30 hours of the LRS3 dataset (Section 2.3) and clusters extracted from the AV-HuBERT model (again using only LRS3 data as input).

3.1 Data preparation

Data pre-processing of the LRS3 dataset with additional information is described by Shi et al.[13]. The audio part of the data preprocessing was performed by extracting audio from the .mp4 video data. The audio data are processed by extracting the MFCC features with 26 dimensions at intervals of 10 ms from the original waveform, which is then utilized as the input for the model.

The video part of the data is processed by detecting the 68 facial keypoints in each video clip using dlib[7] and aligning it with a reference frame. For each frame, the lip-centered region with size 96×96 pixels is cropped. Then it is randomly cropped to size 88×88, randomly flipped upside down and converted to gray scale before using it as input to the model during training.

Data for experiments described in Section 4.2 were obtained by extracting the cluster IDs from the last layer of pre-trained AV-HuBERT models (available online)[13]. The AV-HuBERT model was fed preprocessed audio + video data (as described in Section 2.2), followed by the extraction of the cluster IDs from the last layer of the model. In every utterance, the sequence of the same cluster ID was trimmed to only one ID. Even after this trimming, the cluster ID sequences are much longer than the number of characters in a corresponding utterance. The number of distinct cluster IDs differs from the model pre-training iteration (1000 for iteration 4 and 2000 for iteration 5). The example of the final dataset is shown in Table 3.1.

Source	659 26 1604 123 991 760 1080 231 421 342 956 1335 1771 ...
Target	n o | w o n d e r | n e w s p a p e r s | a r e | d y i n g
Original text	no wonder newspapers are dying

Table 1: Example of one training utterance from final train-val set used to train the model to map extracted cluster IDs to text characters. Source utterances consist of extracted cluster IDs from pre-trained AV-HuBERT. Target utterances consist of space separated characters and ”|” separated words.

3.2 Models

For fine-tuning and cluster extraction, the base (iteration 4 and 5) and large (iteration 5) AV-HuBERT model was used (described in 2.2). This experiment was carried out using the code provided by the AV-HuBERT paper [13] that was implemented using fairseq [12].

Small LSTM encoder-decoder models with attention were used for the cluster ID to character translation task. For this task, the openNMT toolkit[8] was used, for the architecture of the LSTM models and the training process for these models. An in-depth description of the LSTM model using attention is in the paper by Luong et al. [10]. The model consists of an encoder-decoder, where both parts have 2 LSTM layers, and the decoder also uses an attention layer on top of that, this architecture of the model with whole experiment pipeline is shown in Figure 4.2. The experiments of mapping the cluster IDs to characters were performed using two models, lstm500 and lstm1024, which differ in size of the hidden layer 500 and 1024, respectively.

4 Experiments

The initial question was whether we could create a fast method to map cluster IDs from the AV-Hubert model that does not require a lot of computational resources. This chapter describes preliminary experiments and their settings. Two types of experiments were conducted. First, fine-tuning different pre-trained AV-Hubert models to compare with the results presented in the paper by Shi et al.[13] and to compare with the results of the second experiment. Second, training small recurrent neural networks that learn the mappings between cluster IDs extracted from the pre-trained (not fine-tuned) AV-HuBERT models and characters to produce good quality text.

4.1 Fine-tuning AV-HuBERT

First, pre-trained AV-HuBERT models were fine-tuned on the LRS3 30h train-val dataset (Section 2.3). This fine-tuning part was performed for various reasons. First, as a preliminary experiment to verify any correlation between fine-tuning performance and the cluster ID-to-character translation model described in Section 4.2. Second, to compare with the results presented in the AV-HuBERT paper[13] and verify the correct data preparation step.

Fine-tuning was performed on three pre-trained AV-HuBERT models for audio-visual speech recognition: base (iteration 5), base (iteration 4) and large (iteration 5). All experiments were carried out in multimodal settings, using both audio and video, or only video input modalities during fine-tuning. For this task, the sequence-to-sequence (S2S) transformer decoder was added after the AV-HuBERT encoder to decode unigram-based subword units with vocabulary of size 1000 tokens for all three models. Base models were fine-tuned for 30000 steps, large model for 18000.

The decoding of these models was done on the LRS3 test set which is 1 hour of data (1321 utterances) in 3 different settings, where only audio, video and both audio + video were used for the decoding. This represents the results for ASR, lip reading, and audio-visual ASR tasks. The results and comparison with the original results of the AV-HuBERT paper[13] results are presented in Table 4.1.

The main interest of this work is the AV-HuBERT and its fine-tuned performance compared to the translation task (Section 4.2). In addition we can compare the results for Visual HuBERT fine-tuning, however, fine-tuning AV-HuBERT for audio ASR was left out because in the original paper the Audio-HuBERT[6] is fine-tuned with cluster IDs produced by AV-HuBERT, which is not the goal of this work.

model	fine-tuning modality	decoding modality	WER (%)
base iter 4	AV	AV	**6.89** %
		V	62.58 %
		A	7.58 %
base iter 5	AV	AV	7.29 %
		V	**59.31** %
		A	8.25 %
large iter 5	AV	AV	11.83 %
		V	61.71 %
		A	11.57 %

Table 2: WER % of AV-HuBERT fine-tuned on both (audio, video) modalities (AV). The decoding modality represents input modality during decoding on LRS3 test set, audio only ASR task (A), video only lip reading (V), multimodal ASR audio+video (AV).

model	fine-tuning modality	decoding modality	WER (%)
base iter 4	V	AV	30.81 %
		V	65.18 %
base iter 5	V	AV	11.74 %
		V	53.64 %
large iter 5	V	AV	26.55 %
		V	62.72 %
orig base iter 5	V	AV	-
		V	51.8 %
orig large iter 5	V	AV	-
		V	44.8 %

Table 3: WER % of AV-HuBERT fine-tuned for lip-reading, compared to values presented in AV-HuBERT paper[13].

4.2 Translation on AV-Hubert clusters

The main objective of this work is to try to answer if the AV-HuBERT cluster IDs could be mapped to characters using a small or no neural network model. This part serves as preliminary experiments for this question and, as can be seen in Table 4.2, it can be done by training a small recurrent neural network. The dataset and the models used in this part are described in the second half of Section 3.1 and Section 3.2, respectively.

The LSTM models were trained on 31000 utterances (corresponding to transcribed 30 hours of LRS3 data) consisting of cluster IDs as source and English text as target. whole pipeline for these experiments can be seen in Figure 4.2. Training for all experiments was performed using cross-entropy with label smoothing as a loss function [14] and Adam as optimizer. The number of training steps was set at 50000 with 4000 warm-up steps, early stopping of 10 validation steps, and accuracy as early stopping criteria. Training in these experiments usually stops due to early stopping under 10000 steps.The results of the translation experiment are shown in Table 4.2. Two models, LSTM_1024 and LSTM_500, with hidden sizes of 1024 and 500, were used and trained on different cluster IDs.

Cluster IDs were obtained using two methods. First, clusters were extracted, with code provided on Github, from the very last output layer of the pre-trained AV-HuBERT models. In this case `base` and `large` refer to the AV-HuBERT model, and `iter` indicates from which pre-training iteration the checkpoints are. Iteration 5 contains 2000 unique clusters, whereas for iteration 4 this number is 1000. Second, the k-means algorithm was run on the intermediate features extracted from the layer corresponding to the number in Table 4.2. This way, the number of unique clusters was significantly reduced to only 100.

Fine-tuning vs ASR on clusters

In Table 4.2 below, we can see the comparison of the word error rate (WER) between fine-tuned AV-HuBERT for audio-visual ASR and small LSTM trained to map cluster IDs to characters. From the numbers obtained during the experiments, there is not

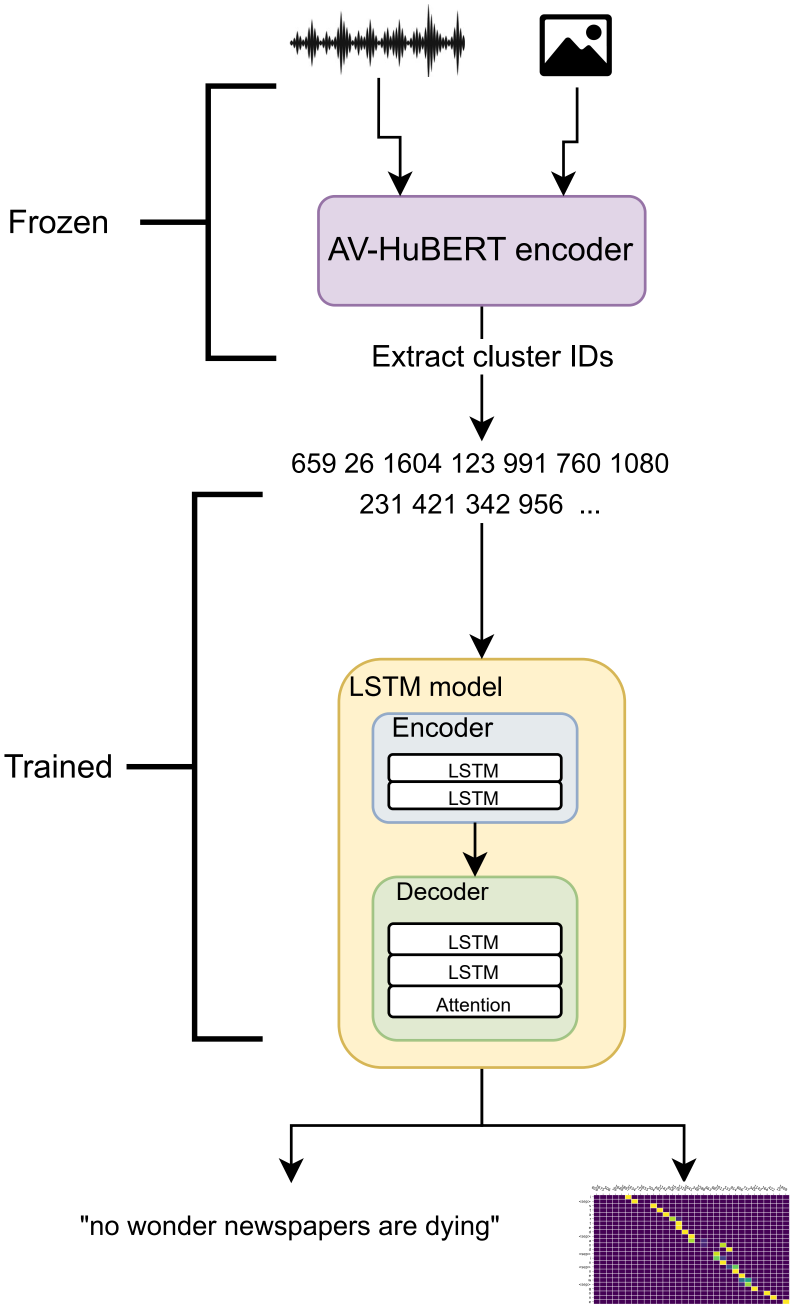

Figure 2: Architecture of LSTM model used for the clusterID-to-character translation.

any obvious correlation between the LSTM and fine-tuned AV-HuBERT performance. If we take a look only into LSTM_1024 results, there is tendency to get better results on clusters from bigger model and final iteration of pre-training. On the other hand, pre-trained models shows that last iteration is better, however BASE model performs better than LARGE model. This could be a result of different number of fine-tuning steps as described in Section 4.1. It is recommended to extend these experiments as described in

Cluster IDS model	Layer	LSTM model	WER (%)
base_iter4	12	LSTM_500	20.64 %
		LSTM_1024	29.77 %
base_iter5	12	LSTM_500	23.06 %
		LSTM_1024	18.95 %
large_iter5	12	LSTM_500	**18.22** %
		LSTM_1024	**17.79** %
K-means base_iter5	12	LSTM_500	44.15 %
		LSTM_1024	40.47 %
K-means large_iter5	12	LSTM_500	68.26 %
		LSTM_1024	65.80 %

Table 4: Performance of LSTM models on clusterID-to-character translation task. First two columns show from which pre-trained model and which layer were the cluster IDs extracted. In case of the last two rows, K-means clustering with 100 clusters was performed on top of the model's intermediate features.

Conclusion 5. Also repeating the translation experiments can provide more reproducible results as described translation models tends to be a bit volatile them to get more general idea about the translation between cluster IDs and English sentence as these small models tend to highly depend on training hyper-parameters.

Possible initialization for cluster assignments

As previous experiments show, we are able to train a small network to produce text from cluster IDs. The quality of this text is not on the same level in comparison to state-of-the-art ASR or machine translation models and techniques, which was expected. This part presents another set of experiments, that is, using these LSTM models to obtain attention maps between cluster IDs and characters during inference. The information obtained from the attention values could be used as initial assignments for a future mapping algorithm that does not require training a neural network. An example of the attention map can be seen in Figure 4.2, where the vertical axis corresponds to the text generated during the inference and the horizontal axis corresponds to the sequence of cluster IDs (inputs into the LSTM model during the inference).

The number of cluster IDs extracted from the AV-HuBERT models is quite high, assuming that these clusters contain more information about the data. This can be seen in Figure 4.2, where some clusters are not used during inference to produce the text output. In this example the empty middle part contains silence in audio file, however, different cluster IDs appear in this part, perhaps containing information about the video content. On the contrary, in some parts, the values indicate attention between single ID and multiple characters, as can be seen in Figure 4.2.

model	**fine-tune → dec**	**cluster IDs model**	**WER (%)**
LSTM_500	—	base_iter4	20.64 %
		base_iter5	23.06 %
		large_iter5	**18.22** %
LSTM_1024	—	base_iter4	29.77 %
		base_iter5	18.95 %
		large_iter5	**17.79** %
base iter 4	AV→AV	—	**6.89** %
	AV→V		62.58 %
	V→AV		30.81 %
	V→V		65.18 %
base iter 5	AV→AV	—	7.29 %
	AV→V		**59.31** %
	V→AV		**11.74** %
	V→V		**53.64** %
large iter 5	AV→AV	—	11.83 %
	AV→V		61.71 %
	V→AV		26.55 %
	V→V		62.72 %

Table 5: Summarization of experiments described in this section. First two rows provide WERs of LSTM models trained on cluster-to-character translation task. Last 3 rows provide results of fine-tuning AV-HuBERT models for audio-visual ASR and lip reading.

Other behaviors observed during the inference and training:

- Sometimes, especially if token normalization (instead of sentence normalization) is used during the training, the models produce repetitions. Repetitions are mainly in the form of repeating the end part of the sentence at the end. Figure 4.2
- On the other hand, when using sentence normalization, the repetitions are almost completely mitigated. However, high attention values between the first cluster ID and some characters appear. The same cluster ID appears at the beginning of every utterance, no matter from which AV-HuBERT model the clusters are extracted (a different number for each model). Figures 4.2 and 4.2.

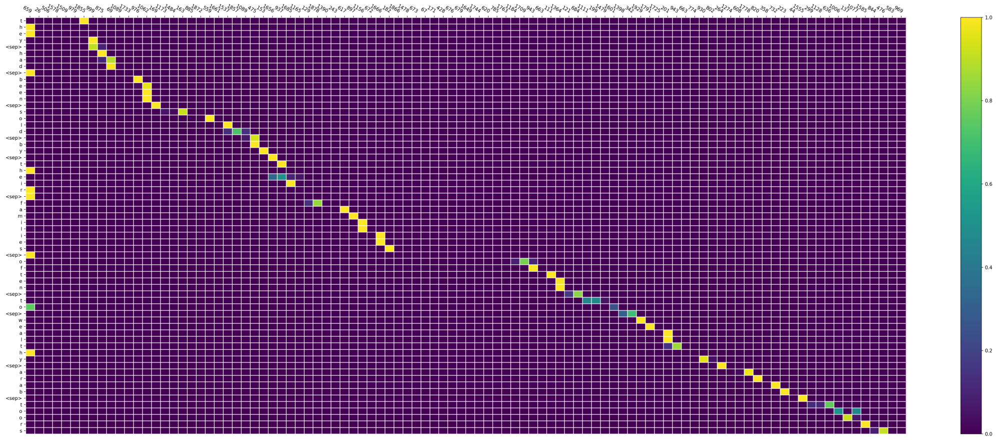

Figure 3: Example of attention values obtained during inference with short pause in speech in the middle.

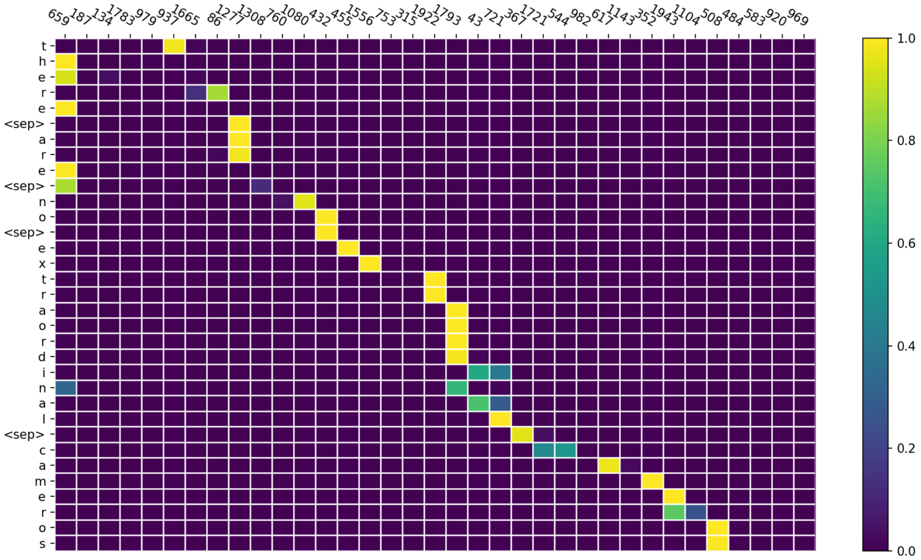

Figure 4: Sometimes different group of characters use same cluster ID.

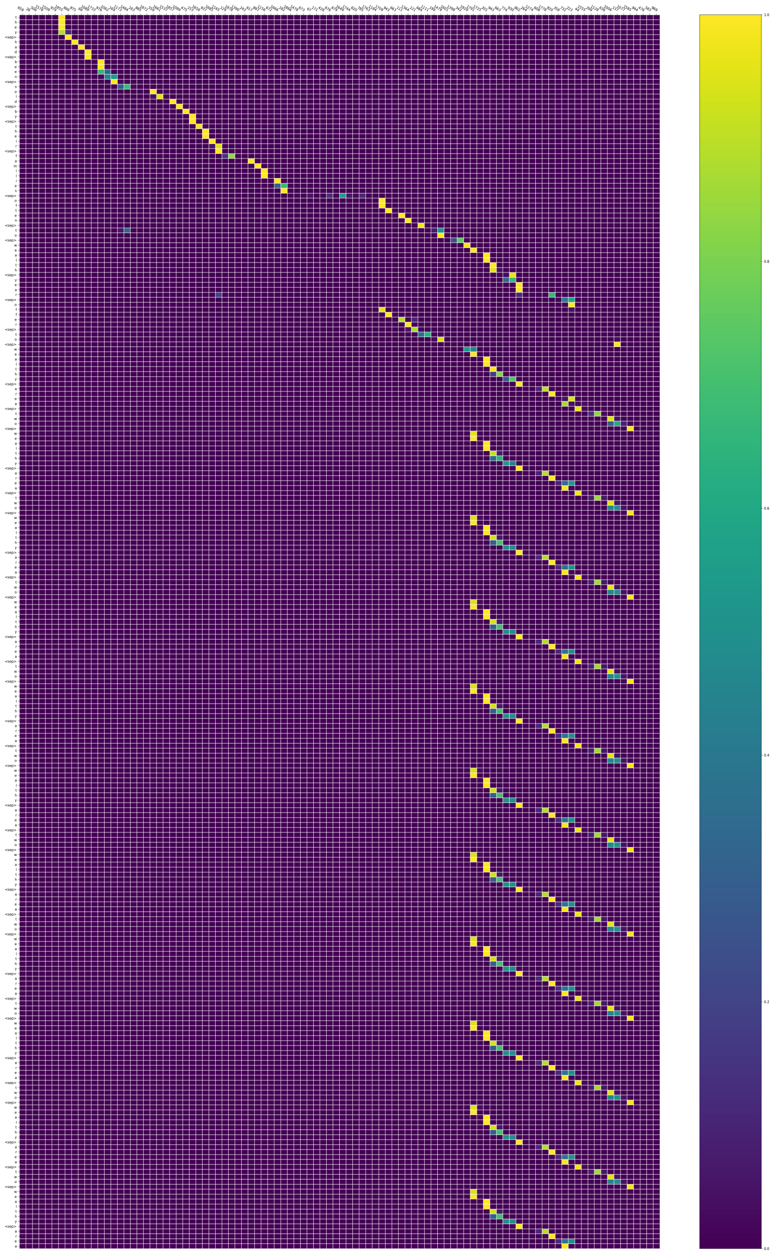

Figure 5: Repeating end of the sentence when using token normalization.

5 Conclusion

This work explores the performance of the multimodal ASR task, mainly targeting AV-HuBERT and the use of information extracted from this model. First, fine-tuning experiments were performed on pre-trained AV-HuBERT models for audio-visual and visual only task on small amount of data (30 hours). The results were comparable for the visual modality with those provided in the AV-HuBERT paper[13], and good performance was also achieved for the audio-visual modality, which was expected in this scenario. In the second part, we perform experiments to determine usability of information learned by these big pre-trained models in form of the cluster IDs learned during the training. The main question was if such clusters contain enough information about the data that they can be used either without any neural network, or small neural network (in comparison to today's big models) to perform ASR or evaluate pre-trained models for specific downstream task, ASR in this case. Small LSTM+attention neural network models were trained from scratch to perform a translation task between extracted cluster IDs and English sentences. WER of 18% was achieved even without any special neural network or loss function adapted for this task. This suggests that it is possible to use only these clusters for an ASR downstream task. However, there is no obvious correlation between LSTM performance and cluster training, as shown in Section 4.2. Additionally, during inference, attention values were extracted, providing some initial cluster-to-character mapping.

The results in this work provide only preliminary results for this topic and should be investigated in more detail in future works. Performed experiments can be extended in multiple directions. One of the directions can be to extend the first part of this work by training small neural networks on clusters or features extracted from either different pre-trained models, not only AV-HuBERT, or from models pre-trained on data different from LRS3 dataset. Another direction could be to further investigate the cluster-to-character mappings and what information is contained by the clusters with no or very small attention value during text generation.

References

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. LRS3-TED: a large-scale dataset for visual speech recognition. *CoRR*, abs/1809.00496, 2018.

[2] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020.

[3] Junwen Bai, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, and Tara N. Sainath. Joint unsupervised and supervised training for multilingual asr. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6402–6406, 2022.

[4] David Chan, Shalini Ghosh, Debmalya Chakrabarty, and Björn Hoffmeister. Multimodal pre-training for automated speech recognition. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 246–250, 2021.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, abs/2106.07447, 2021.

[7] Davis E. King. Dlib-ml: A machine learning toolkit. *J. Mach. Learn. Res.*, 10:1755–1758, dec 2009.

[8] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, jul 2017. Association for Computational Linguistics.

[9] Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Paden Tomasello, Jacob Kahn, Gilad Avidov, Ronan Collobert, and Gabriel Synnaeve. Rethinking evaluation in ASR: are our models robust enough? *CoRR*, abs/2010.11745, 2020.

[10] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[11] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80(6):9411–9457, Mar 2021.

[12] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[13] Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. Learning audio-visual speech representation by masked multimodal cluster prediction, 2022.

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.