Self-Supervised Learning for Automatic Speech Recognition

MSc Natural Language Processing

M1 Supervised Project

by

Zakaria Berbara

Nasser-Eddine Monir

Eric Papain Mezatio

under the guidance of

Imran Sheikh

Academic year 2020-2021

Chapter 1

Introduction

Automatic speech processing is a field of artificial intelligence covering several tasks such as speech recognition, speech detection, language identification and speaker identification. Among these, one of the first speech recognition systems was designed by Bell Labs researchers in 1952 to identify isolated numbers spoken by a single speaker [1]. The speech recognition systems of the 1950's did not exceed vocabulary size of more than ten words and speech recognition technologies evolved little during this period. The first systems for processing continuous speech appeared in the late 1960's with work on a voice command system for playing chess and the design of the Dynamic Time Warping (DTW) algorithm by researchers for a system with a vocabulary of about 200 words. In the 1970's, there was a major revival of research in the field of speech recognition thanks to DARPA's 'Speech Understanding Research' system, which revealed the use of cepstral analysis and Hidden Markov Model (HMM); which are still widely used in systems today. In the 1980's, team at IBM built the Tangora system based on HMMs and having a vocabulary of twenty thousand words. The progress in speech recognition continued and the latest technological breakthrough happened in the early 2010 with the success of Deep Neural Networks (DNN). But, it should be noted that all these advances were only made possible thanks to the exponential progress made on the computing capacity of computer resources used by researchers.

Most advanced ASR systems are trained on large amount of speech data and associated text transcriptions. Thanks to efforts from the speech research community significant amount of read speech datasets have been made available in several languages. However, ASR models trained on read speech datasets are known to perform poor on spontaneous conversational speech. Moreover, collection and manual transcription of spontaneous conversational speech is known to be highly expensive. With nearly 7,000 languages spoken all over the world [2], such approaches for building ASR systems have managed to cover only a few tens of languages.

The speech research community has recently started putting efforts on developing methods to learn generic speech representations from unlabeled audio sources using deep learning methods [3, 4]. These types of learning methods, recently termed as self-supervised learning methods, have also proven their effectiveness in other areas, for instance natural language processing [5] and computer vision [6, 7]. Drawing motivation from this research, this project will study self supervised learning for ASR. More specifically, it will focus on self supervised methods for learning phoneme like units from unlabeled speech data.

The rest of this report is organized as follows. Chapter 2 presents concepts related to feature extraction from speech signal, automatic speech recognition and artificial neural networks. Chapter 3 begins with an introduction to self-supervised learning and then describes details on blind phoneme segmentation. This is followed by a brief description of state-of-the-art methods to learn speech representations in a self-supervised manner. The report concludes with Chapter 4, which presents points concerning the implementation of the methods studied under this project.

Chapter 2

Background

This chapter briefly presents some concepts related to feature extraction from speech, speech recognition and neural networks which would form the basis of methods being studied in this project.

2.1 Feature Extraction

In order to perform any form of automatic speech processing, the first step is to extract relevant features from a raw speech signal. There are several different methods to do so and the choice of a speech feature extraction method is based on its effectiveness and robustness on the target task. Mel Frequency Cepstral Coefficents (MFCCs) [8] are a widely used features for ASR. Introduced in the 1980's, and for a long period considered as the state-of-the-art, it focused on representing the information which human listeners would find important.

Figure 1 shows a block diagram for the steps involved in extracting MFCC features from raw continuous speech signal. To obtain MFCCs, first the speech signal is framed into short 20-40 ms frames, with an assumption that on short time scales the signal doesn't really change and remains statistically stationary. Then the power spectrum of each frame is calculated using Fourier transform. The human ear (cochlea) resolves frequencies non-linearly across the audio spectrum and in order to obtain the same non-linearity, the power spectrum is transformed using a Mel filterbank. Finally, a compression operation is performed by taking the logarithm of the filterbank energies and by decorrelating them with the Discrete Cosine Transform (DCT).

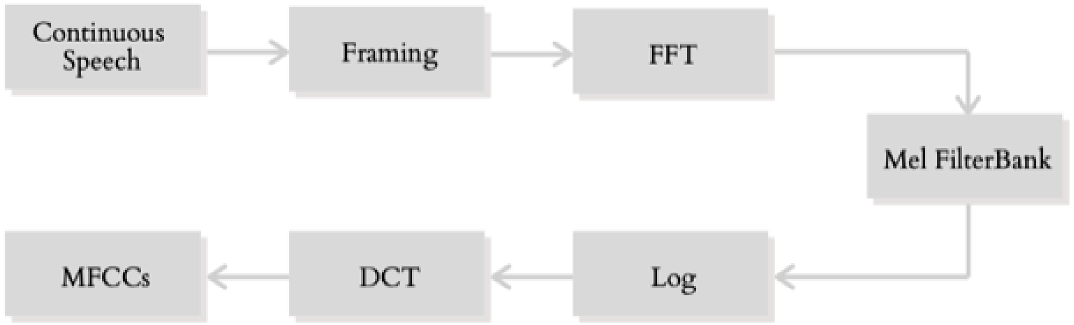

Figure 1: Block diagram for extraction of MFCC features from speech signal.

2.2 Speech Recognition

The main objective for an ASR decoder is to decode the speech signal into an optimal word sequence W^* among all possible word sequences. Denoting, the sequence of MFCC features extracted from the raw speech signal as $X = (x_1, \ldots, x_T)$, the decoding objective is given as:

$$W^* = \underset{W}{argmax} P(W|X) \tag{1}$$

By using the Bayes Formula, it is in fact possible to formulate a new equation from this previous one.

$$W^* = \underset{W}{argmax} P(X|W)P(W) \tag{2}$$

This equation splits the ASR problem into two parts, namely the acoustic model and the language model. $P(W)$ represents the likelihood of the sequence of words spoken and are computed by a language model. The language model is learned from a collection of text data.

The ASR acoustic model is responsible for mapping acoustic feature vectors to linguistic units like phonemes and words. In other words, it computes the likelihood of a feature vector X given a model of the linguistic unit W, i.e $P(X|W)$. An acoustic model is typically a sequence model like the Hidden Markov Model (HMM). HMMs are graphical models that are indeed efficient and flexible to model and infer temporal pattern recognition from sequential data like those embedded in speech recordings and handwriting images. They are composed of a Markov chain, which is considered as the hidden part, in addition to an observable process that is probabilistically dependent on the Markov chain. Traditionally, each phoneme is modeled by an HMM comprising of 3 to 5 states where each state is represented by a Gaussian Mixture Model (GMM) which models the distribution of observed speech feature vectors in the corresponding speech signal [9]. Modern approaches for ASR, replace the GMM with a more powerful DNN model.

2.3 Neural Network

Neural networks are computational models that try to imitate the functioning of human nervous system in order to solve complex mathematical problems. There are several types of neural network that can be configured according to the tasks to be solved. For instance, convolutional neural networks emulate receptive field of the human eye excelling at image processing tasks, whereas recurrent neural networks can readily handle sequential data like text, speech, protein sequences, etc.

2.3.1 Simple Feedforward Neural Network

Feedforward neural networks are called so because activations in this network flow from the input nodes, then through the hidden nodes or hidden layer and finally to the output nodes or output layer. As data passes through the artificial mesh of the network, each layer processes one aspect of the data, filters out outliers, detects familiar entities and produces the final output. Among feedforward neural networks, Multilayer Perceptrons (MLP) were the initial and successful neural networks.

Figure 2 presents a pictorial representation of an MLP. The description of this MLP is enumerated below.

- **Input Layer (Layer 0):** This layer represents the layer of neurons that receive inputs and pass them on to other layers. The number of neurons in this layer must be proportional to the number of attributes or features representing the input data $(X_1, X_2, ..., X_n)$.
- **Hidden Layers(Layer 1, Layer 2):** These layers are included between the input layer and the output layer and they constitute the complexity of the network. They are the hidden layers of the model. They contain a large number of neurons which apply transformations to the inputs before passing them on to the next layers and so on up to the output layer.

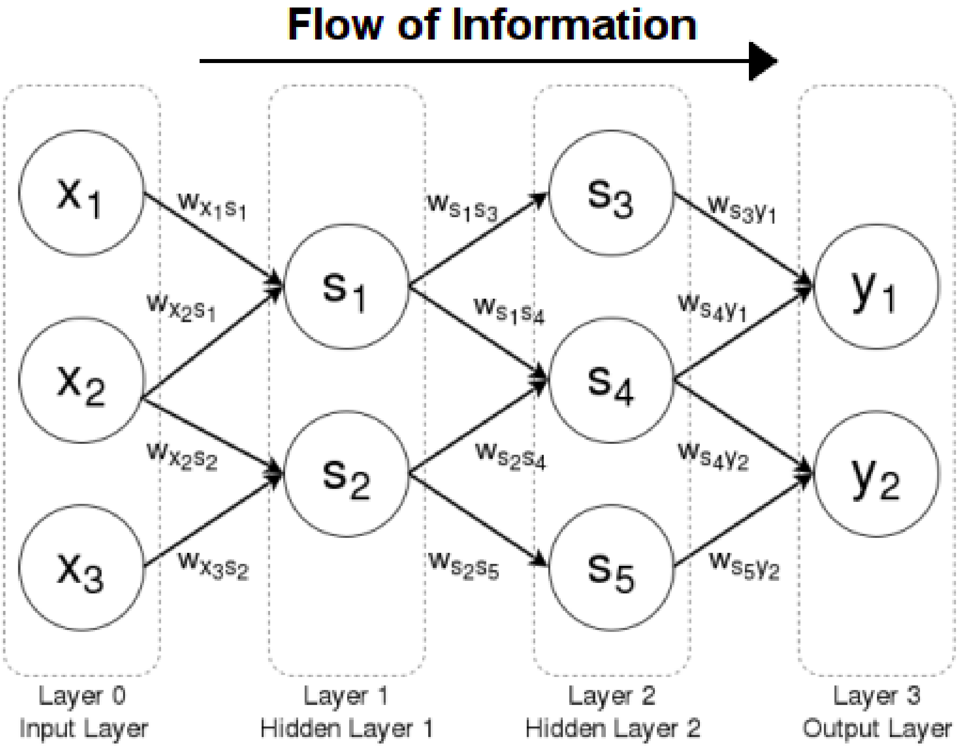

Figure 2: A multilayer perceptron. (From: `https://brilliant.org/wiki/feedforward-neural-networks/`)

- **Output Layer (Layer 3):** This layer transforms the output of the final hidden layer into output activations.

- $W_x = \{w_{x_1s_1}, w_{x_2s_1}, ...\}$ represent weights that transform incoming activations (x_i) to outgoing activations (s_j). Similarly, $W_s = \{w_{s_1s_3}, w_{s_2s_3}, ...\}$ represent weights that transform incoming activations (s_i) to outgoing activations (s_j) and $W_y = \{w_{s_3y_1}, w_{s_3y_2}, ...\}$ represent weights that transform incoming activations (s_i) to output activations (y_j)

Such an MLP can be mathematically formulated as:

$$S = \sigma(W_x\ X) \quad \text{for Layer 1} \tag{3}$$

$$S = \sigma(W_s\ S) \quad \text{for Layer 2} \tag{4}$$

$$y = \text{softmax}(W_y\ S) \quad \text{for Layer 3} \tag{5}$$

2.3.2 Recurrent Neural Network

Recurrent Neural Network (RNN) are special type of neural networks designed to handle sequential inputs [10]. Figure 3 presents a pictorial representation of an RNN. RNNs can be mathematically formulated as:

$$h_t = \sigma_h(W_h\, h_{t-1} + W_x\, x_t) \tag{6}$$

$$y_t = \sigma_y(h_t\, W_y) \tag{7}$$

where, x_t represents the input sample at time step t, h_t represents the corresponding hidden layer representation of the RNN, and y_t is the corresponding output. σ represents a non-linearity function, which could be different at the hidden and the output layers.

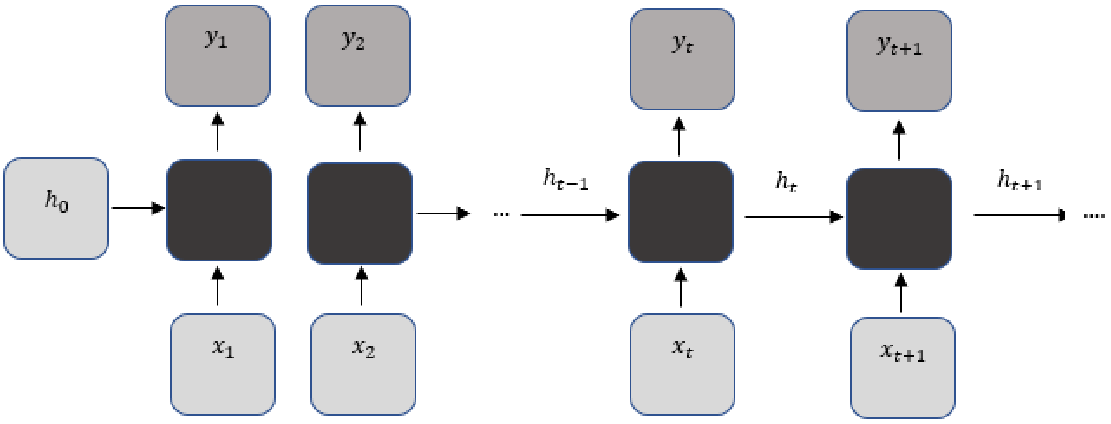

Figure 3: Recurrent neural network with inputs, hidden states and outputs unrolled over time.

Chapter 3

Self-supervised Learning

Advances in machine learning, particularly in deep learning and neural networks, in the last decade have led to the best performing systems in computer vision, natural language processing, robotics, audio signal processing and many other fields and applications. However, it is very well known that the success of these learning methods came out very large quantities of human labeled or annotated datasets. Hence the term 'Supervised Learning', as the training of the models relies on explicit labels or supervision. At the same time the research community has repeatedly recognized the importance of 'Unsupervised Learning' and 'Semi-supervised Learning' methods. While the former learns distributions and/or representations from completely unlabeled datasets, the later tries to achieve representations and models effective on a given task using a combination of (lesser) labeled and (more) unlabeled datasets.

Self-supervised learning[1] is another approach evolving out of the deep learning paradigm. It builds on the ideas that (a) deep learning models can learn powerful representations at intermediate layers of the model, and (b) learning can be more effective with explicit supervision, even if the supervision signal is not a task specific label. More interestingly, the supervision signal can come from a data sample itself or from a known transformation of the data sample. For instance,

- in some methods, a part of the input data sample is masked and this masked part is to be predicted as output by using the remaining part as input. An example being prediction of masked words in a sentence [5].

- in other methods, an input sample undergoes one of the possible known transformations and the type of transformation is predicted from original and transformed data samples. An example being predicting rotation of images [7].

[1] alternatively also referred as 'Unsupervised Representation Learning'

Similar to the above mentioned examples for text and image processing, self-supervised learning has been tried in context of speech processing [3, 11, 12, 4, 13]. These include learning effective representations for speech recognition task [3, 11, 12] as well as for other tasks like speaker verification [4] and speech emotion recognition [13]. Based on the approach adopted for self-supervision the learning methods proposed in these prior work can be categorized as follows.

- Encoding speech into latent representations, followed by prediction of a future latent representation [11, 4, 13], similar to a predictive language model [14].
- Encoding speech into latent representations, followed by masking and prediction of masked representations [3]. This is similar to the text processing example mentioned above [5].
- Encoding speech into latent representations, followed by regression of features obtained using traditional speech processing methods [12]. This is similar to the image processing example mentioned above [7].

In this project, we are interested in the speech recognition task and more particularly on automatic phoneme segmentation using techniques and representations from self supervised learning approaches. Within this scope, we would like to briefly describe a prominent prior work on blind phoneme segmentation and two effective speech representations learned out of self-supervision. We would like to highlight that blind phoneme segmentation is itself a self-supervised approach.

3.1 Blind Phoneme Segmentation

The objective of blind or automatic phoneme segmentation is to accurately mark boundaries of phonemes appearing in a speech signal in an unsupervised manner. Unsupervised segmentation of speech signal into phoneme like linguistic units has been studied in different prior work [15, 16, 17, 18, 19]. The central idea of these methods can be generalized into the following steps:

1 Extract frame level spectral feature vectors from the raw speech signal. For example, MFCCs presented in Section 2.1.

2 Slide over the sequence of features and accumulate them into bigger windows or an accumulated representation.

3 Compare adjacent accumulated windows or representations to obtain similarity or distance graphs over time. Peaks or valleys in this graph represent spectral change points in the speech signal.

4 Select and refine the change points to obtain boundaries corresponding to phoneme like linguistic units.

These methods differ in the type of spectral features used, the accumulation of feature vectors, adjacent representation comparison method and also the boundary refinement techniques. For example, [15] uses feature averaging and euclidean distances, [16] uses margin methods, [17] computes Bayesian Information Criterion (BIC) on windows of features, [18] monitors spectral changes using Legendre polynomial approximation and [19] applies RNN directly over MFCCs.

Among all the methods listed above, we choose to adopt the blind phoneme segmentation method using RNNs [19] due to its simplicity and effectiveness. This work used two types of features. The first type of speech feature vector consists 13 dimensional MFCCs and the second one consists of one hot categorical features of 8 dimension, computed by performing a K-means clustering [20] on 10000 MFCCS frame randomly selected from the training set. Figure 4 presents a visualization of the phoneme segmentation

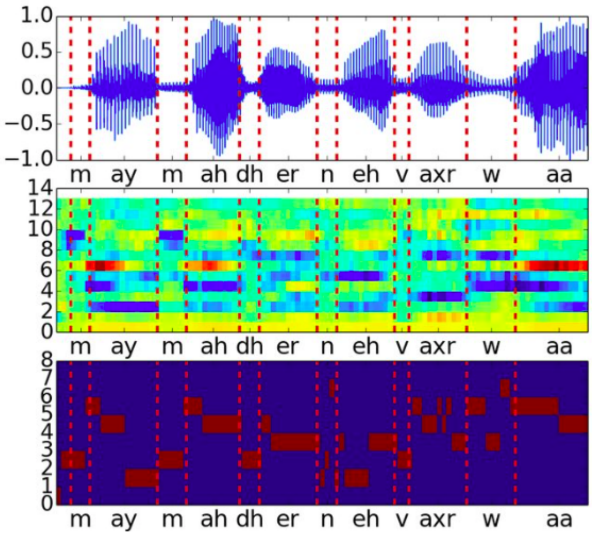

Figure 4: Visualization of phoneme segmentation on MFCC feature sequence (in middle) and categorical features sequence (at bottom) obtained after K-means clustering on MFCCs. (Figure taken from [19].)

With the two type of features, they proposed three types of models. Two of these models used the categorical feature vector of 8 dimension. The first model being a K-order Markov chain prediction model approximated as:

$$p_K(x_t \mid x_0^{t-1}) = \frac{1}{K}\sum_{i=1}^{K} p_1(x_t \mid x_{t-i}) \tag{1}$$

with the first order Markov model transition probabilities computed as:

$$p_1(x_t \mid x_{t-i}) = \frac{f(x_{t-i}, x_t)}{f(x_{t-i})} \tag{2}$$

where, $f()$ could be as simple as a count function on categorical features. Given this approximated Markov chain model the prediction graphs, as discussed in the step 3 above, can be obtained as:

$$\begin{align} E_{\text{markov}}(t) &= -\log\ p_K(x_t \mid x_0^{t-1}) \tag{3}\\ &= -\log \sum_{i=1}^{K} p_1(x_t \mid x_{t-i}) \tag{4}\end{align}$$

The second model based on categorical feature vectors makes use of an RNN model which is expected to perform better than an approximated Markov. The RNN model can be trained to predict the categorical one hot feature $\hat{x}_t$ given history captured in the hidden state h_t of the RNN, as represented by equation (7) in section 2.3.2. The training procedure would optimize the standard cross entropy loss function. During test, the prediction graphs can be obtained as:

$$E_{\text{RNN-cat}}(t) = -\sum_{i=1}^{d} 1_{x_t=i} \log\ p(\hat{x}_t \mid h_{t-1}) \tag{5}$$

The third model is based on MFCC feature vectors and makes use of RNN model to perform a regression of MFCC feature vectors. During training the RNN model would be optimized to minimize the root mean square error between the actual MFCC feature vector x_t and the predicted MFCC feature vector $\hat{x}_t$. The same function can be used to get the prediction graphs during test as:

$$E_{\text{RNN-MFCC}}(t) = \frac{||x_t - \hat{x}_t||^2}{d} \tag{6}$$

where d is the number of MFCCs in a feature vector.

3.2 wav2vec Representations

The wav2vec 2.0 model is one of the most recent work on obtaining effective speech representations using self-supervised learning [11]. The high performance of these representations on speech recognition task motivates us to experiment with them in our task. While the model training and architecture themselves are quite complicated, we try to present a brief description of the self-supervised learning approach adopted in this work.

Figure 5 presents a high level block diagram of this approach. As shown in this figure, raw audio waveform is first transformed by a feature encoder into latent representations (Z). The feature encoder is a stack of Convolutional Neural Networks (CNN). The latent representations are then presented to two blocks, each having its own training loss function. The transformer block uses the recently proposed transformer neural network architecture and functions similar to the BERT masked language model [5], wherein a part of the sequence is masked and then predicted using the unmasked parts of the sequence. The quantization block performs discretization of the latent representations (Z) which are actually predicted at the output of the transformer block. Training objective comprises optimization over loss functions of the transformer block as well as the quantization block.

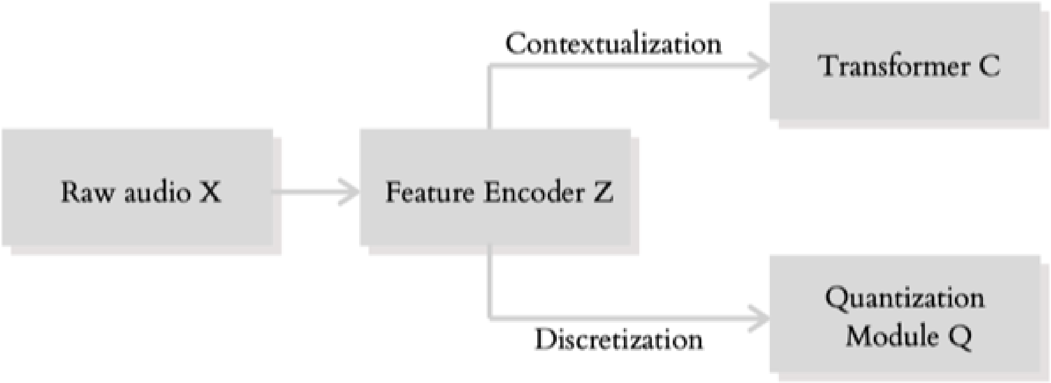

Figure 5: Block diagram of wav2vec 2.0 model.

3.3 Representations from Predictive Coding

Based on cognitive science studies, it is known that humans learn new categories of sounds and images by observing and then making predictive observations. A human listener that hears the first part of a sound can more or less intuitively predict the future sequences depending on the context. This is exactly the same motivation behind Predictive Coding (PC) models [21].

Two types of PC models have been proposed under self-supervised learning for speech representations. One is referred as Autoregressive Predictive Coding (APC) model and the other as Contrastive Predictive Coding (CPC) model [22, 4]. The effective performance of representations obtained from these models, on speaker verification and phoneme recognition tasks, motivates us to experiment with them. Again, since the model training and architecture themselves are quite complicated, we present a brief description of the self-supervised learning approach adopted in these models.

An APC model is similar to an RNN based language model for text. Given the context of a sequence of words in the history, represented by the hidden state (h_{t-1}) of the RNN, the RNN language model tries to predict the word at the next time step ($\hat{x_t}$). However, representations extracted from a speech signal are highly continuous in nature and hence can be easily predicted by an RNN model. To complicate the prediction task, and to result into more effective representations, APC models try to predict a representation which is n time steps ahead ($\hat{x}_{t+n}$). In contrast to this training objective of the APC model, a CPC model aims to distinctly identify the target representation at time step n from randomly sampled imposter representations, using a noise contrastive loss function.

Chapter 4

Methodology and Action Plan

4.1 Planned Study

The main objective of this project is to achieve a more efficient blind phoneme segmentation model. In doing so we will adopt the blind phoneme segmentation method discussed in section 3.1. In particular we will adopt the RNN-MFCC model represented by equation (6) in section 3.1. As compared to the original work, which uses MFCC feature vectors as representations for the speech signal, we would like to study the performance with effective representations learned from self-supervised learning methods. We will experiment with representations obtained from the wav2vec model discussed in section 3.2 and autoregressive predictive coding models and contrastive predictive coding models discussed in section 3.3. These representations will be extracted from the pre-trained wav2vec[1] and predictive coding[2] models made available by the authors of the respective models.

4.2 Evaluation

The metrics for evaluation of blind phoneme segmentation from [19] will be used in our study. This includes:

- Recall: the fraction of actual boundaries in the reference transcription that are correctly identified by a method.
- Precision: the fraction of total boundaries identified by a method that are correct.

[1] `https://github.com/pytorch/fairseq/tree/master/examples/wav2vec`

[2] `https://github.com/iamyuanchung/Autoregressive-Predictive-Coding`

- F1-score: the harmonic mean of Recall and Precision. Computed as:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

- R-val: a metric designed to handle over-segmentation. Computed as:

$$R = 1 - \frac{\sqrt{(1 - \text{Recall})^2 + OS^2} + \frac{|\text{Recall} - OS - 1|}{\sqrt{2}}}{2} \tag{2}$$

  where over-segmentation parameter $OS = \text{Recall}/\text{Precision} - 1$

4.3 Corpus and Tools

Experiments will be carried out on the publicly available Librispeech [23] corpus which contains English read speech. Since some of the pre-trained models used in our work are already trained on Librispeech dataset, special care will be taken so that part of Librispeech already seen during representation learning does not overlap with the train, validation or test sets used in our study.

ScikitLearn[3] Python library [24] will be used for clustering and dimensionality reduction tasks. PyTorch[4] Python library [25] will be used for training recurrent neural network for the blind phoneme segmentation task.

4.4 Timeline

The project is planned to progress through the following stages.

- Month 1 (Jan 2021)
  - Introduction to tools and corpora discussed in sections 4.3 and 4.3.
- Month 2-3 (Feb - Mar 2021)
  - Implementation of blind phoneme segmentation on MFCC features.
- Month 4-5 (Apr - May 2021)
  - Exploration of representations from pre-trained wav2vec and predictive coding models.
  - Compilation of results, final report and defense preparations.

[3] `https://scikit-learn.org/stable/index.html`

[4] `https://pytorch.org`

Bibliography

[1] D. B. Roe and J. G. Wilpon. Whither speech recognition: the next 25 years. *IEEE Communications Magazine*, 31(11):54–62, 1993.

[2] M. Paul Lewis, editor. *Ethnologue: Languages of the World.* SIL International, Dallas, TX, USA, twenty-third edition, 2020.

[3] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469, 2019.

[4] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. An Unsupervised Autoregressive Model for Speech Representation Learning. In *Proc. Interspeech 2019*, pages 146–150, 2019.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, June 2019.

[6] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4182–4192, 13–18 Jul 2020.

[7] I. Misra and L. van der Maaten. Self-supervised learning of pretext-invariant representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6706–6716, 2020.

[8] Romain Serizel, Victor Bisot, Slim Essid, and Gaël Richard. *Acoustic Features for Environmental Sound Analysis*, pages 71–101. Springer International Publishing, Cham, 2018.

[9] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[10] Gregory Gelly. *Reseaux de neurones recurrents pour le traitement automatique de la parole*. Theses, Université Paris-Saclay, September 2017.

[11] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Dec 2020.

[12] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks. In *Proc. Interspeech 2019*, pages 161–165, 2019.

[13] Zheng Lian, Jianhua Tao, Bin Liu, and Jian Huang. Unsupervised Representation Learning with Future Observation Prediction for Speech Emotion Recognition. In *Proc. Interspeech 2019*, pages 3840–3844, 2019.

[14] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, June 2018.

[15] Anna Esposito and Guido Aversano. Text independent methods for speech segmentation. In Gérard Chollet, Anna Esposito, Marcos Faundez-Zanuy, and Maria Marinaro, editors, *Nonlinear Speech Modeling and Applications*, pages 261–290, 2005.

[16] Y. P. Estevan, V. Wan, and O. Scharenborg. Finding maximum margin segments in speech. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, pages IV–937–IV–940, 2007.

[17] George Almpanidis and Constantine Kotropoulos. Phonemic segmentation using the generalised gamma distribution and small sample bayesian information criterion. *Speech Communication*, 50(1):38 – 55, 2008.

[18] Dac-Thang Hoang and Hsiao-Chuan Wang. Blind phone segmentation based on spectral change detection using legendre polynomial approximation. *The Journal of the Acoustical Society of America*, 137(2):797–805, 2015.

[19] Paul Michel, Okko Rasanen, Roland Thiollière, and Emmanuel Dupoux. Blind phoneme segmentation with temporal prediction errors. In *Proceedings of ACL 2017, Student Research Workshop*, pages 62–68, July 2017.

[20] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 1027–1035, USA, 2007.

[21] María Andrea Cruz Blandón and Okko Räsänen. Analysis of predictive coding models for phonemic representation learning in small datasets. *CoRR*, abs/2007.04205, 2020.

[22] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[24] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830, November 2011.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.