Deep learning methods - project 2

Audio classification with transformers

Wojciech Klusek, Aleksander Kuś

5 may 2024

Task description

Project task:

- Classification of speech commands employing neural networks, with a primary emphasis on utilizing Transformer architectures.
- Multiple network architectures should be compared using the accuracy metric.
- Investigate the influence of changing hyper-parameters.

Dataset description

The characteristics of the **TensorFlow Speech Recognition Challenge** dataset:

- 1 second clips of voice commands.
- All audio files in the dataset belong to exactly one of the 31 classes.
- The labels that need to be predicted in are yes, no, up, down, left, right, on, off, stop, go. Everything else should be considered either unknown or silence.
- The folder _background_noise_ contains longer clips of "silence" that need to be broken up and used as training input.

In our tests we decided to only used to split data into test and train datasets with 80-20 proportion.

Networks

We have leveraged two pre-trained architectures provided by Facebook AI:

- Wav2Vec 2.0
- HuBERT

Wav2Vec 2.0

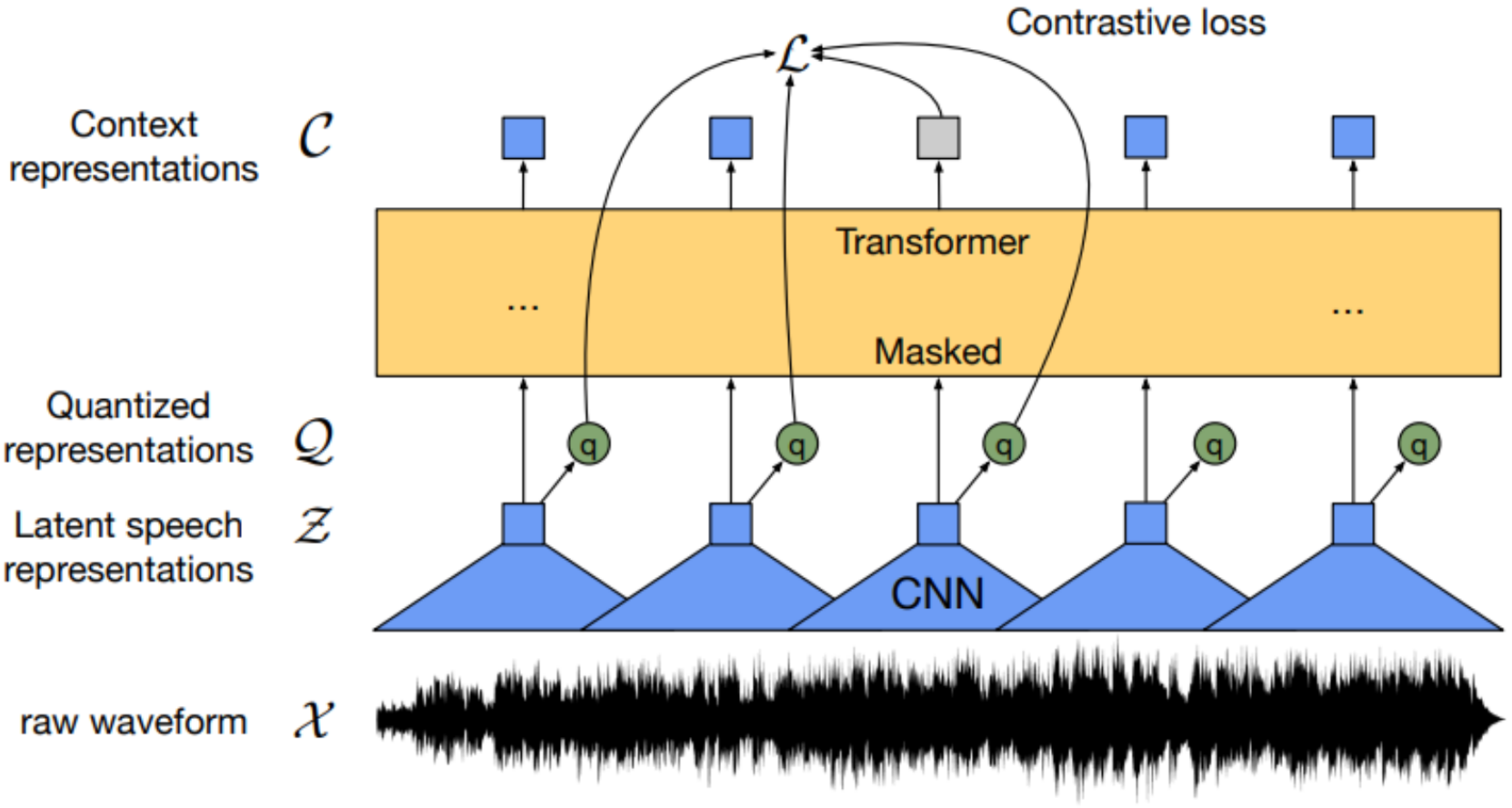

Figure: Wav2Vec [1]

HuBERT

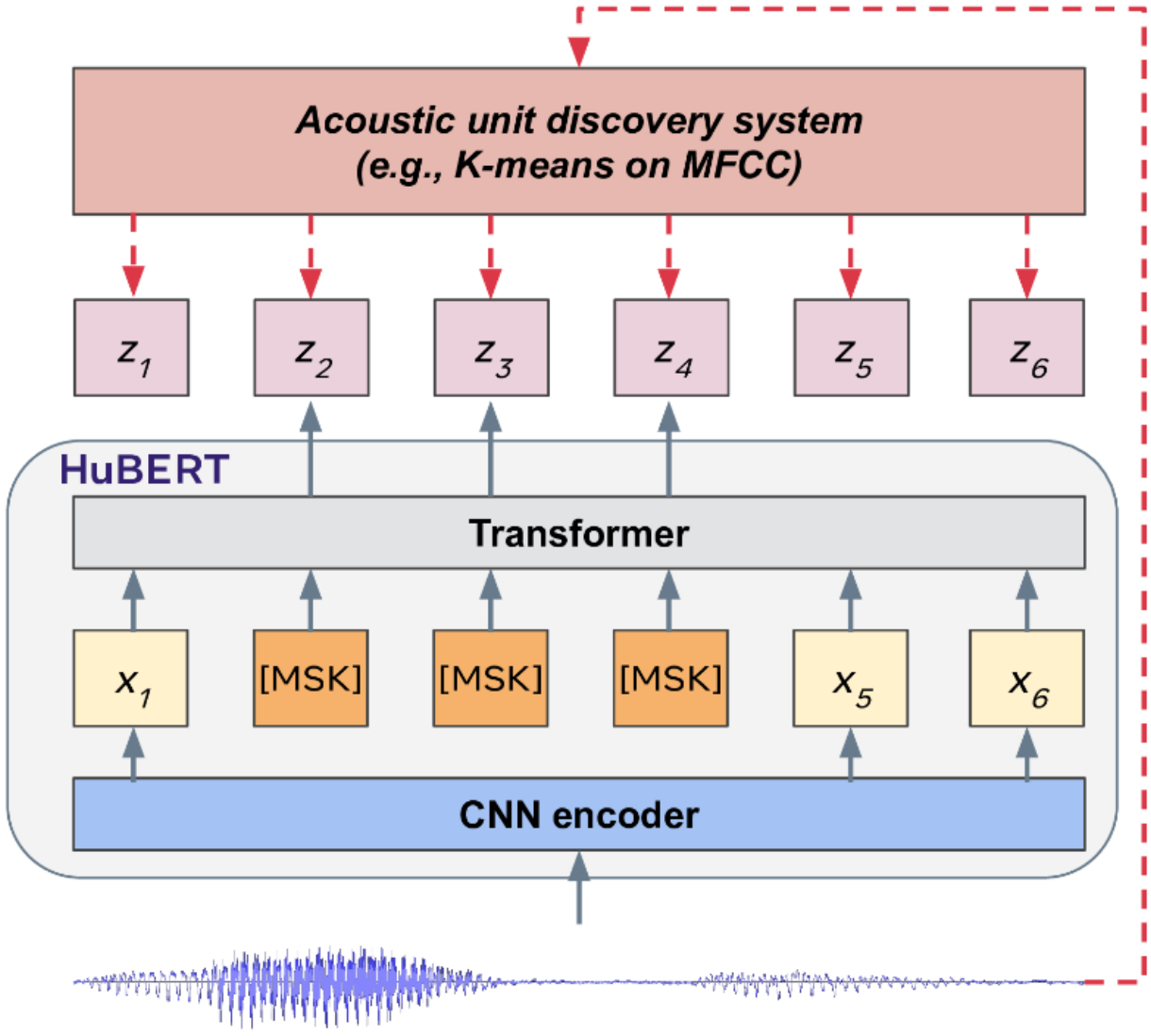

Figure: Hubert [2]

Network parameters

Training process parameters:

- *epochs* - The number of epochs during training. An epoch represents one complete pass through the entire training dataset.
- *learning_rate* - This parameter controls how much the weights of the network are updated during training.
- *batch_size* - The batch size specifies the number of training samples to be fed to the network in one forward and backward pass.
- *warmup_ratio* - The proportion of training steps dedicated to "warming up" the learning rate. During the warmup phase, the learning rate gradually increases from a small value to its target value, allowing the model to stabilize before proceeding with full training.
- *gradient_accumulation_steps* - How often gradients are accumulated before updating the model's weights. It allows for larger effective batch sizes, which can be beneficial for training with limited GPU memory.

Used datasets

1. *train* - the original training dataset from the task description. We have split this set to our own train and test sets with 80/20 ratio. The only modification we made to this set is adding the *silence* class, which we did by splitting audio files from the *_background_noise_* folder into one second bits.
2. *train-unknown* - after adding the silence class, we also wanted to test out a different way of training our models. We have moved every file not belonging to the relevant classes to a separate class called *unknown*.

Research hypotheses

1. Models trained on the Wav2Vec architecture will achieve higher accuracy scores than the ones trained on HUBERT.
2. Merging all insignificant classes to one class named "unknown" will allow the models to achieve higher accuracy scores.
3. Increasing the learning rate parameter will make the models achieve higher accuracy scores.

Conducted experiments and results

1. Network architecture comparison - we built two machine learning models based on the aforementioned architectures and compared their accuracy.
2. Merging unknown class - we have also tested if using the *train-unknown* dataset would produce better results.
3. Learning rate parameter increase - we wanted to test the influence of changing hyper-parameters during training on the overall accuracy of the model.

Network architecture comparison

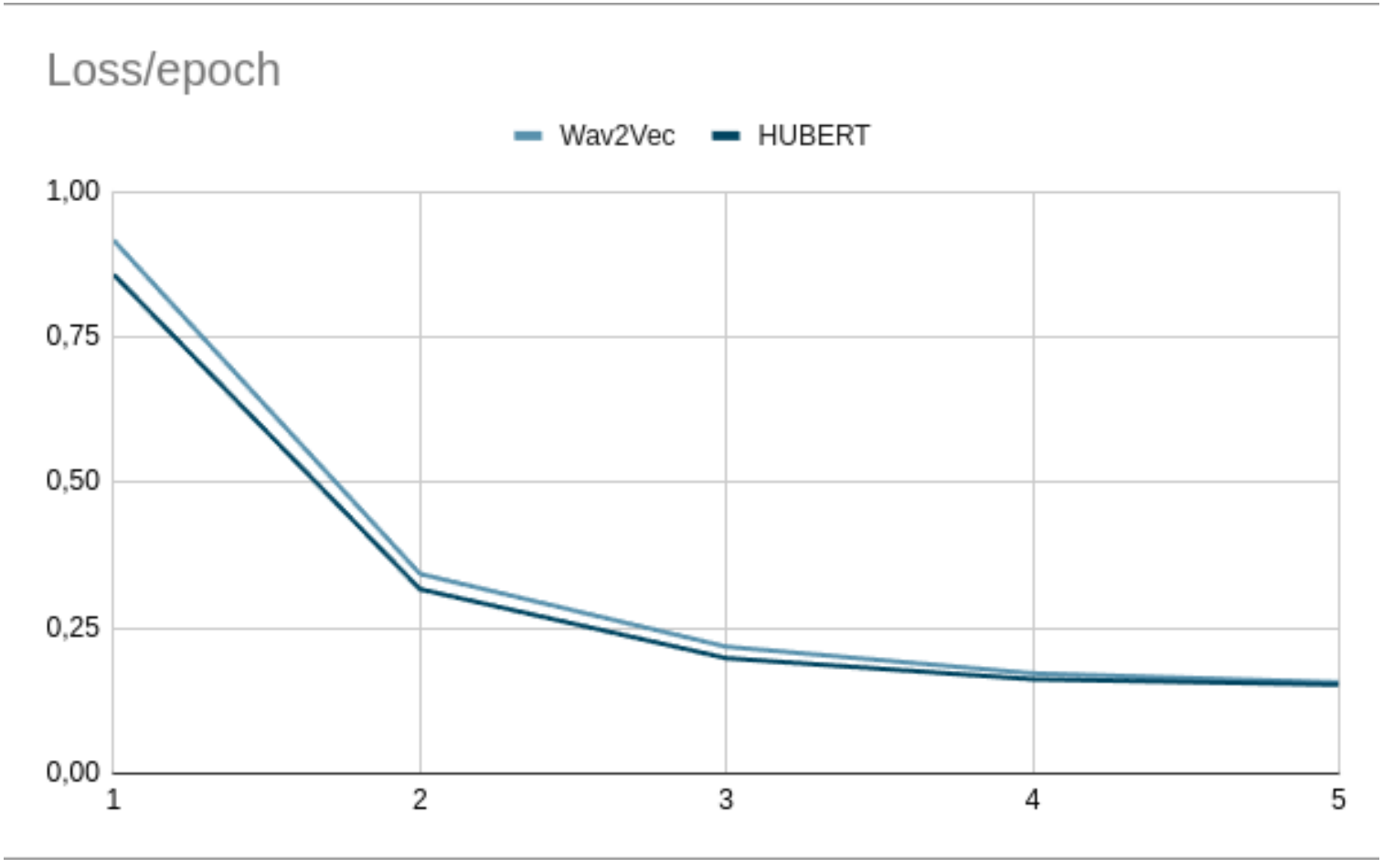

Figure: Loss function value per epoch

Network architecture comparison

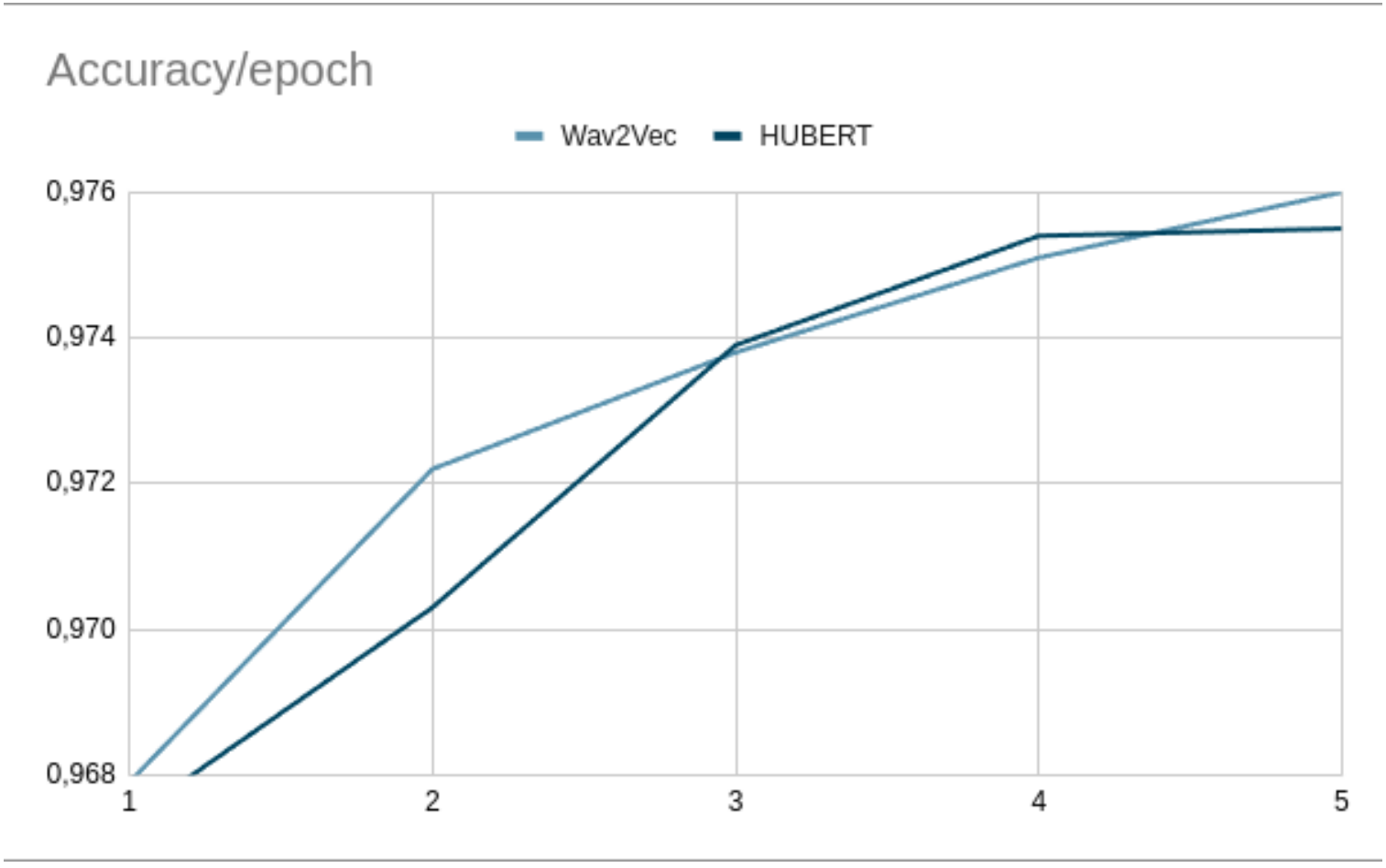

Figure: Accuracy value per epoch

Network architecture comparison

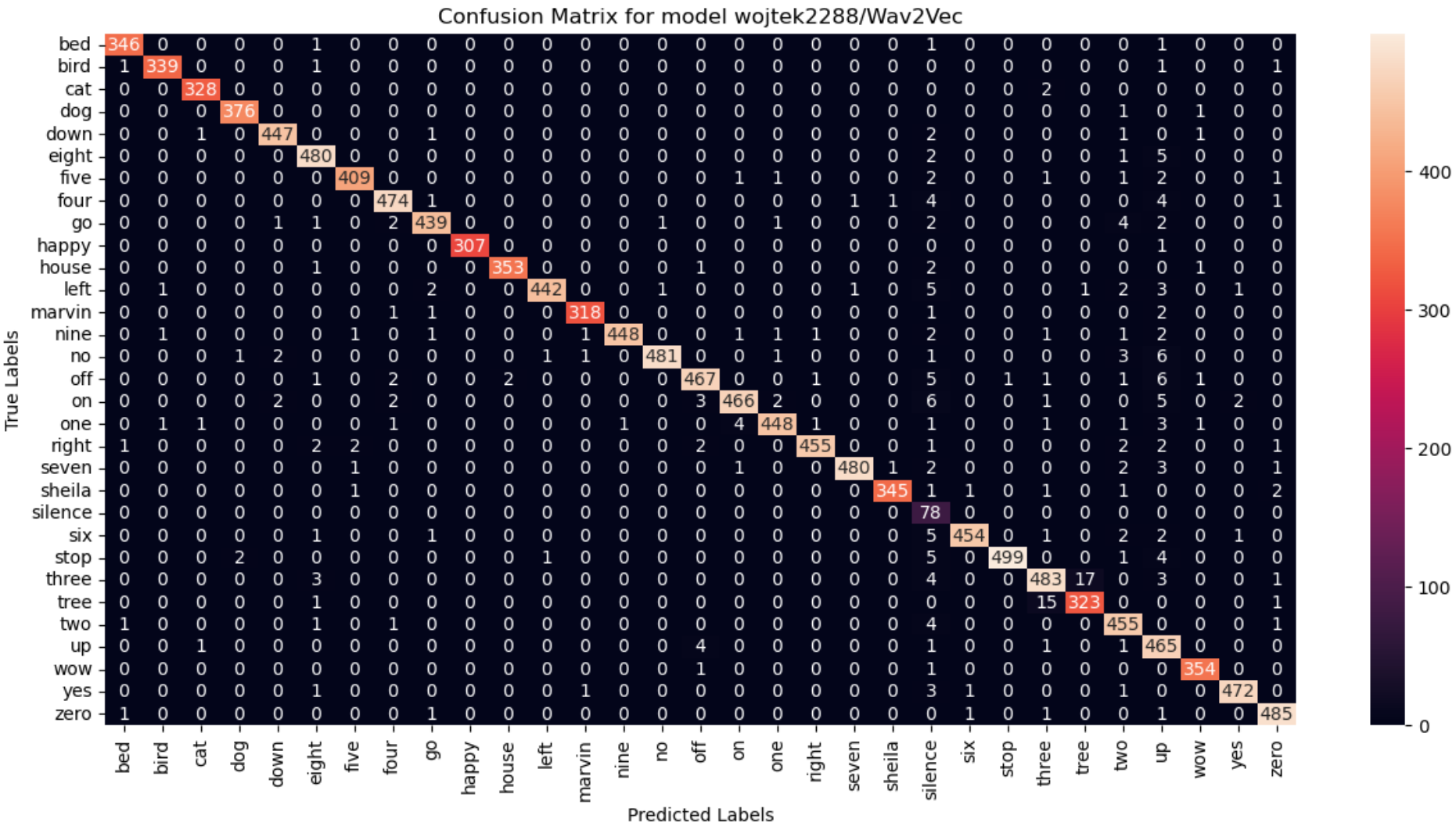

Figure: Confusion matrix for the Wav2Vec model

Network architecture comparison

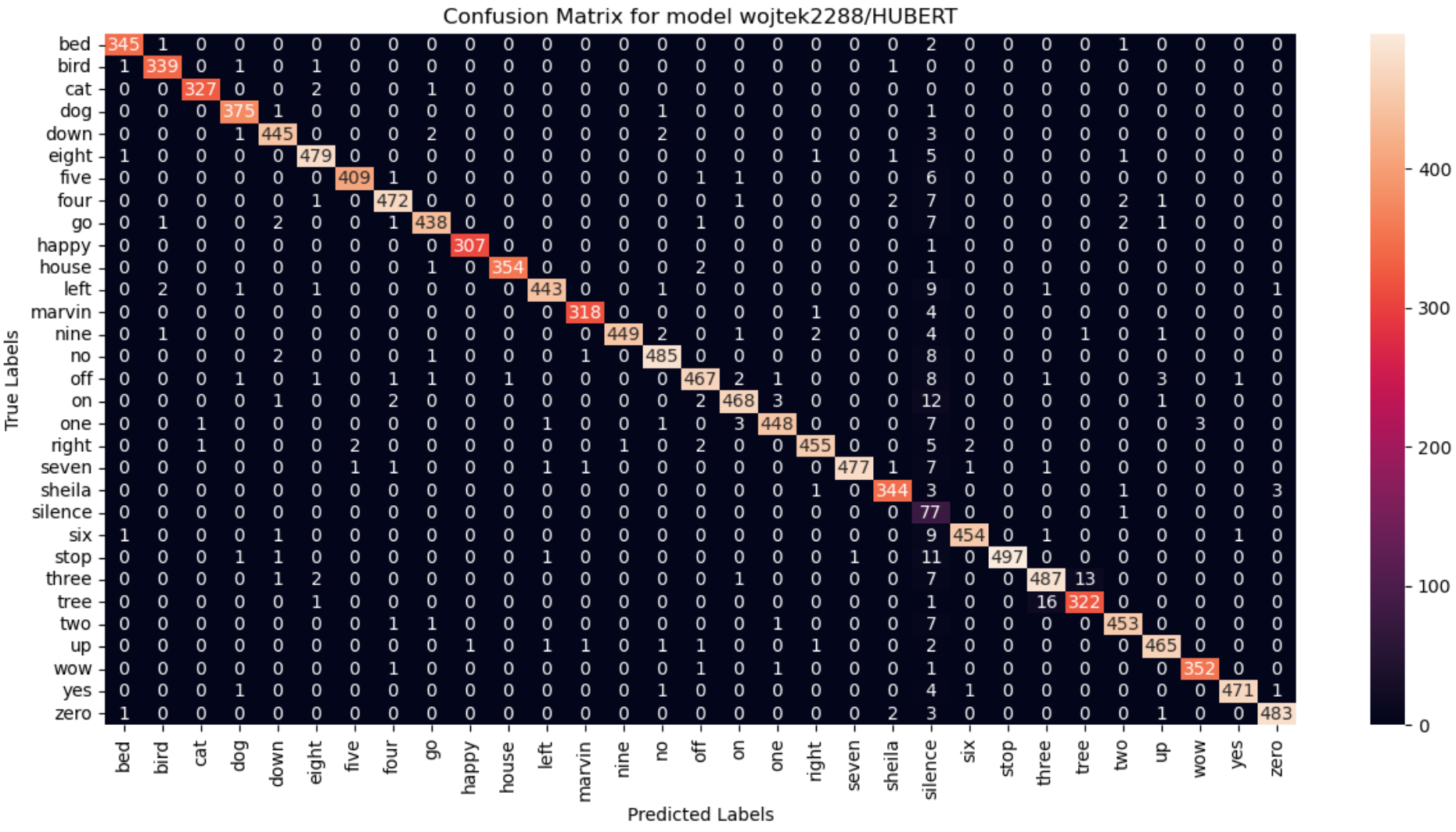

Figure: Confusion matrix for the HUBERT model

Network architecture comparison

The overall accuracy for both models on training and test data was as follows:

1. Wav2Vec - 0.98 for training and test,
2. HUBERT - 0.98 for training and test.

Merging unknown class

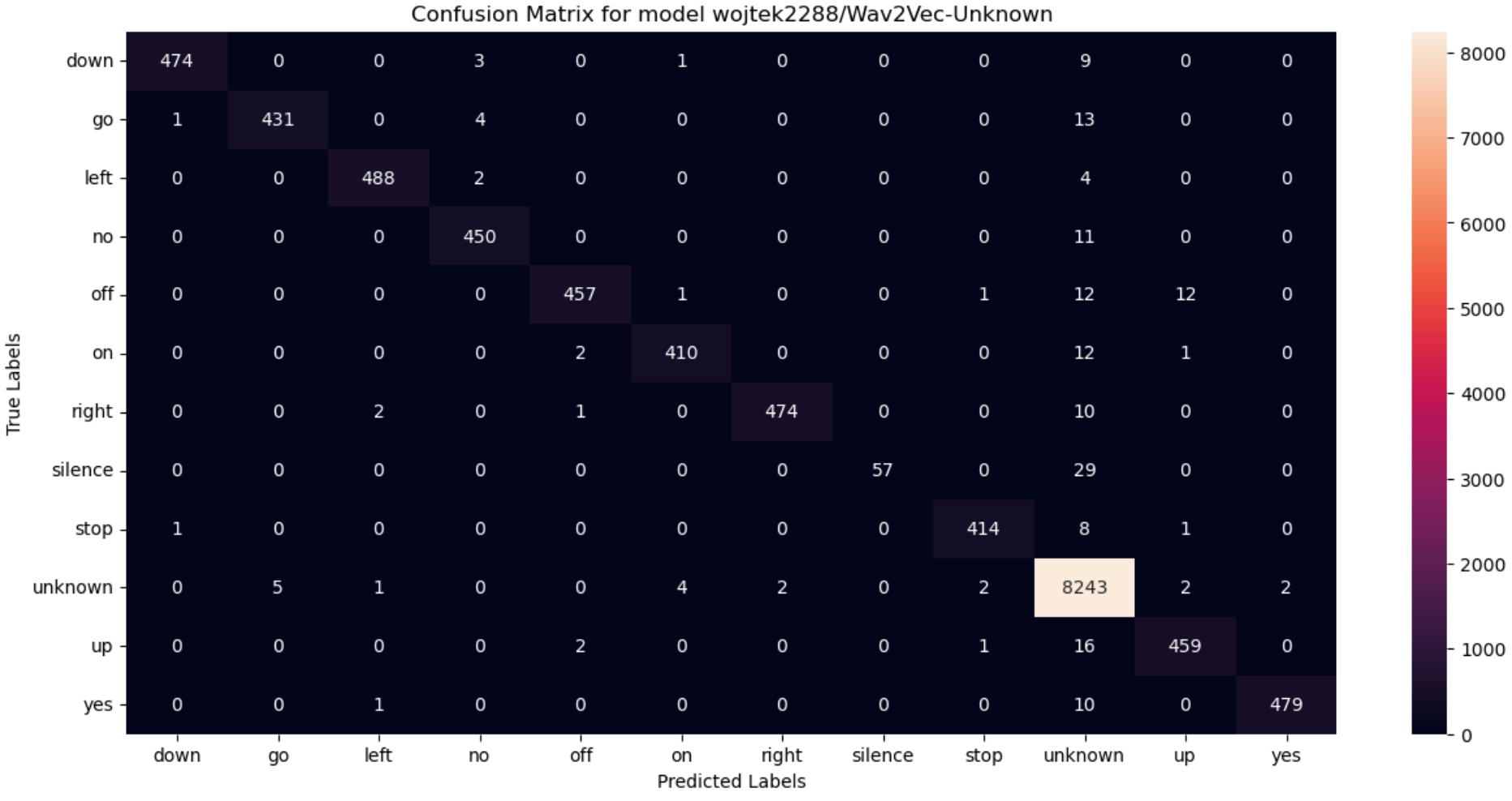

Figure: Confusion matrix for the Wav2Vec model with merged unknown class

Merging unknown class

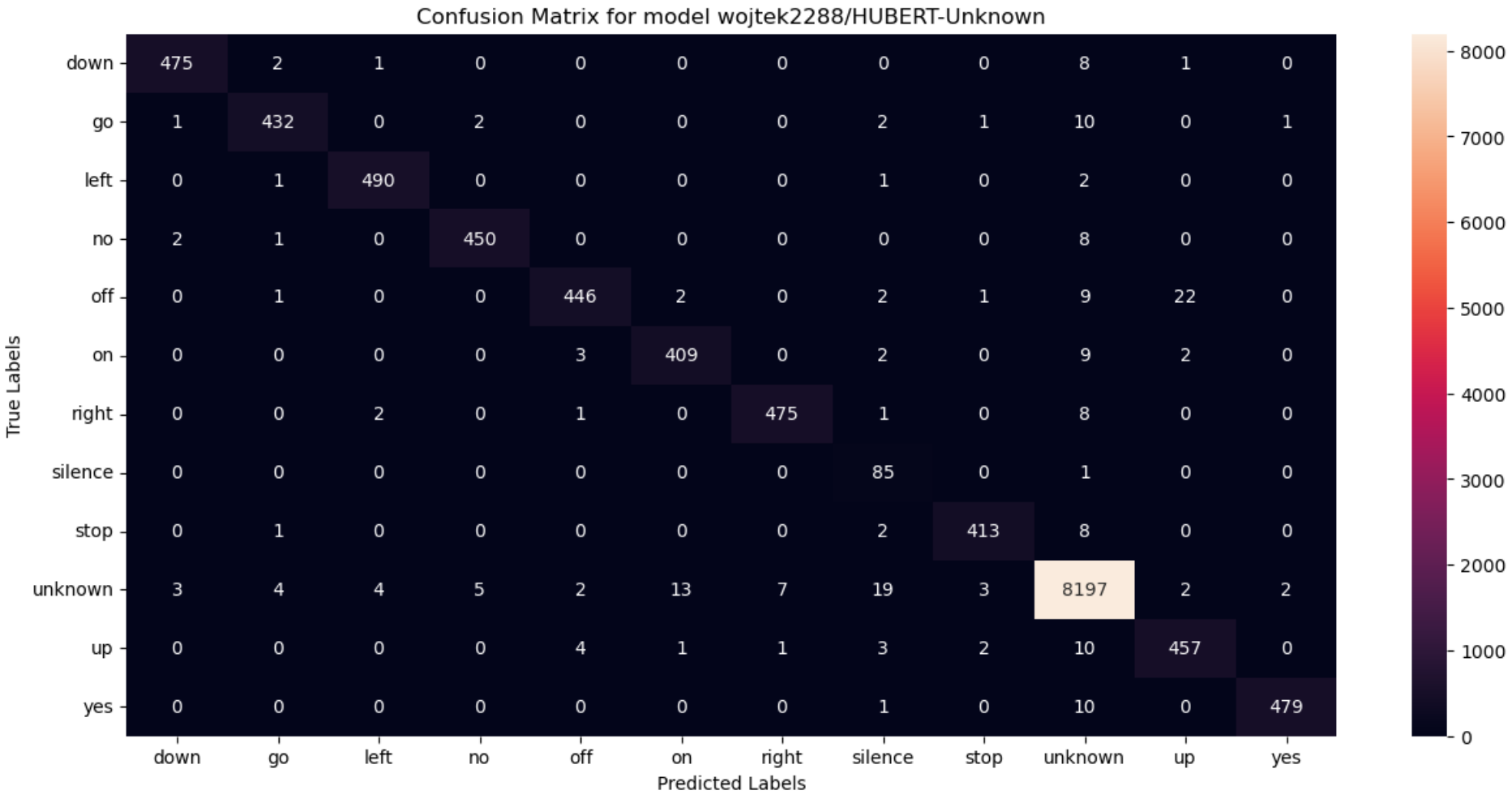

Figure: Confusion matrix for the HUBERT model with merged unknown class

Merging unknown class

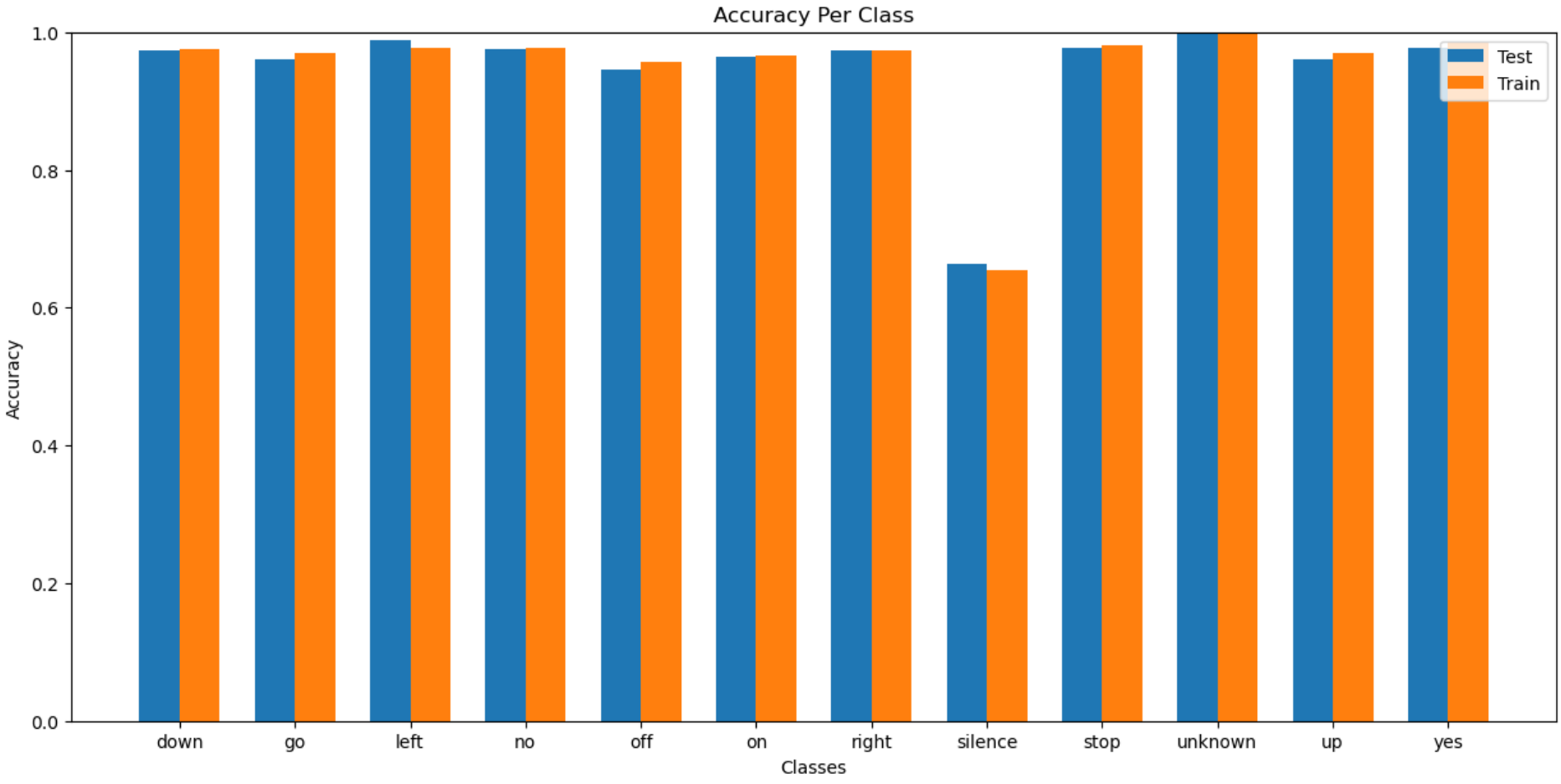

Figure: Accuracy per class for the Wav2Vec model with merged unknown class

Merging unknown class

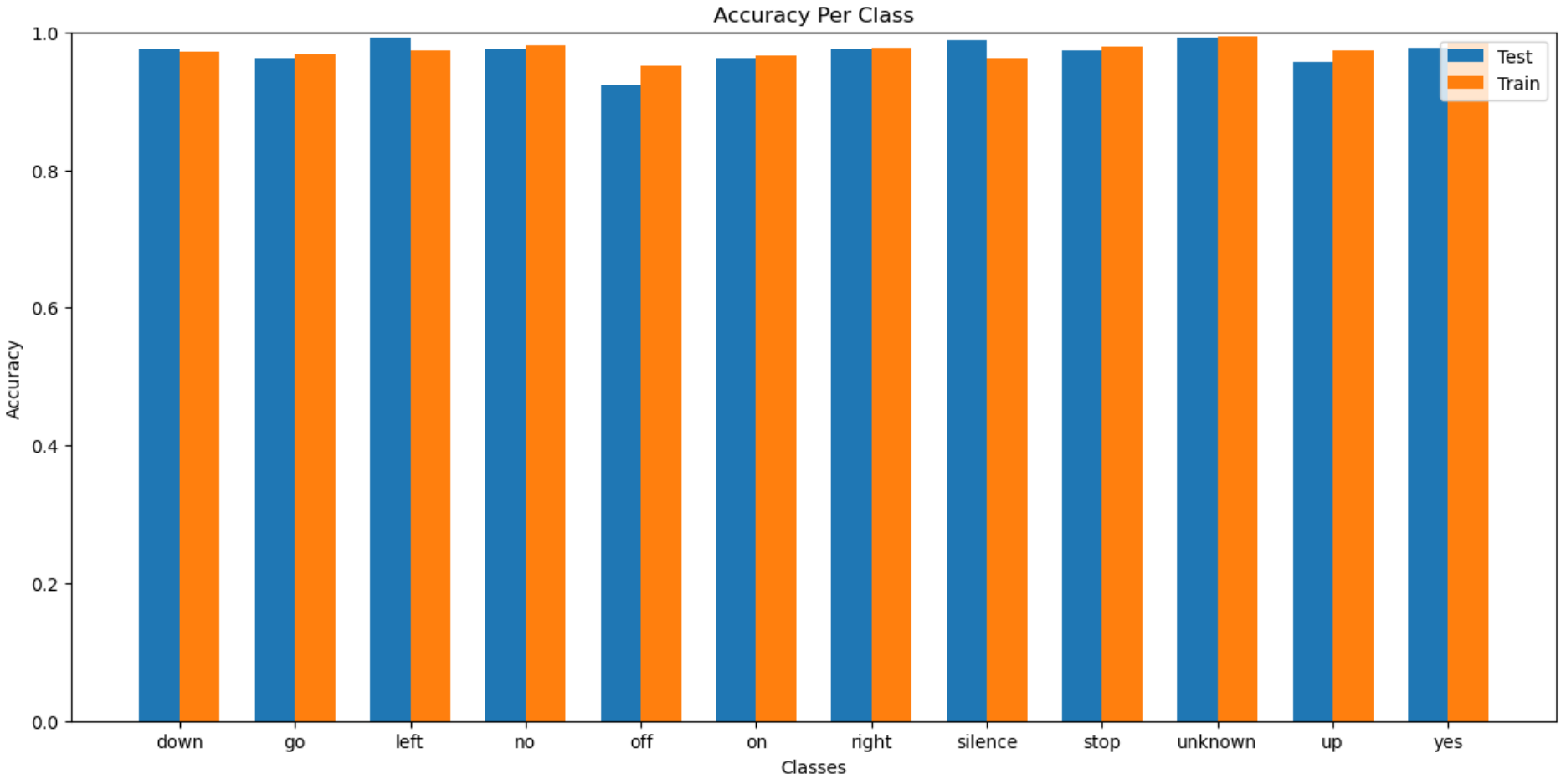

Figure: Accuracy per class for the HUBERT model with merged unknown class

Merging unknown class

The overall accuracy for both models after merging the classes on training and test data was as follows:

1. Wav2Vec - 0.99 for training and test,
2. HUBERT - 0.99 for training and for 0.98 test.

Learning rate parameter increase

We tested two different learning rate values for both of our models. They were as follows:

- $3 * 10^{-5}$,
- $1 * 10^{-4}$.

The overall accuracy did not change and totalled:

1. Wav2Vec - 0.98 for training and test,
2. HUBERT - 0.98 for training and test.

Learning rate parameter increase

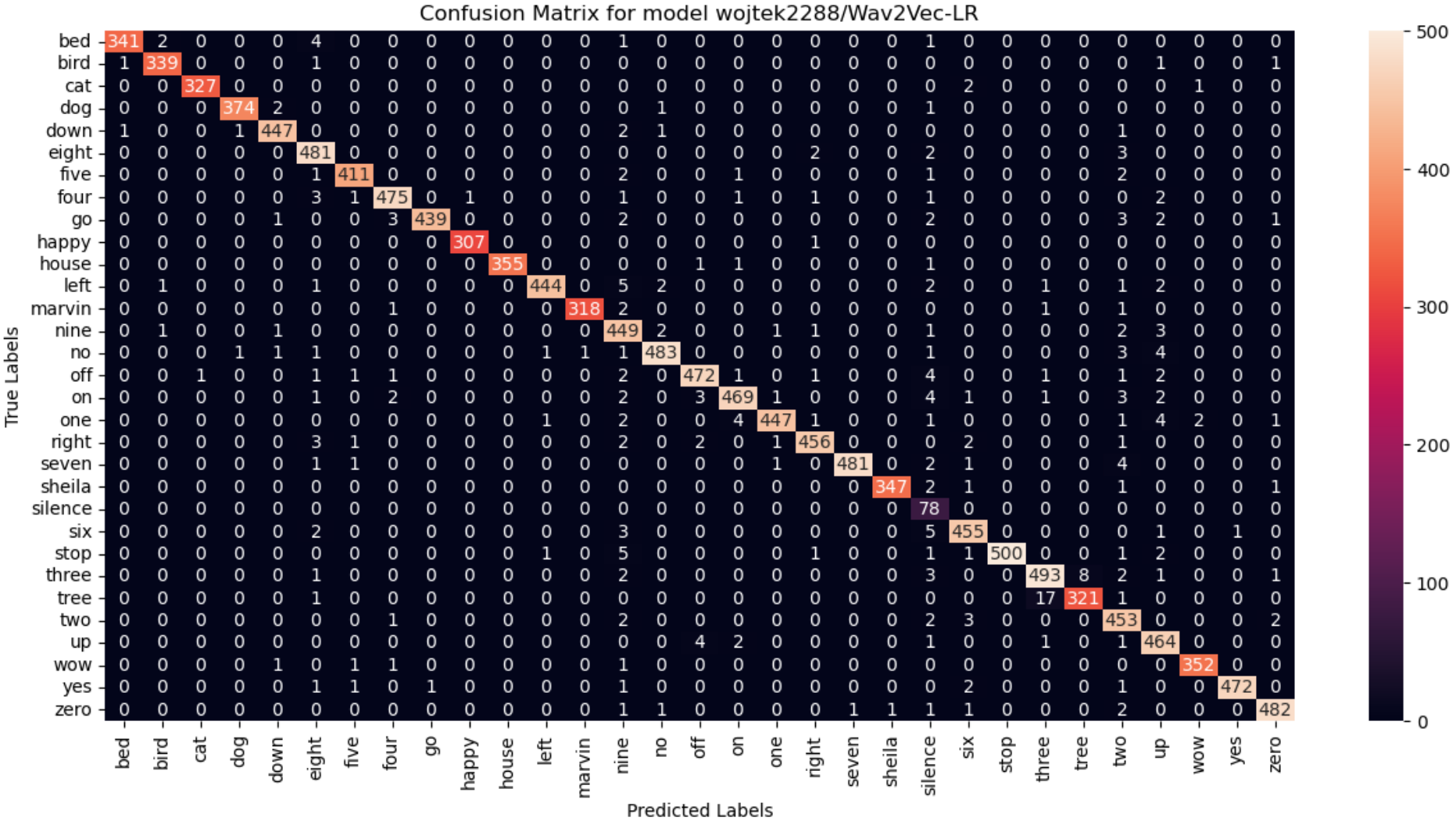

Figure: Confusion matrix for the Wav2Vec model with higher learning rate

Learning rate parameter increase

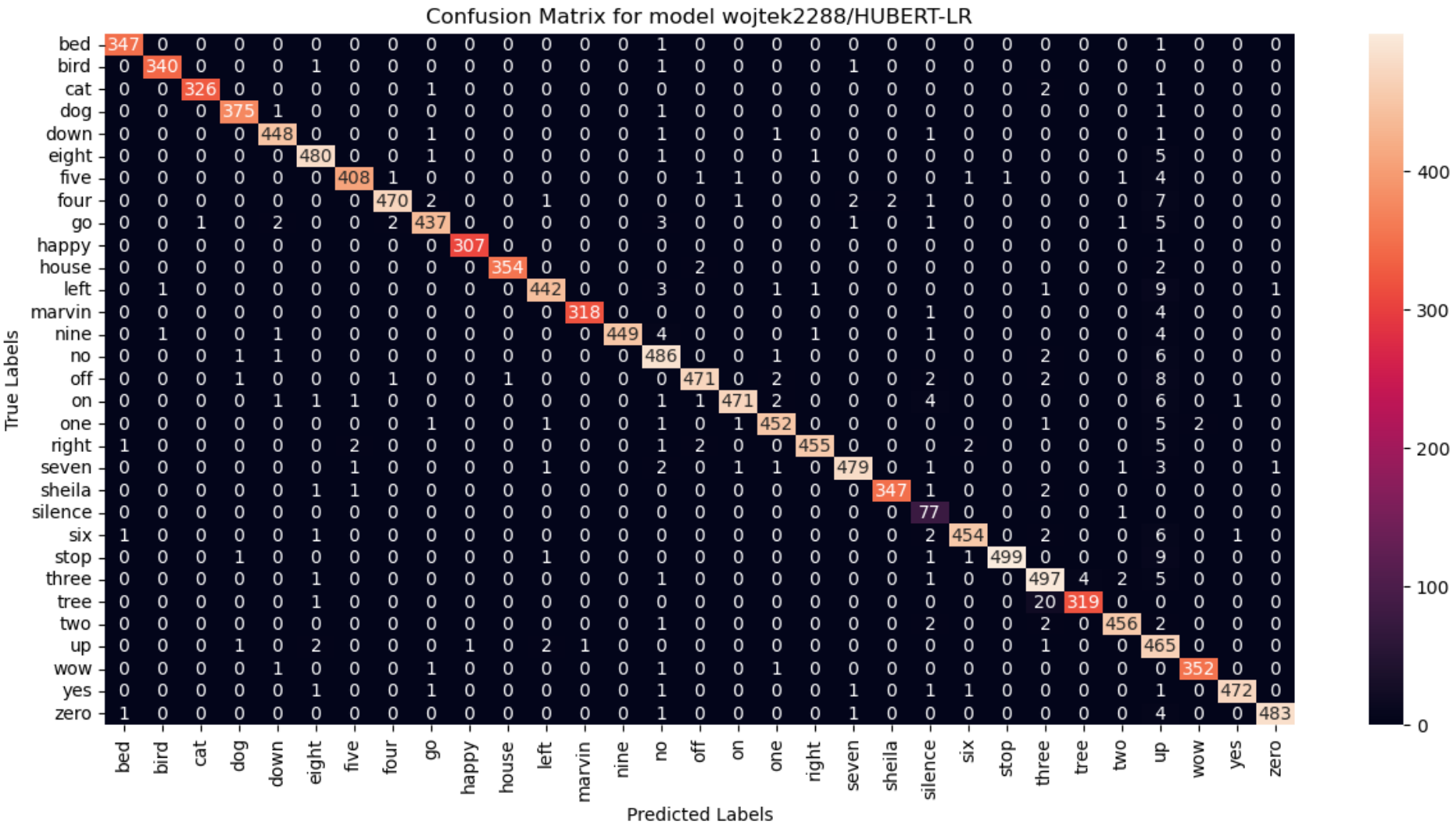

Figure: Confusion matrix for the HUBERT model with higher learning rate

Hypotheses Verification

With the results of our conducted experiments we are able to verify our hypotheses:

1. Models trained on the Wav2Vec architecture will achieve higher accuracy scores than the ones trained on HUBERT - **REJECTED**. Both models performed very similarly and we cannot point out the better one.
2. Merging all insignificant classes to one class named "unknown" will allow the models to achieve higher accuracy scores - **CONFIRMED**. Lowering the number of classes improved our results slightly.
3. Increasing the learning rate parameter will make the models achieve higher accuracy scores - **REJECTED**. Changing the learning rate parameter in both models did not impact the overall accuracy scores of these models.

Conclusions

We can conclude our experiment with the following remarks:

1. Models based on the transformer architecture are well suited for audio classification and speech recognition tasks.
2. In terms of achieved results, we did not find major differences between the Wav2Vec and HUBERT architectures.
3. A surprising factor for us was the fact that changing the learning rate parameter did not influence the results.
4. The second dataset, when we merged many classes into one, gave us a higher overall accuracy for both of our models.

Bibliography

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli.
wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

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, 2021.