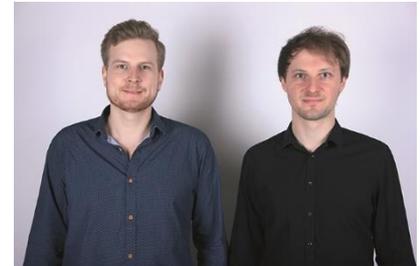


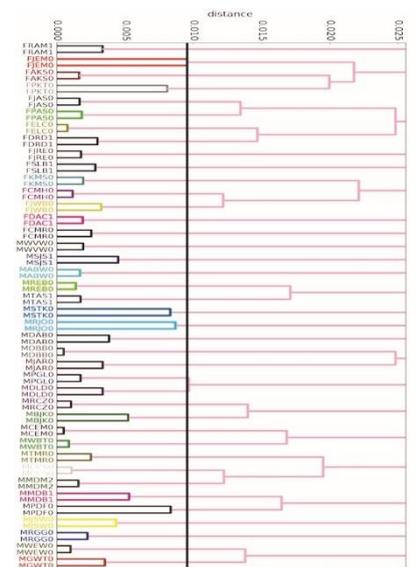
### Machine Learning for Speaker Clustering

Untersuchungen der letzten Jahre haben gezeigt, dass sich Neural Networks, im Speziellen die Unterart Convolutional Neural Networks (CNNs), für das Sprecherclustering eignen. So wurden die Fähigkeiten traditioneller Machine Learning Methoden wie die des Gaussian Mixture Model (GMM), kombiniert mit Mel-Frequency Cepstral Coefficients (MFCC), durch die der Neural Networks deutlich übertroufen. Eine weitere vor allem in der maschinengestützten Spracherkennung und Transkription eingesetzte Unterart der Neural Networks sind die Recurrent Neural Networks (RNNs). Allerdings finden diese im Problembereich des Sprecherclustering bis anhin kaum Beachtung, obwohl sie die dafür geeignete Fähigkeit haben, speziell Merkmale aus der Sequenz der Audiodaten (d.h. der zeitlichen Reihenfolge) zu lernen. Diese Bachelorarbeit befasst sich mit der Frage, ob sich RNNs und deren Kombination mit anderen Unterarten von Neural Networks für das Sprecherclustering eignen. Dazu wurden zwei verschiedene Neural Network Architekturen untersucht. Eine der beiden Architekturen besteht aus einem Bidirectional Long Short-Term Memory (BLSTM) Neural Network, welches für die Sprecheridentifikation schon in vorhergehenden Arbeiten sehr gute Resultate zeigte. Die andere Architektur ist eine Kombination aus CNN und Gated Recurrent Unit (GRU) Layern. Als Inputdaten wurden den Neural Networks Audiosignale in Form von Spektrogrammen bereitgestellt. Die beiden Architekturen wurden mit dem TIMIT Datensatz trainiert und getestet. Resultate der BLSTM Architektur ergaben ein fehlerfreies Clustering bei 40 Sprechern, was zum Zeitpunkt der Durchführung von keinem anderen Machine Learning Ansatz aus vergleichbaren Arbeiten erreicht wurde. Es konnte auch gezeigt werden, dass ein Clustering mit mehr als 40 Sprechern gute Resultate erzielt.



Diplomierende  
Patrick Gerber  
Sebastian Glinski

Dozierende  
Oliver Dürr  
Thilo Stadelmann



Dendrogram des Sprecherclustering  
für 40 Sprecher