Improving unsupervised acoustic word embeddings using segment- and frame-level information

Lisa van Staden

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University.

Supervisor: Dr Herman Kamper
Department of Electrical and Electronic Engineering

December 2021

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	Handtekening / *Signature*
L. van Staden Voorletters en van / *Initials and surname*	December 2021 Datum / *Date*

Abstract

English

Many speech processing tasks involve measuring the acoustic similarity between speech segments. Conventionally, these speech comparisons are performed using dynamic time warping (DTW), a computationally expensive alignment-based approach. Recent research has shown that fixed dimensional vectors, which are representations for speech segments of variable length, can be used in these tasks. These vectors, called *acoustic word embeddings* (AWEs), allow for efficient comparisons. A number of studies have shown that AWEs can be used in tasks such as unsupervised term discovery (UTD) and query-by-example-search in a zero-resource setting, where transcriptions for speech are not available and full speech recognition is therefore not possible. Therefore, some studies have focussed on developing unsupervised AWEs methods in this setting. However, the intrinsic quality of supervised AWEs is still vastly superior compared to unsupervised AWEs. This serves as motivation to investigate methods to improve the quality of unsupervised AWEs. Additionally, this is also of interest to the language acquisition field, considering that infants do not require transcriptions to learn speech.

We focus on three different problem areas present in current AWEs. Firstly, we consider the nuisance factors in AWEs. The acoustic properties of different speakers and genders vary dramatically and in an unsupervised environment these properties, which we call nuisance factors, can still be captured to a large extent. This is addressed by applying speaker and gender conditioning and adversarial training to existing AWEs models, the autoencoder recurrent neural network (AE-RNN) and correspondence autoencoder recurrent neural network (CAE-RNN). We find that these methods reduce some speaker and gender information and marginally improve the AWEs.

Secondly, we consider if improvements at the frame-level will have a positive effect on the quality of the AWEs. Many AWE studies have focussed on the word-level, but a few other zero-resource studies have instead focussed on developing short-time frame-level speech representations that capture meaningful contrasts such as phonemes. These contrasts are more relevant at a shorter time scale than most AWEs approaches, that focus on discriminative words. Three existing representation types are considered: contrastive predictive coding (CPC), autoregressive predictive coding (APC) and the correspondence autoencoder (CAE). These are used as input features to the CAE-RNN and compared to using conventional mel-frequency cepstral coefficients (MFCCs). Additionally, we introduce a fourth learned representation method: correspondence autoregressive predictive

coding (CAPC), that combines the mechanisms of the frame-level CAE and APC models. We find that better input features have a significant impact on the quality of the AWEs with the best results from using the CPC features.

The last problem we consider is the training strategy used for AWE models. Motivated by the idea that human infants are first exposed to speech from only a small number of speakers which gradually increases, we apply a speaker number-based curriculum learning strategy to the AE-RNN and CAE-RNN and compare it to using a multiple speaker strategy. We find that this training strategy does not make a difference to the quality of the AWEs.

Taken together, in our experiments we find that the most impactful solution is to use learned frame-level representations as input. Speaker and gender normalising has a marginally positive effect on the quality of the AWEs and the training strategy has no impact. Going forward, these improved AWEs can be used in downstream tasks. Although we only considered AWEs from the AE-RNN and CAE-RNN, the problems we focussed on are not necessarily model-specific and our findings are relevant to other AWE modelling research.

Afrikaans

Baie spraakprosesseringstake behels dat die akoestiese ooreenkoms tussen spraaksegmente gemeet word. Konvensioneel, word hierdie spraakvergelykings uitgevoer met behulp van dinamies tyd-buiging, 'n raampie-ooreenstemgebasseerde metode wat berekenings gewys duur is. Onlangse navorsing toon dat vaste dimensionele vektore, wat voorstellings vir spraaksegmente van verskilende lengtes is, in hierdie take gebruik kan word. Hierdie vektore, wat *akoestiese woordinbeddings* (AWI) genoem word, maak dit moontlik om doeltreffend spraakvergelykings uit te voer. 'n Aantal studies het al gewys dat AWIs gebruik kan word in take soos toesiglose term-ontdekking en navraag-na-voorbeeld soek in 'n nul-hulpbron-spraakinstelling, waar transkripsies vir spraak nie beskikbaar is nie. Daarom is daar in sommige studies gefokus op die ontwikkeling van toesiglose AWI-modelering metodes in hierdie instelling. Die intrinsieke kwaliteit van AWIs onder toesig is egter steeds ver hoër in vergelyking met toesiglose AWIs. Dit dien as motivering om metodes te ondersoek wat die kwaliteit van toesiglose AWIs kan verbeter. Verder, is dit ook van belang vir die taalverwerwingsveld, aangesien babas nie transkripsies benodig om spraak aan te leer nie.

Ons fokus op drie verskillende probleemareas wat in huidige AWIs voorkom. Eerstens beskou ons die oorlasfaktore in AWIs. Die akoestiese eienskappe van verskillende sprekers en geslagte wissel dramaties en in 'n toesiglose instelling kan hierdie eienskappe, wat ons na verwys as oorlasfaktore, nog tot in 'n groot mate vasgevang word in die AWIs. Ons spreek dit aan deur spreker- en geslagsvoorwaardelikheid, en teenstrydige opleiding op bestaande AWI-modelle, die outoenkodeerder herhalende neurale netwerk (OE-HNN) en korrespondensie outoenkodeerder herhalende neurale netwerk (KOE-HNN), toe te pas. Ons vind dat hierdie metodes van die spreker- en geslagsinligting verminder en dat dit die kwaliteit van die AWIs effens verbeter.

Tweedens kyk ons of verbeterings op raamvlak 'n positiewe uitwerking op die kwaliteit van die AWIs sal hê. Baie AWI-studies he al gefokus op die segmentvlak, maar 'n paar ander nul-hulpbronstudies het eerder gefokus op die ontwikkeling van kort-tydperk spraak-voorstellings op die raamvlak, wat betekenisvolle kontraste, soos foneme, kan opvang. Ons oorweeg drie verskillende bestaande voorstellingtipes: kontrasterende voorpsellende kodering (KVK), outoregresiewe voorspellende kodering (OVK) en korrespondensie outoenkodeerder (KOE). Hierdie word as invoerkernmerkvektore vir die KOE-HNN gebruik en ons vergelyk dit met die gebruik van die konvensionele Mel-frekewensie kepstrale koëffisiënte. Ons stel ook 'n vierde metode vir geleerde voorstellings voor: korrespondensie outoregresiewe voorspellende kodering, wat die meganismes van die raamvlak KOE- en OVK-modelle kombineer. Ons vind dat hierdie beter invoerkernmerkvektore 'n groot impak op die kwaliteit van die AWIs het waar die beste resultate van die gebruik van KVK invoerkernmerkvektore is.

Die laaste probleem wat ons oorweeg, is die opleidingstrategie wat vir AWI-modelle gebruik word. Gemotiveer deur die idee dat babas aanvanklik aan spraak blootgestel word van slegs 'n klein getal sprekers, pas ons 'n sprekergetalgebaseerde kurrikulumleerstrategie toe op die OE-HNN en KOE-HNN en vergelyk dit met die gebruik van 'n meervoudige sprekerstrategie. Ons vind dat hierdie opleidingstrategie nie 'n verskil maak aan die kwaliteit van die AWIs nie.

Alles saamgevat, vind ons dat die mees effektiewe oplossing is om raamvlak geleerde voorstellings as invoerkenmerkvektore te gebruik. Normalisering van spreker- en geslagingligting in AWIs het 'n effens positiewe impak op die kwaliteit daarvan en die verskil in opleidingstrategie het geen impak nie. Hierdie verbeterde AWIs kan vorentoe gebruik word in stroomaf take. Alhoewel ons slegs AWIs van die OE-HNN en KOE-HNN oorweeg het, is die probleme waarop ons gefokus het nie noodwendig modelspesifiek nie. Daarom is ons vonds relevant vir ander AWI-modellering navorsing.

Acknowledgements

For the past little more than two years there has been a number of people whose support has made this thesis possible.

First, I would like to thank my supervisor, Herman Kamper, whose guidance has been a great deal more than should be expected. Thank you for pushing me to present and write papers and for always being so kind. My Masters was made possible by the funding that I received from the Investec Scholarship programme for which I am grateful. To my parents, thank you for believing in me as much as you do, I would not have been where I am today without all your support. Thank you to all my friends from the Media Lab for contributing to such a great work environment and for providing much needed social interactions (before lockdown) and a special thank you to the few that were there for the final stretches: Burger, Jason and Kevin, all your encouragement definitely helped keep my anxieties at bay. To my other friends and family, thank you for the positive impact on my general happiness. Lastly, to my favourite person, Jacques, thank you for all your love and encouragement.

Contents

List of Figures

List of Tables

Nomenclature

Variables and functions

$\boldsymbol{c}$	A context vector.
$\boldsymbol{x}$	A frame from sequence X, used as input to a model.
$\boldsymbol{y}$	A frame from sequence Y.
$\boldsymbol{z}$	A latent variable.
γ	weight hyperparameter
$\mathcal{Z}$	A set of latent variables.
μ	mean
σ	standard deviation
A	A sequence sliced from X.
a	A frame anchor position.
f	A function that maps to a score value.
g	A function to be optimised.
k	A number of time-steps.
L	A loss function.
M	The number of frames in a set.
N	The number of utterances in a set.
p	A value indicating probability.
s	A number of time-steps used in auxiliary functions.
T	The length of a sequence.
t	A time-step position.
X	A sequence of frames of an utterance, used as input to a model.
Y	A sequence of frames of an utterance, used in a pair with X.

Acronyms and abbreviations

AE autoencoder

AE-RNN autoencoder recurrent neural network

AP average precision

APC autoregressive predictive coding

AWE acoustic word embeddings

CAE correspondence autoencoder

CAE-RNN correspondence autoencoder recurrent neural network

CAPC correspondence autoregressive predictive coding

CNN convolutional neural network

CPC contrastive predictive coding

DNN deep neural network

DTW dynamic time warping

FNN feedforward neural network

FMLLR feature-space MLLR

GAN generative adversarial network

GMM gaussian mixture model

GRU gated recurrent unit

HMM hidden Markov model

LDA linear discriminant analysis

LPC linear predictive coding

MAE mean absolute error

MAP maximum a posteriori

MFCC Mel-frequency cepstral coefficient

MLLR maximum likelihood linear regression

MSE mean square error

NCE noise-contrastive estimation

PCA principal component analysis

RAILS randomized acoustic indexing and logarithmic-time search

RNN recurrent neural network

.

TDNN time-delay neural network

UBM universal background model

UTD unsupervised term discovery

VTLN vocal tract length normalisation

Chapter 1

Introduction

A number of speech processing tasks rely on measuring the acoustic similarity between speech segments [1–7]. Usually, similarity is measured using dynamic time warping (DTW), an algorithm that finds an optimal alignment between speech segments [8]. However, DTW is computationally expensive. This has led to research in methods of finding fixed-dimensional speech representations, referred to as *acoustic word embeddings (AWEs)* [9–15]. These methods attempt to capture the acoustic information in speech segments of variable length and condense it in such a way that segments containing the same words are mapped to similar embeddings. Since speech segments can then be represented in the same fixed-dimensional space, measuring the acoustic similarity can be done with a computationally inexpensive distance calculation.

Only a few languages have the large amount of resources required for full speech recognition tasks, yet there are 7 139 known languages in the world [16]. Gathering labelled data for lower resourced language is expensive and not all languages even have writing systems. Fortunately, many of the downstream tasks for which AWEs are useful can be performed in a setting where transcribed speech resources are unavailable. Such tasks include query-by-example search [1, 3, 4, 17], where a speech segment is used as a query to search over a database of speech, and full speech segmentation [5–7], where the aim is to discover the boundaries for spoken terms in a collection of spoken utterances. Speech processing in settings without any labelled speech data is referred to as *zero-resource speech processing* and with the introduction of the *ZeroSpeech Challeges* it has become a popular field of research [18–21]. A number of studies have specifically focussed on developing AWEs for this zero-resource setting [9, 11, 13, 22].

However supervised AWE methods (where word labels are available during training) still greatly outperform unsupervised AWE methods [13, 22, 23]. Some studies have considered finding better quality zero-resource language AWEs by using models that have been trained in a supervised fashion on higher resourced languages [24–26]. Although these multilingual studies have been successful, it is still of interest to research unsupervised AWE methods as it relates closely to language acquisition – human infants acquire language without access to transcribed speech data [27–29].

We consider a variety of problems in unsupervised AWEs that can be addressed at the frame-level (short-time intervals in speech), segment-level (words or phrases) and

batch-level (groups of words or phrases) by specifically focusing on the AWEs produced by an autoencoder recurrent neural network (AE-RNN) [11] and those of a correspondence autoencoder recurrent neural network (CAE-RNN) [13] model. The AE-RNN was the first encoder-decoder model used as an AWE approach. Here, an input speech segment is mapped to a latent variable and the model is trained to reconstruct the latent variable into the input segment. These latent variables can then be used as AWEs. The CAE-RNN has the same architecture as the AE-RNN, but the input-output pairs differ. Here, a pair of speech segments that are predicted to be similar by a unsupervised term discovery (UTD) system are used such that one segment is mapped to the latent variable and reconstructed into the other. In a word discrimination task the AWEs of the CAE-RNN performed comparable or slightly better compared to a DTW approach, indicating that the CAE-RNN model is one of the current best unsupervised AWE approaches [13]. The aforementioned word discrimination task, called the *same-diff* task [30], is also what we use to measure the intrinsic quality of our AWEs. The specific problems surrounding unsupervised AWEs that we consider are described below.

The acoustic properties of speech across different speakers and between adult men and women vary dramatically. This includes properties like pitch, timbre and pronunciation [31, 32]. In the unsupervised setting, where there is no word labels, these properties could still be captured in the AWEs. This can lead to a scenario where AWEs for different words from the same speaker can be more similar than AWEs representing the same word from different speakers, and similarly for AWEs between men and women. In Chapter 3 we confirm that unsupervised AWEs contain significant speaker and gender information by showing that a classifier can predict the correct speaker identities or gender from AWEs with high accuracy. This problem is addressed with two different methods: we apply speaker or gender conditioning to the decoder component of the AE-RNN or CAE-RNN and we adversarially train both models against a speaker or gender classifier. We find that both of these methods reduce some of the speaker and gender information and result in marginal improvement in the AWE performance.

Most unsupervised AWE approaches have only focused on the segment-level, where they aim to encourage models to produce AWEs that are discriminative between words. For example, the CAE-RNN uses top-down information presented by corresponding speech segments in UTD pairs to form AWEs that capture segment-level relevant linguistic information. However, also in the zero-resource speech field, there have been other studies that focussed on unsupervised representation learning at the short-time frame-level [33–38]. In these studies they aim to learn frame-level speech representations that discriminates well at a shorter time interval than words, like phonetic categories. Training deep neural network (DNN) models on the audio waveform directly is computationally expensive and most AWEs studies train models on feature engineered frame-level representations, like MFCCs [9,11,13]. It is worth investigating if the quality AWEs can be improved if improved

features are used as input to AWE models. We implement three existing types of frame-level representations: the frame-level correspondence autoencoder (CAE) [37], contrastive predictive coding (CPC) [39] and autoregressive predictive coding (APC) [38, 40]. We also introduce a fourth type that combines mechanisms from the frame-level CAE and APC model and we call it CAPC. The frame-level CAE uses top-down information presented in UTD pairs, like the CAE-RNN, but here the two segments in a pair are further aligned so that each frame the one speech segment has a corresponding frame in the other. This aim is to learn frame-level representations that only capture information that is present in corresponding frames and therefore disregarding noise or speaker information. In both the CPC and APC approaches, representations are learned by training the models to predict future frames from previous frames. Here, the aim is for the frame-level representations to capture the information shared between frames over short intervals. This usually includes higher level information like phonemes and disregards noise. In our experiments, in Chapter 4, we find that using these learned representations as input to the CAE-RNN result in AWEs that significantly outperform those from the CAE-RNN trained on MFCCs. Additionally, in probing tasks we show that the different learned representations capture different information.

Most unsupervised AWE approaches train models from a fixed training set with multiple speakers. Consider the order in which human infants are exposed to speech input: initially, we are only exposed to the speech from a limited number of people and this pool of people gradually grows as we get older. Some studies have shown that the strategy used to feed input into a DNN model can have an impact in terms of the order of difficulty that the input is presented in [41–47]. Keeping in mind both of these ideas, we suggest a curriculum learning training strategy for AWE models where the order of difficulty is determined by the variance in speakers. In Chapter 5 this training strategy is compared to using the conventional multiple speaker strategy and we find that, unfortunately, there is almost no difference in the quality of AWEs produced by models trained on the two different strategies.

In summary, we investigate various approaches to improve the quality of unsupervised AWEs from the CAE-RNN in Chapters 3, 4 and 5 and also from the AE-RNN in Chapters 3 and 5. The various approaches address the AWE models at different levels. Using learned representations as input, addresses the frame-level. By using higher level speaker and gender information for augmenting the models for speaker and gender conditioning or changing the loss functions with adversarial training, we address the segment-level. In our speaker number based curriculum learning approach, we focus on speaker variance, which is also batch-level information. We find that the best improvement in AWE is from using learned representations as input to the CAE-RNN. Normalising out speaker and gender information leads to marginal improvement in AWEs and using a curriculum learning training strategy does not lead to improvement.

1.1. Thesis outline

The remainder of the thesis is organised as follows:

Chapter 2: We discuss background that relates to unsupervised AWEs in general, the downstream tasks for which they are useful and previous unsupervised AWE approaches. Other information relevant to the chapters that follow is also discussed: the details of the AE-RNN and CAE-RNN that will be used, the same-diff task evaluation method and the English, Xistonga and Hausa datasets.

Chapter 3: This chapter focusses on normalising out speaker and gender information in AWEs. First, other work related to speaker and gender normalising in acoustic modelling is briefly discussed. This is followed by the discussion of the methods we will employ for normalising out speaker and gender information: speaker and gender conditioning and adversarial training. Next is our experimental setup for these methods. Then we discuss development experiment results to address some of the questions relating to the methods used and we report the test results of the best methods on English and Xitsonga data. The chapter is concluded with a summary of our findings.

Chapter 4: This chapter focusses on implementing and using learned representations as input to the CAE-RNN. We briefly discuss unsupervised representation learning methods that has been presented in other studies, specifically representations that use top-down information and predictive coding approaches. There are four learned representations that we consider and we discuss their details and how our experiments will be set up: the frame-level CAE, CPC, APC and CAPC. We report the test results of using the four learned representations as input to an AWE model and compare it to using MFCCs. Different probing tasks are set up to further analyse the different learned representations and the different effects they have on produced AWEs. The chapter ends with a short conclusion section.

Chapter 5: This chapter focusses on speaker number-based training strategies. First, we discuss other studies that have incorporated a curriculum learning training strategy for their models as well as work that is related to how the number of speakers impact speech perception in humans. The details of the different training strategies that we will compare are discussed: a multiple speaker, single speaker, curriculum learning and reverse curriculum learning strategies. How these different strategies will be set up is discussed next. Different combinations of the strategies are used on the subsequently trained AE-RNN and CAE-RNN and we report the development investigation results followed by the test results on Hausa data for the best combinations. We also report as short analysis on the different training

strategies by considering the average precision (AP) score progression through a range of epochs. We conclude with a summary of our findings.

Chapter 6: In the final chapter we summarise and discuss the different approaches we followed in an attempt to improve unsupervised AWEs and what our findings were. This is followed by what we think should be focussed on in future work.

1.2. Contributions

In this thesis we use ideas and methods that have been presented in other areas of deep learning or acoustic modelling and we apply them to unsupervised AWE models. Even though we only consider the AE-RNN and CAE-RNN, we are the first to apply these methods to any AWE model. More details are listed below.

- We are the first to apply speaker or gender normalising to AWE models directly, using speaker or gender labels and to the best of our knowledge, we are the first to apply direct gender normalising to an acoustic model.
- We are the first to use learned representations as input to AWE models. Previous studies have mostly used feature engineered frame-level speech representations like MFCCs, perceptual linear prediction or frequency-domain linear prediction features [9, 13, 48].
- The CAPC model of Chapter 4 is a novel frame-level representation learning model that combines mechanisms of the frame-level CAE model [37] and APC. We show that CAPC input features lead to improved results compared to both those of the CAE model and APC on the English data, indicating that the combination of the two is complementary.
- We are the first to apply a curriculum learning training strategy to AWE models and also to the best of our knowledge, the first to introduce a curriculum learning strategy based on the number of speakers. Previous speech processing studies have consider curriculum learning training strategies based on levels of noise [43, 47] and level of emotion ambiguity [45].

1.3. Publications and code

The majority of Chapter 3 is based on a paper presented in *Proceedings of the Annual Symposium of the Pattern Recognition of South Africa (PRASA)* (2020) [49] and the majority of Chapter 4 is based on a paper presented at the *IEEE Spoken Language Technology Workshop (SLT)* 2021 [50]. The repository containing all the code used for the

experiments in this thesis can be found at:

https://bitbucket.org/leesah20/acousticwordembeddings/src/master/

Chapter 2

Background

In the chapters that are to follow we will discuss the related work relevant to the approach focused on in each individual chapter. But in this chapter we focus on work related to unsupervised AWEs in general. There have been a number of different modelling approaches used for unsupervised AWE and we highlight some of these studies in Section 2.1. One of our main motivations for AWEs is that they can be used in downstream tasks instead of DTW methods for more efficient computation. The work related to two different downstream tasks for which AWEs have been used is highlighted in Section 2.2. Further we discuss the details of the information that is relevant to the chapters that will follow: the AE-RNN and CAE-RNN in Section 2.3, the word discrimination task we use for evaluation in Section 2.4 and the English, Xitsonga and Hausa datasets that will be used for our experiments in the chapters to follow in Section 2.5.

2.1. Modelling unsupervised acoustic word embeddings

Here we will highlight studies that have focused on modelling unsupervised AWEs, starting with the early approaches and then those that used DNN models.

Levin et al. [9] was the first to introduce AWEs. They compared various unsupervised and supervised approaches for extracting AWEs within a setting where only a few hours of speech are available. The first approach they considered is downsampling, a method that does not require any training data or labels. Here, a chosen number of frames is sampled from a speech segment and concatenated together to form an AWE. The frames can be sampled uniformly or with a non-uniform method (like in this study where they use an hidden Markov model (HMM) to divide the segment into regions of which the averages are then concatenated). Downsampling has also been used for AWE in a speech segmentation task (Section 2.2.2) and as a baseline AWE method in other studies [13, 15]. We also use this approach as a baseline in Chapter 4.

The other unsupervised approaches that the study considers are reference vectors and Laplacian eigenmaps. Reference vectors are a type of Lipschitz embedding [51, 52]. Here the DTW costs between a speech segment and each segment in a reference set is determined and concatenated together to form the reference vector. The reference set

must contain enough speech segment for it to form an adequate basis of all possible speech segments, this can result in the reference vectors being very large. Linear dimensionality reduction methods can be used to mitigate this, the unsupervised reduction method that they use is principal component analysis (PCA). They found that the dimensionality can be reduced significantly with this method with only a slight decrease in performance.

The only unsupervised AWE extraction approach that they found matched the performance of DTW is by using Laplacian eigenmaps [53]. Here, a graph is created where the vertices are the speech segments of the training data. A vertex has is connected to all other vertices that are one of its k nearest neighbours, determined by the DTW cost. A set containing chosen number of projection maps is then used to map a speech segment to a set of values, which can then be used as the AWE. The projections maps are set up so that training speech segments that are close together in the graph will be mapped to embeddings that are close together. This type of AWE has been used in query-by example search (Section 2.2.1) and speech segmentation (Section 2.2.2) tasks.

Although some of these unsupervised AWEs have proven useful [1, 5–7], they found that all their supervised approaches, where they had access to the word label for each training segment, significantly outperforms all unsupervised approaches and DTW in a word discrimination task. Others have since focussed on producing higher quality AWEs by using DNN models.

Chung et al. [11] was the first to propose a DNN model for learning AWEs. Their model is the autoencoder recurrent neural network (AE-RNN) and based off the encoder-decoder recurrent neural network (RNN) of [54]. This model consists of an encoder and decoder which each comprises a stack of RNNs. The encoder maps an input sequence of features to a fixed dimensional latent variable. The decoder then reconstructs this latent variable to match the input sequence. The latent variable is then used as the AWE.

Kamper et al. [12] and Settle and Livescu [48] both used Siamese networks to train AWE from speech segments that are known to be similar or different. In [12] a convolutional neural network (CNN) architecture was used to train two speech segments at a time on the same model with the same parameters. The loss was minimised or maximised, depending on if the segments were known to be similar or not, respectively. In [48] an RNN architecture was used and the model was trained on three segments at a time, where one is chosen to be the anchor and of the other two, one is chosen to be similar to the anchor and the other one to be different. A triplet loss function is then used on the three AWEs. UTD can be used here to find known similar segments, but in this case the results are poor [22].

Kamper [13] proposed a correspondence autoencoder recurrent neural network (CAE-RNN) model that is similar to the AE-RNN with the exception that the model is not trained to reconstruct the input segment but rather a different segment that is known to be similar to the input. The input-output pair of similar pairs can be obtained in

an unsupervised fashion by using a UTD system [55] (Section 2.2.2). In the same study they also consider a variational autoencoder (AE) for learning AWEs, but the AWEs of the CAE-RNN were superior. Their CAE-RNN trained on unsupervised input-output pairs was shown to give comparable or slightly better performance compared to a DTW approach [13], making it one of the best unsupervised AWE models. More details on the AE-RNN and CAE-RNN are discussed in the next section (Section 2.3).

Some studies have considered improving the quality of unsupervised AWEs produced by the CAE-RNN by training the model in a supervised fashion and then using it to encode zero-resource language AWEs [25] or modifying the architecture by incorporating variational AE elements [22].

2.2. Downstream tasks for unsupervised AWEs

Unsupervised AWEs are useful in downstream tasks where the tasks involve finding the acoustic similarity between speech segments. We highlight two of these tasks that can be used in low resource settings, query-by-example search on speech in Section 2.2.1 and unsupervised spoken term segmentation in Section 2.2.2.

2.2.1. Query-by-example search

Query-by-example search on speech is the task of searching over a collection of speech by finding matches for a spoken query [56–58]. This task can be particularly useful for zero-resource setting applications as it does not necessarily require large vocabulary speech data nor transcriptions [57, 59].

Early methods for query-by-example search include HMM and DTW approaches. The HMM approaches include using discrete HMMs to model phonetic confusion networks to represent the query [60] or using the state sequence of an ergodic HMMs to represent the queries [61]. Studies that followed DTW approaches were found to produce better results, like in Hazen et al. [58] where they used an independently trained phoneme classifier to convert the frames in the spoken query and speech collection into sequences of phonetic posteriorgrams to which DTW is then applied to find the matches. Zhang and Glass [57] followed this approach, but they avoided using a pretrained phonetic classifier by training a gaussian mixture model (GMM) on the input speech without transcription and then using it to generate Gaussian phonetic posteriorgrams. The DTW algorithm is computationally expensive (it runs in polynomial time) and some studies have considered less expensive DTW approximated alignments for query matches [62–64]. One of these methods,randomized acoustic indexing and logarithmic-time search (RAILS), was introduced by Jansen and Durme [63]. Here, frames are first hashed to bit signatures and when a query is presented an approximate nearest neighbours algorithm is used to

retrieve likely frame-level matches. The frame-level matches are used to form similarity matrices on which image processing techniques are used to find segment level matches. This method reduces computational complexity to logarithmic time.

Besides being computationally expensive, another limitation to DTW approaches is that longer phoneme sounds are weighted more, which must sometimes be accounted for [57]. Levin et al. [17] introduced using AWEs for query-by-example search. Here they used a similar system to [63] called segmental RAILS. However, here the queries and speech collection were converted to laplacian eigenmap AWEs which then allows the frame-level processing to be skipped and the AWEs can be hashed directly. In their experiments they found that this system achieves considerable better accuracy results and the runtime is greatly decreased compared to [63].

Settle et al. [1] used segmental RAILS for query-by-example search as well, but they used DNN produced AWEs from the Siamese RNN model in [48] (which does require a small amount of word-level transcriptions). They found that the DNN produced AWEs result in large improvements in accuracy, compared to using the laplacian eigenmap AWEs like in [17]. Other studies have since also had success with query-by-example search using AWEs [3, 4].

2.2.2. Unsupervised spoken term discovery and segmentation

Park and Glass [2] were the first to introduce unsupervised spoken term discovery. The aim of this task is to find repeating patterns in speech and cluster matching terms together [2, 55, 65]. In [2] a segmental DTW algorithm was used to find matching subsequences between different speech utterances and an adjacency graph was then used to cluster the terms. Jansen and Van Durme [55] presented a less computationally expensive spoken term discovery system that runs in linear linearithmic time where as DTW runs in quadratic time. They used an approximate nearest neighbours algorithm with randomised projections to find a similarity matrices to which a search and retrieval algorithm is then applied to. This system has been used to discover similar speech segment pairs that are used as input and output to zero-resource speech models [37, 66] including AWE models [13].

Unsupervised speech segmentation is another task that aims to find the boundaries for spoken terms, but instead of isolating only discovered repeated terms, all speech input is segmented. Some studies have focussed on modelling words from subword units, jointly [67] or in a hierarchical fashion where subword units are first discovered and then words are modelled on top of these units [68, 69]. One of these studies, Walter et al. [69], used unsupervised speech segmentation for digit recognition. Here, they presented a two-stage hierarchical system. In the first stage, subword units are discovered by segmenting these units in the input, clustering them based of DTW distances and modelling each unit type on a HMM. In the second stage, HMMs of repeating subword sequences are fed into a

word-level HMM. There is a word-level HMM for each digit, so that the output of each is a pronunciation dictionary.

Kamper et al. [7] introduced a system with the same digit recognition goal, but instead of modelling on subword units, they modelled on AWEs. Here, the boundaries for segments are iteratively hypothesised, starting with random boundaries. First, each segment is mapped to a laplacian eigenmap AWE. Then the AWEs are modelled on a bayesian GMM that cluster similar word types together so that each component represents one word type. Next, the likelihood scores are calculated for all the AWEs which is used to determine a segmentation score. Then the Gibbs sampling algorithm is used to sample the likely segmentation boundaries based off the GMM probabilities. They found that this approach outperforms the one followed by [69]. This approach has also been extended to accommodate a large vocabulary where downsampled AWE [6] were used and made more efficient by using k-means clustering instead of a GMM [5].

2.3. The autoencoder recurrent neural network and correspondence autoencoder recurrent neural network

In the following chapters we try to improve the AWEs from encoder-decoder RNN models, specifically the AE-RNN and CAE-RNN. The details of these two models are discussed below.

The architecture for the two models are the same as shown in Figure 2.1. In both models, the encoder and decoder each consists of a stack of RNN layers. The encoder

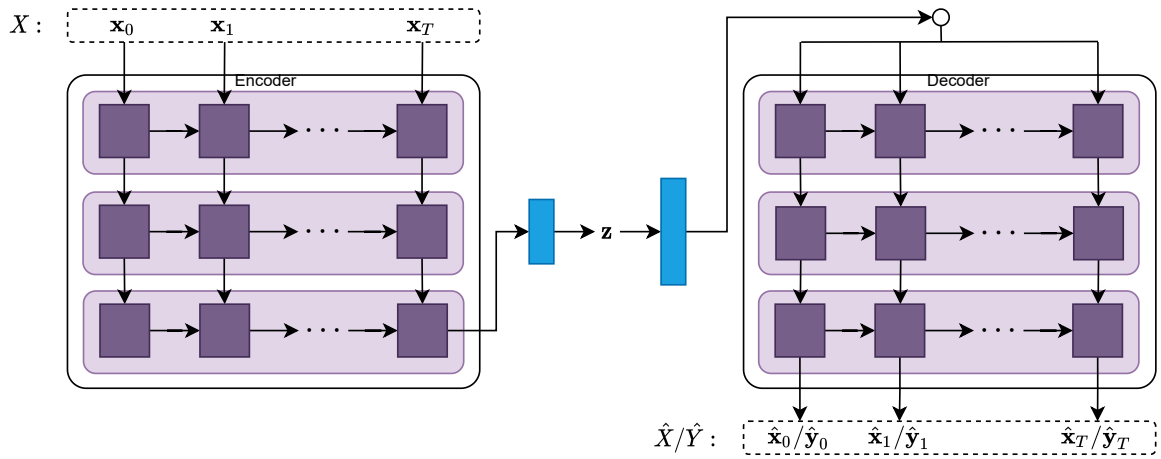

Figure 2.1: The CAE-RNN's weights are initialised from those of an AE-RNN. The AE-RNN is trained to reconstruct the input sequence $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ from a latent variable $\boldsymbol{z}$. The CAE-RNN is trained to reconstruct a different sequence $Y = (\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_T)$ predicted to contain the same word as the input X.

maps an input sequence X of variable length into a fixed-dimensional latent variable $\boldsymbol{z}$. This latent variable could be the last hidden state of the last encoder RNN layer, but in our case we add a linear layer after the encoder to transform the last hidden state into $\boldsymbol{z}$. We use these latent variables $\boldsymbol{z}$ as acoustic embeddings. The decoder then maps $\boldsymbol{z}$ to an output sequence, denoted by $\hat{X}$ for the AE-RNN and $\hat{Y}$ for the CAE-RNN.

The AE-RNN is trained so that $\hat{X}$ gives a reconstruction of the original input sequence [11]. We do this by minimising the mean square error (MSE) between the true and reconstructed sequences:

$$L_{\text{AE-RNN}} = \frac{1}{|X|}||X - \hat{X}||_{2,1}^2 \tag{2.1}$$

Instead of reconstructing the input sequence, the CAE-RNN is trained to reconstruct another instance of the same (predicted) word as the input sequence [13]. Since our training data is unlabelled, we use a UTD system to automatically discover speech segments which are predicted to be similar. For a given pair of sequences (X, Y), the CAE-RNN is fed with input X and then trained to reconstruct Y as its output. We do this by minimising the MSE between the true sequence Y and the predicted output $\hat{Y}$.

$$L_{\text{CAE-RNN}} = \frac{1}{|Y|}||Y - \hat{Y}||_{2,1}^2 \tag{2.2}$$

The intuition behind the CAE-RNN is that the model learns to only encode information that is shared between the input-output segments (such as the word identity) while throwing out nuisance information.

2.4. Evaluation of acoustic word embeddings

We use the same-diff task to evaluate the intrinsic quality of the AWEs produced by the different models [30]. This task works as follows. First an evaluation set of isolated known words is encoded into AWEs using the trained AWE model that is in question (the AE-RNN or CAE-RNN in our case). A decision of whether two AWEs, $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ represent either the same or different words can then be made based on if the distance between the AWEs are less than a chosen threshold. Here we use the cosine distance as in [9] and [13]. The Euclidean distance can also be used, but [9] found that the cosine distance generally leads to better results. The equation used to determine if $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is similar is as follows, where θ is the angle threshold.

$$\cos^{-1}\left(\frac{\boldsymbol{z}_i^\top \boldsymbol{z}_j}{||\boldsymbol{z}_i|| \cdot ||\boldsymbol{z}_j||}\right) \leq \theta \tag{2.3}$$

Next, for varying thresholds, a curve of the precision versus recall is created. Where precision is defined as:

$$\text{Precision} = \frac{\text{Number of word pairs correctly declared as similar}}{\text{Total number of word pairs declared as similar}}$$

and recall is defined as:

$$\text{Recall} = \frac{\text{Number of word pairs correctly declared as similar}}{\text{Total number of similar word pairs}}$$

The area under this curved is called the average precision (AP) and we use this as a metric to measure the quality of the AWEs, where higher values are better.

2.5. Data

In Chapters 3 and 4, we make use of two different languages for the experiments in this thesis, English, from the Buckeye corpus [70] and Xitsonga, from from the NCHLT corpus [71]. In Chapter 5, we use Hausa data from the Globalphone corpus [72].

The English training, validation and test sets each contain around six hours of speech with 12 different speakers in the training set. For Xitsonga, we have a single set of 2.5 hours with 24 speakers. The Hausa training set contains around six hours of speech and the test and validation set each contain around one hour of speech.

Some of our experiments are trained on these full sets, but others like the experiments with the CAE-RNN (Section 2.3) require pairs from a UTD system. The UTD system allows our training to remain independent of speech transcription labels. In the English training set around 14k unique pairs from 12 different speakers of are discovered. In Xitsonga training set around 6k unique pairs with 24 different speakers are discovered. Both languages have an equal number of male and female speakers. In the Hausa training set around 33k unique pairs are discovered from 83 different speakers.

Since there is no validation data available for Xitsonga, we perform all development experiments on the English validation data and then use exactly the same hyperparameters on the Xitsonga set, replicating a true zero-resource setting.

All speech audio is transformed to into either 13- or 39-dimensional MFCCs.

2.6. Chapter summary

We discussed work related to unsupervised AWEs and the downstream tasks in which they can be used. We specifically highlighted work related to query-by-example, UTD and full speech segmentation. This should hopefully give the reader a good sense of how and why AWEs came to be as well as why we are interested in using the AWEs from the CAE-RNN

(it is currently one of the best AWE models). Furthermore we discussed the information that is relevant to the remainder of this thesis. The details of both the AE-RNN and CAE-RNN were discussed as we will use both these models in our experiments. We also described the same-diff task that we use to calculate the AP score of AWEs from one model which we use as a measure of intrinsic quality. Lastly some of the details of the datasets that we will use were discussed. In Chapters 3 and 4 we use the English and Xitsonga datasets. In Chapter 5 we rather use the Hausa dataset. The Hausa dataset has a very unbalanced distribution of speech segments across different speakers, with the majority of segments belonging to one speaker. This uneven distribution is appropriate for the curriculum learning strategy of Chapter 5, but not for the methods used in the other two chapters.

Chapter 3

Speaker and gender normalised acoustic word embeddings

In this chapter we investigate improving robustness of unsupervised acoustic word embeddings (AWEs) models against speaker and gender[1] identity information. We specifically aim to augment the autoencoder recurrent neural network (AE-RNN) and correspondence autoencoder recurrent neural network (CAE-RNN) discussed in Section 2.3 so that their resulting AWEs are speaker and gender normalised.

Two main ideas from previous acoustic modelling research are considered for our approaches. In the first approach we condition the decoder component of the AE-RNN or CAE-RNN on a trained speaker or gender embedding. We also compare initialisation methods for these embeddings, we consider random initialisation and using pretrained embeddings. In the second approach, we use adversarial training where an additional loss term encourages the intermediate representation to be a poor signal for speaker or gender classification. Both of these approaches make use of speaker and gender identity annotations. However, in a low-resource environment these coarse annotations are presumably much easier to obtain than transcriptions.

We evaluate the different approaches by analysing the speaker and gender information retained in the resulting AWEs and by measuring the intrinsic quality using the word discrimination task discussed in Section 2.4.

We show that in both the AE-RNN and CAE-RNN, conditioning models on speaker and gender identity or using adversarial training leads to a reduction in some of this information captured by the AWEs. However, for the English dataset, the intrinsic quality of the AWEs are only marginally improved. The quality of Xitsonga AWEs show greater improvement, with the biggest improvement being from speaker conditioning. We find that conditioning on speaker or gender information leads to better results than adversarial training and that using pretrained embeddings for the initialisation of speaker and gender embeddings do not make a significant difference.

[1]In this chapter we make use of the term *gender* instead of *sex* in order to be consistent with the terminology used in the released corpora.

The work in this chapter is based on the work that was presented at *Proceedings of the Annual Symposium of the Pattern Recognition of South Africa (PRASA)* (2020) in Van Staden and Kamper [49].

3.1. Intuition and related work

We can divide the set of acoustic properties contained in any segment of speech into two different types: the first type defines those that relate to linguistic content and the second type defines those that are independent of linguistic content. We refer to the latter type as *nuisance factors*. Nuisance factors include those that are similar in speech from the same speaker or speakers of the same gender but differ across different speakers or genders even if the contain some of the same word.

In many acoustic modelling tasks it is often only the linguistic content that is important. Therefore it is necessary for acoustic models to generalise well across the variability of different nuisance factors. The intrinsic quality of an AWE depends on the representation of a speech segment's linguistic content, but the AWE can also capture unwanted nuisance factors. We hypothesise that AWE models that are robust against these factors will result in AWEs with better represented linguistic content and therefore ultimately be of higher intrinsic quality. In this chapter we specifically consider speaker and gender identity as nuisance factors.

In Section 3.1.1 we discuss related work on speaker adaptive techniques used in speech processing models. Two of these techniques include incorporating speaker embeddings and adversarial learning, which we expand upon in Section 3.1.2 and Section 3.1.3, respectively.

3.1.1. Speaker adaptation and normalisation in acoustic modelling

Speaker adaptation refers to adapting data or a model to better fit a group of speakers and normalisation refers to making data or a model invariant to speaker information. Our goal is to normalise AWEs. However, take note that normalisation can fall under adaptation and we use the two terms somewhat interchangeably.

There is evidence that suggests that speaker normalisation takes place in the human auditory cortex. Behavioural experiments have shown that humans use speaker information to correctly label vowel sounds: given a sound that is ambiguous between /u/ (as in *"boot"*) and /o/ (as in *"boat"*), humans will label this sound as /o/ after a sentence spoken by a person with a long vocal tract is played and label it /u/ after a sentence spoken by a person with a short vocal tract is played [73]. Recent studies have demonstrated that the parabelt region of the auditory cortex holds speaker invariant representation of speech and conclude that the auditory cortex therefore applies some form of normalisation to

speech [32,74]. This serves as motivation for applying normalisation techniques to acoustic modelling.

Making use of speaker adaptation techniques is present in early acoustic modelling research. Parameter transformation methods like maximum a posteriori (MAP) [75,76] and maximum likelihood linear regression (MLLR) [77,78] are used for speaker adaptation of HMMs. Here the parameters of the HMM model are re-estimated to better match target speakers (often based on a small number of enrolment utterances). Re-estimating the parameters of a DNN is more challenging due to the unpredictability of the weights and the high number of parameters.

Some have looked at transforming neural network parameters by adding linear layers at different levels of a network. Neto [79] investigated adding a layers after the input layer and the output layer, respectively. Gemello et al. [80] proposed adding a layer after the hidden layers of network. Siniscalchi et al. [81] changed the shape of the activation functions in neural networks for different speakers. In all these cases the weights of a trained acoustic model is frozen to which the transformation method is then added and trained with the same objective function. Unfortunately, these methods are prone to overfitting and require a lot of data for the adaptation phase to generalise well.

Research has also been done on speaker normalisation of features. Feature transformation methods like feature-space MLLR (FMLLR) [82] and vocal tract length normalisation (VTLN) [83] are used to extract features with GMM-HMMs systems.

Another approach is to allow the DNN to directly normalise across input speech features without updating any of the network's parameters. Abdel-Hamid et al. [84] introduced using trained speaker embedding representations, called *speaker codes*, to condition a neural network model at various layers. Here a DNN consists of an adaptation network that is prepended to a trained speaker-independent network. The layers in the adaptation network transform the input features into normalised features using the speaker information represented by the speaker codes. Saon et al. [85] also proposed a system that makes use of speaker embeddings as input, however, their system is trained end-to-end, where one DNN model is trained to perform speaker adaptation and the acoustic modelling task simultaneously, and instead of learning speaker codes they use pretrained speaker embeddings called *i-vectors*. They found that this system achieves similar results to using input features transformed by FMLLR or VTLN methods.

Xue et al. [86] proposed that instead of having a prepended adaptation network, only one DNN is trained and conditioned on speaker embeddings at all layers. They found that this method produced better results than in [84] and if the speaker embeddings were initialised with i-vectors, the results further increased.

There has also been research done on directly normalising speaker information in acoustic modelling with adversarial learning. Meng et al. [87] introduced a model that is trained to learn acoustic units (senones) and encourages speaker invariance at the same

time by mini-maximising the loss of a speaker classifier. They found that this methods improve results and that the learned representations had less variability between speakers.

No direct speaker normalisation methods have been introduced to AWE models. The CAE-RNN model does apply some speaker normalisation since the AWEs are trained to encode information that is similar between segments spoken by different speakers. We aim to explicitly normalise AWE by using the ideas presented in the existing conditioning and adversarial methods.

3.1.2. Speaker and gender information as vectors

One popular approach in speaker verification and speaker recognition tasks is to use embeddings that capture speaker characteristics. These speaker embeddings have also proven to be useful in speaker adaptation tasks (as discussed in the previous section).

Dehak et al. [88] proposed a framework for speaker embeddings called i-vectors (named after *identity vectors*). This framework consists of a GMM based universal background model (UBM) that gathers high-dimensional statistics of the data which is then projected to the low-dimensional i-vectors. I-vectors are not used as speaker representations only, they have also been used to represent other types of information like gender identity [89,90].

DNNs can be used to replace or add to the UBM from the i-vector framework to improve results [91–93]. There has also been proposals to replace the whole i-vector feature extraction method with a DNN. Variani et al. [94] proposed a DNN model to extract speaker embeddings called *d-vectors* (named after *deep vectors*). Here a feedforward neural network (FNN) is used to map input speech features to speaker probabilities; the sequence output from the last hidden layer is then averaged and used as the d-vector. However, this model was designed for text-dependent systems (all the input utterances were the same phrase spoken by different speakers). There have been adaptations of this model for text-independent speakers. Li et al. [95] expanded this model for text-independent systems by adding CNN and time-delay neural network (TDNN) [96] layers. They argue that the CNN layers learn local speaker trait patterns and that the TDNN layers extend the temporal context.

Snyder et a [97] proposed a different text-independent model that also makes use of TDNN layers. Their model consists of TDNN layers followed by a statistics pooling layer and then linear layers that map to speaker probabilities. The speaker embeddings can be extracted from either of the two linear layers that follow the pooling layer. They found that i-vectors still produce better results on long utterances, but on short utterances (shorter than 20 seconds) x-vectors produce better results. In an i-vector versus x-vector analysis, Raj et al. [98] found that x-vectors capture speaker and channel characteristics well and suggest that they therefore will be useful in a speaker adaptation task.

3.1.3. Adversarial learning for DNN models

Schmidhuber [99] introduced the idea of having two competing neural network models in a task called *predictability minimisation*. From the first model, consider one hidden layer with h units. Here their second model is trained to predict the value of one of the units given the $h-1$ other units as input and the first model is trained to minimise this predictability. The aim of predictability minimisation is to aid the first model in learning a main task by serving as a regulariser that encourages statistical independence between units in the same layer.

More recently, Goodfellow et al. [100] proposed a generative adversarial network (GAN) framework that shows that with the aid of adversarial learning, neural networks can be very good at modelling distributions of high-dimensional data. Here a generative model estimates the distribution of given training data and a discriminative model is trained to predict if a sample data point is from the training data or produced by the generative model. The models are put against each other by training the generative model to produce samples that the discriminative model will classify as from the training data.

Based on these ideas, Ganin et al. [101] proposed an adversarial learning framework to produce vector representations of images that are invariant to domain information. Here a model maps an input image to a representation which is then used separately as input for two different models, a domain and a target classifier. The loss of the domain classifier is mini-maximised, this means that the domain classifier is trained to minimise its loss, but a negative term is added to the representations' loss function to maximise this classification loss. This encourages the model to not encode domain information into the representations. Others have used this same idea on speech, for noise [102] and speaker [87] robustness.

3.2. Methodology

We use the conditioning and adversarial training approaches discussed in Section 3.1 to try and normalise out speaker and gender information from AWEs. Although gender is a weaker label than speaker identity, as it represents classes of many different speakers, we are still interested to see how gender normalisation will compare to speaker normalisation.

3.2.1. Speaker and gender conditioning

In our first approach for normalising out speaker and gender information from AWEs, we hypothesise that conditioning the decoder components of both the AE-RNN and CAE-RNN on the target speaker or gender will make the model less reliant on speaker or gender information, specifically in the encoder. This means that the resulting AWE will (hopefully) be more invariant to the speaker or gender information. This is different to the approaches of other speaker conditioning models where the embeddings are given

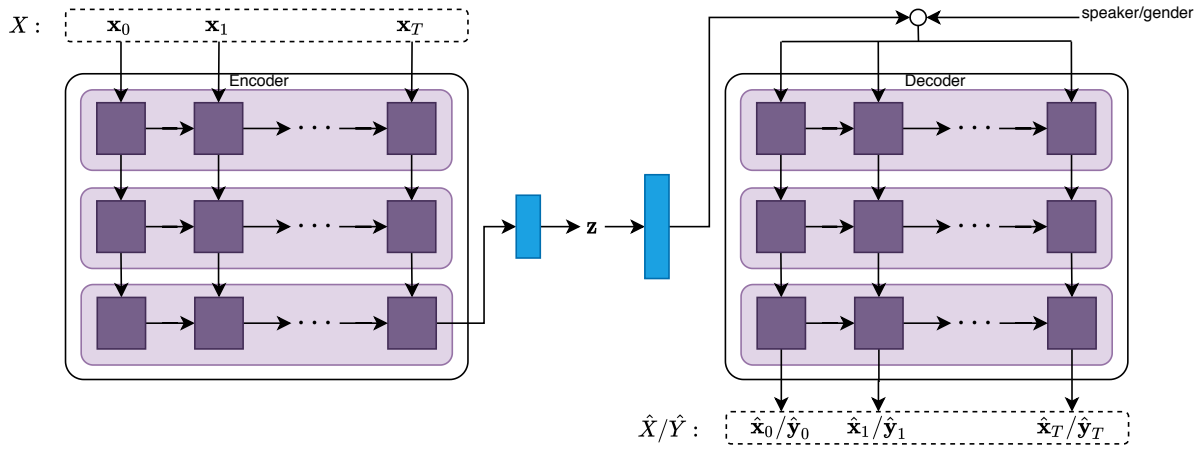

Figure 3.1: The architecture of both the conditioned AE-RNN and CAE-RNN. The AE-RNN will decode the embedding into $\hat{X}$ and the CAE-RNN will decode it into $\hat{Y}$. The speaker or gender vector is concatenated with the input to the decoder at each time step.

as input to the model [85] or at all layers [86]. However, the AE-RNN and CAE-RNN are both trained to reconstruct a speech segment from the AWE; any speaker or gender information contained will be useful for this task and therefore we only condition the decoder component.

An array of trainable embeddings are created for each target speaker and gender in our training set. During training, we append this embedding to the decoder input at each time step. Fig. 3.1 illustrates where in the model the conditioning embedding will be added.

We consider three different ways to initialise the embeddings, namely, randomly, with self-pretrained embeddings or with x-vectors.

Randomly initialised embeddings

The simplest way to initialise the speaker and gender embeddings is to sample them from a random distribution. So, initially each embeddings will not contain any speaker or gender information, but during training it will be updated so that it contains useful information for the primary learning task. This is similar to speaker codes used in [86].

Self-pretrained embeddings

It is possible that the speaker or gender embeddings on their own have not been updated sufficiently by the time the minimisation of the loss function has converged.

In a study on initialisation of word embeddings, Kocmi et al. [103], found that embeddings that are initialised with *self-pretrained* embeddings leads to improved results in language modelling tasks. Here embeddings are extracted from a trained model and then re-used to initialise the embeddings of an untrained model. Inspired by this idea, we extract

the trained randomly initialised speaker and gender embeddings from trained AE-RNN and CAE-RNN models and use them to initialise the embeddings of new untrained models.

X-vectors

Previous studies have shown that i-vectors are useful for speaker adaptation tasks [85, 86]. A more recent type of trained speaker vectors are x-vectors which have also been shown to capture speaker information from short utterances, like in our datasets, better than i-vectors [97].

The model used to extract x-vectors performs transformation at frame- and segment-level. The function, f_{frame}, consisting of five TDNN layers (which will be discussed below), performs frame-level transformations on an input utterance X [96]. The architecture of f_{frame} is depicted in Figure 3.2. At each TDNN layer, frames at time step t are spliced with the surrounding frames to form context vectors, these are then transformed by a linear layer to be used as frames for the next layer. The first layer forms a context vector at t from frames at time steps $[t-2, t+2]$. The temporal context is widened in the next two layers by choosing frames that are further away from t so that the context vectors at t formed at the second and third layer is from time steps $\{t-2, t, t+2\}$ and $\{t-3, t, t+3\}$, respectively. The two layers after this do not add any temporal context and only the frame at t is transformed in both cases. The mean and standard deviation of the output of f_{frame} is then calculated and spliced together, $(\mu_{f_{\text{frame}}}, \sigma_{f_{\text{frame}}})$. This steps ensures that input

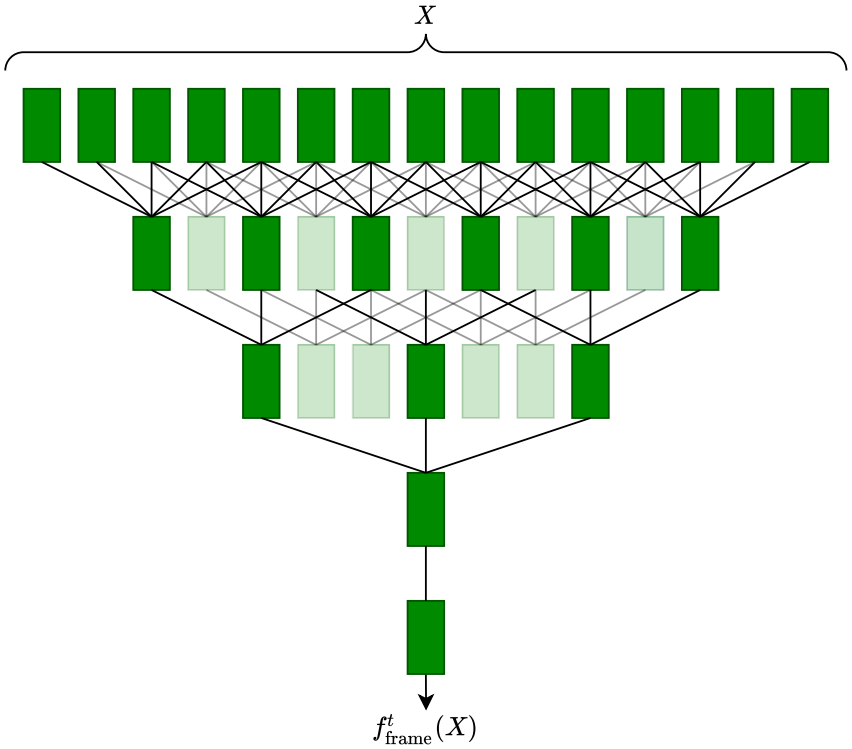

Figure 3.2: The depiction of the frame level operation performed by f_{frame} to produce one output frame at time step t given X as input.

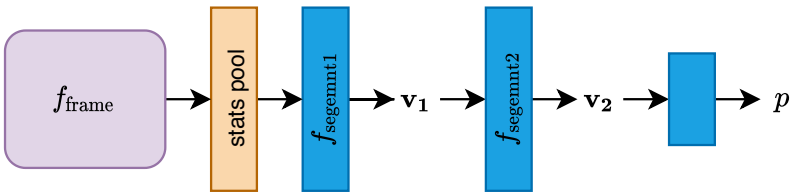

Figure 3.3: The architecture of the model used to extract the x-vectors. Given X as the input, the model is trained to predict the probabilities p of the speakers or genders. Either $\boldsymbol{v}_1$ or $\boldsymbol{v}_2$ can be used as the x-vector.

sequences of different lengths are ultimately transformed to vectors of equal length. Next, two linear layers, f_{segment1} and f_{segment2}, perform transformations at the segment level so that $f_{\text{segment1}}((\mu_{f_{\text{frame}}}, \sigma_{f_{\text{frame}}})) = \boldsymbol{v}_1$ and $f_{\text{segment2}}(\boldsymbol{v}_1) = \boldsymbol{v}_2$. Then $\boldsymbol{v}_2$ is finally mapped by a third linear layer and softmax function to the probabilities of the utterance belonging to each speaker or gender. This model is depicted in Figure 3.3.

For a training set of N speakers the model will return a set of probabilities, $\mathbf{p} = p_1, ..., p_N$. If the true speaker label of the AWE is at c, then the loss function to be minimised for a single utterance is the speaker classification loss function below:

$$L_{\text{SC}} = -p_c + \log\left(\sum_{i=1}^{N} \exp\left(p_i\right)\right) \tag{3.1}$$

We use the same multiclass log loss for gender classification, but here there are only two probabilities determined by the model, female, p_f, and male, p_m. The loss function for a single utterance given as input is the gender classification loss below:

$$L_{\text{GC}} = -p_c + \log\left(\exp\left(p_{\text{f}}\right) + \exp\left(p_{\text{m}}\right)\right) \tag{3.2}$$

Either $\boldsymbol{v}_1$ or $\boldsymbol{v}_2$ can be used as an x-vector. In the original x-vector paper [97] they used $\boldsymbol{v}_1$, but Kanagasundaram et al. [104] found that for short utterances (shorter than 10 seconds) better results are achieved if $\boldsymbol{v}_2$ is used.

3.2.2. Adversarial training

In our second approach to normalise speaker and gender information in AWE, we adversarially train our model against a speaker or gender classifier.

Our approach is similar to the adversarial training method introduced by Ganin et al. [101] and used for speaker invariance by Meng et al. [87]. However, instead of representations that will minimise the target classification loss, our representations (AWEs) are decoded into sequences that will match a target sequence.

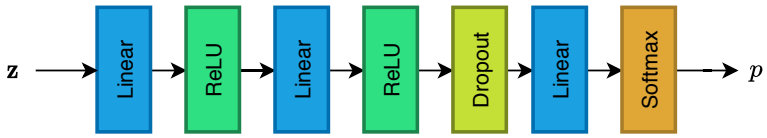

Figure 3.4: The FNN used to classify speaker or gender identity during adversarial training.

We want the AWEs to contain enough information so that it is possible to sufficiently minimise the loss functions defined in (2.1) for the AE-RNN and (2.2) for the CAE-RNN. At the same time we also want to reduce some of the speaker and gender information in the AWEs by maximising the loss of a speaker or gender classifier. For this classifier, we use a FNN trained to predict the probability of an AWE belonging to each speaker or gender in the training set by minimising a multi-class log loss function as defined in (3.1) for speakers and (3.2) for genders. The classifier model is depicted in Figure 3.4. The final loss for adversarial training is is defined as (3.3) for the AE-RNN and as (3.4) for the CAE-RNN below.

$$L^{\text{adv}}_{\text{AE-RNN}} = L_{\text{AE-RNN}} - \gamma L_{\text{C}} \tag{3.3}$$

.

$$L^{\text{adv}}_{\text{CAE-RNN}} = L_{\text{CAE-RNN}} - \gamma L_{\text{C}} \tag{3.4}$$

Here L_{C} is the classification loss for either speaker (3.1) or gender (3.2) of the AWE and the trade-off weight value, γ, is a hyperparameter.

To ensure that the classifier can effectively determine the speaker or gender predictability of the AWEs in training, its weights must be updated regularly. We enter the two models into a turn-based system. During turn A, the weights of the classifier are frozen and we train the model (the AE-RNN or CAE-RNN) to minimise the loss in (3.3) or (3.4). During turn B, the weights of the model is frozen and we train the classifier to minimise the loss in (3.1) or (3.1) given the AWEs produced by the previous turn A as input. This system is depicted in Figure 3.5.

3.2.3. Determining speaker and gender predictability

As an additional analysis task, we investigate the speaker and gender information contained in the AWEs produced by the AE-RNN and the CAE-RNN. We achieve this by using linear speaker or gender classifiers to determine the speaker predictability (SP) and gender predictability (GP) of the AWEs. Speaker or gender classification has also been used in other studies for analysis of AWEs [24, 29] and x-vectors [98]. These classifiers will be trained on the final AWEs produced by the trained AE-RNN and CAE-RNN models. The intuition here is that if the classifiers can achieve a high accuracy then the AWEs contain

a lot of speaker or gender information. Ultimately, all the AWE are compared to see if the SP and GP decrease with speaker and gender conditioning or with adversarial training.

3.3. Experimental setup

3.3.1. X-vectors

We follow the setup of [97] and make the same changes as in [104] for our x-vector model architecture. Here we use a five-layer TDNN based on the model introduced in [96] with the context sizes and dilations for all five layers, in order, as $(5, 3, 3, 1, 1)$ and $(1, 2, 3, 1, 1)$, respectively. The TDNN has a hidden dimensionality of 512 and an output dimensionality of 1500. After the statistics pooling layer, follows two linear layers, both with an output dimensionality of 150. We take the output from the second linear layer to use as the x-vectors.

Then finally there is a linear and softmax layer to transform the output of the previous layer into the speaker probabilities.

This model is trained on the English and Xitsonga full datasets for 1 000 epochs with a batch size of 128 at a learning rate of $1 \cdot 10^{-4}$ using the Adam optimizer [105].

3.3.2. Speaker and gender classifier

During the adversarial training of the AE-RNN and CAE-RNN models, we use the speaker and gender classifiers as depicted in Figure 3.4. This is a FNN with three linear layers with a dimension of 200. After the second ReLU layer we add a dropout layer with a rate of 0.5. We use a learning rate of $1 \cdot 10^{-3}$, a batch size of 50 and the Adam optimiser.

To measure the speaker and gender predictability we also use speaker and gender classifiers. Here, all the parameters are the same as above, except that the classifiers are rather used to predict linear separability and therefore only consist of one linear layer.

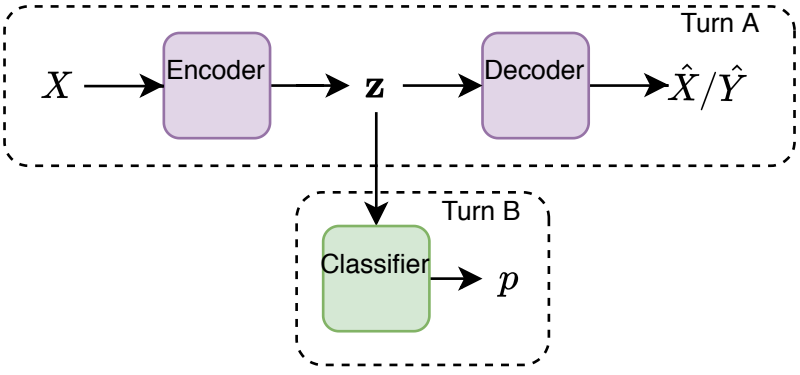

Figure 3.5: In the adversarial training system the AE-RNN or CAE-RNN is trained during turn A and the classifier is trained during turn B.

3.3.3. AE-RNN and CAE-RNN

For the AE-RNN and CAE-RNN, we follow the model setup of [13]. Here the dimension of the latent variables is set to 130. The encoder and decoder of both the AE-RNN and CAE-RNN consist of a stack of three gated recurrent unit (GRU) layers, all with a hidden dimension of 400.

We use learning rates of $1{\cdot}10^{-3}$ and $1{\cdot}10^{-4}$ for the AE-RNN and CAE-RNN, respectively, and both models use a batch size of 256 and the Adam optimiser [105].

When we train our models with the parallel trained speaker and gender embeddings, we use an embedding size of ten. In our developmental experiments, we found that larger embeddings are detrimental to the AWE results. Therefore the self-trained embeddings also have ten d imensions. The x-vectors are much larger at 150 dimensions. If they are used as is, the models converge at a high training loss and the results are worse than those produced by the models without conditioning. We assume this is due to either the large size of the embeddings making it difficult for the GRU to apply sufficient weights to the sequence information, or the large size of the values making up the x-vectors leading to exploding gradients during back propagation. Therefore we use x-vectors to which linear discriminant analysis (LDA) has been applied for dimensionality reduction. Here the embedding size is one less that the number of speakers, so 11 for the English dataset and 23 for the Xitsonga dataset. We also do experiments where the x-vectors are kept at 150 dimensions, but during training we rescale the values of a given x-vector so that the euclidean norm of the vector is at most one.

With the adversarial training approach, the AE-RNN is initially trained for 50 epochs and the CAE-RNN only for 1 epoch, and then the classifier is trained for 30 epochs. After this initial training, the turn system starts, where during turn A, $\gamma = 1 \cdot 10^{-4}$ (Equation (3.3)) for the AE-RNN and $\gamma = 1 \cdot 10^{-2}$ (Equation (3.4)) for the CAE-RNN.

With the English data as input, the AE-RNN and CAE-RNN are trained for 150 and 25 epochs respectively and we use early-stopping on the validation data. Since we do not have validation data from the Xitsonga dataset, we average the number of epochs that it takes to produce the best AWEs on the English validation data for each of the different speaker or gender adaptation methods.

3.4. Experiments

We report results on the English and Xitsonga datasets for conditioning and adversarial approaches. Our findings during the English development experiments are reported in Section 3.4.1 and the English and Xitsonga test results in Section 3.4.2.

3.4.1. Development experiments

During our development experiments there are a few different questions to address: (1) how many layers to condition the models at, (2) which initialisation method produces the best results for speaker and gender conditioning and (3) is it beneficial to combine the conditioning and adversarial approaches?

In previous speaker adaptation work the DNN models were conditioned on speaker information at multiple layers [84, 86]. We also experiment with conditioning at multiple layers and the results for speaker and gender conditioning are shown in Table 3.1 and Table 3.2, respectively.

Here the models (the AE-RNN and CAE-RNN) are conditioned at the layers so that the first layer to be conditioned at is the input to the first GRU of the decoder (as shown in Figure 3.1), the second is at the input to the the second GRU and the third is at input to the third GRU. The speaker and gender embeddings are randomly initialised here. The results shown for when the models are conditioned at zero layers refer to no conditioning. We see that adding conditioning at any number of layers does result in a marginal improvement in the AP and that the SP and GP scores do reduce. The best AP scores on speaker conditioning for both models are found when the models are only conditioned at one layer and interestingly, here the SP and GP scores are higher than in the other two cases (as seen in Table 3.1). However the different AP scores for different layers of conditioning are very close together (within one standard deviation of one another) and therefore we will not conclude that adding speaker conditioning to more than one layer will reduce the AP score, but only that it offers no advantage to the models.

In Table 3.2 we see that even though the SP and GP scores are reduced, gender conditioning on the AE-RNN does not lead to a significantly improved AP score. The improvements on the AP score for the CAE-RNN are better yet still marginal and again the scores are very close together.

Table 3.1: Development results on conditioning on speaker embeddings for different layer counts. Where the layer count is 0 it refers to no conditioning. Displayed is the average precision (AP), speaker predictability (SP) and gender predictability (GP).

Model	Number of layers	AP (%)	SP (%)	GP (%)
AE-RNN	0	26.89 ± 0.39	65.27 ± 4.30	84.28 ± 1.70
	1	$\mathbf{27.83 \pm 0.22}$	52.10 ± 1.19	77.15 ± 0.68
	2	27.67 ± 0.29	48.32 ± 1.22	76.42 ± 0.75
	3	27.79 ± 0.62	48.69 ± 2.82	74.04 ± 0.26
CAE-RNN	0	32.05 ± 0.37	66.00 ± 4.93	82.75 ± 2.61
	1	$\mathbf{34.45 \pm 0.73}$	53.32 ± 1.35	77.03 ± 0.60
	2	33.92 ± 0.53	51.01 ± 0.68	79.59 ± 0.48
	3	33.70 ± 0.65	49.24 ± 0.38	78.98 ± 1.64

Table 3.2: Development results on conditioning on gender embeddings for different layer counts. Displayed is the average precision (AP), speaker predictability (SP) and gender predictability (GP).

Model	Number of layers	AP (%)	SP (%)	GP (%)
AE-RNN	0	26.89 ± 0.39	65.27 ± 4.30	84.28 ± 1.70
	1	26.8 ± 0.28	51.13 ± 3.05	75.81 ± 1.64
	2	26.71 ± 0.31	51.61 ± 4.85	76.6 ± 3.37
	3	$\mathbf{26.95 \pm 0.39}$	51.74 ± 5.53	78.98 ± 2.51
CAE-RNN	0	32.05 ± 0.37	66.00 ± 4.93	82.75 ± 2.61
	1	32.28 ± 0.63	51.92 ± 3.33	78.85 ± 2.53
	2	32.72 ± 0.78	50.94 ± 3.98	78.18 ± 3.84
	3	$\mathbf{32.88 \pm 0.47}$	53.08 ± 7.08	82.51 ± 1.25

Table 3.3: Development results on conditioning on speaker embeddings that have been initialised differently. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %.

Model	Initialisation	AP	SP	GP
AE-RNN	random	27.83 ± 0.22	52.10 ± 1.19	77.15 ± 0.68
	self-pretrained	$\mathbf{27.91 \pm 0.34}$	52.10 ± 2.35	75.93 ± 1.84
	x-vector	27.54 ± 0.31	60.39 ± 2.77	80.07 ± 1.42
	x-vector (LDA)	27.35 ± 0.39	53.87 ± 0.87	77.09 ± 0.31
CAE-RNN	random	34.45 ± 0.73	53.32 ± 1.35	77.03 ± 0.60
	self-pretrained	33.70 ± 0.91	53.50 ± 2.93	81.47 ± 1.12
	x-vector	33.06 ± 0.20	59.6 ± 2.74	81.05 ± 2.31
	x-vector (LDA)	$\mathbf{35.22 \pm 0.75}$	52.71 ± 1.21	78.24 ± 0.91

It is expected that because the CAE-RNN is trained to do indirect speaker and gender normalisation that the AWEs produced by it would have lower SP and GP scores than those produced by the AE-RNN, but surprisingly, the SP and GP scores are similar for the two models.

We experiment to see which is the best way to initialise the speaker and gender embeddings. The results are shown in Tables 3.3 and 3.4 for speaker and gender conditioning, respectively. Here we consider the initialising from a random normal distribution (Section 3.2.1), with self-pretrained embeddings (Section 3.2.1) and with x-vectors (Section 3.2.1). For the speaker conditioning we additionally also experiment with x-vectors that have been projected into a lower dimensional space using LDA. The models are only conditioned at one layer.

In the speaker conditioning initialisation results (Table 3.3) we see that initialising the embeddings of the AE-RNN with self-pretrained embeddings produce the best results, but this AP score is very close to that of the randomly initialised embedding. With the CAE-RNN, the LDA projected x-vectors initialisation resulted in the highest AP score, but

Table 3.4: Development results on conditioning on gender embeddings that have been initialised differently. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %.

Model	Initialisation	AP	SP	GP
AE-RNN	random	26.80 ± 0.28	51.13 ± 3.05	75.81 ± 1.64
	self-pretrained	$\mathbf{27.51 \pm 0.51}$	55.58 ± 1.28	78.00 ± 0.38
	x-vector	27.15 ± 0.27	59.90 ± 1.99	81.47 ± 1.21
CAE-RNN	random	32.28 ± 0.63	51.92 ± 3.33	78.85 ± 2.53
	self-pretrained	$\mathbf{33.76 \pm 0.84}$	57.59 ± 2.20	79.34 ± 0.79
	x-vector	33.36 ± 0.40	61.79 ± 1.03	83.79 ± 0.86

it is only a marginal improvement upon that of the randomly initialised embeddings. With both models, the AP cores of the different initialisation methods are close together which indicates that the initialisation method does not have a significant impact for speaker conditioning.

The gender embeddings initialisation results (Table 3.4) show that the self-trained embedding initialisation leads to the highest scores across both models, but again the improvements on the AP score is only marginal. Take note that condition on gender with randomly initialised embeddings do not improve upon the results of the original AE-RNN, but the self-trained embeddings and x-vectors do. This could indicate that the randomly initialised gender embeddings are not updated sufficiently to contain meaningful information by the time that the model is finished training.

We are interested to see if combining adversarial training with speaker or gender conditioning will be complementary. In Table 3.5 we see results for training the models adversarially against a speaker or gender classifier. With both speaker and gender adversarial training, the AP score is marginally improved upon the original models' score. However, the SP and GP is only marginally reduced in the AE-RNN and with the CAE-RNN it has remained similar. Table 3.6 shows the results for combining conditioning and adversarial methods. Here both models are conditioned on speaker and gender embeddings that have been initialised with the methods (in parenthesis) that lead to the best results as seen in Tables 3.3 and 3.4. We see that combining methods improves upon the AP scores of adversarial training only for both speaker and gender experiments in the AE-RNN and CAE-RNN. However, none of the results improve upon speaker or gender conditioning only and therefore combining the two methods does not have a significant benefit.

3.4.2. English and Xitsonga test results

We apply the conditioning and adversarial training methods to the English and Xitsonga test datasets. All conditioning was performed at only one layer as during the development

Table 3.5: Development results on training the models adversarially against a negative speaker or gender loss. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %

Model	Adversarial term	AP	SP	GP
AE-RNN	Speaker	27.30 ± 0.44	58.01 ± 2.99	78.79 ± 0.90
	Gender	27.03 ± 0.10	60.33 ± 4.30	83.55 ± 0.91
CAE-RNN	Speaker	32.35 ± 0.19	61.43 ± 5.19	83.49 ± 2.14
	Gender	32.11 ± 0.09	63.50 ± 3.13	85.92 ± 1.51

Table 3.6: Development results on combining speaker and gender conditioning with adversarial training. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %

Model	Conditioned	Adv.	AP	SP	GP
AE-RNN	speaker (self-pretrained)	speaker	27.60 ± 0.45	58.56 ± 1.72	79.22 ± 1.35
	gender (self-pretrained)	gender	27.56 ± 0.61	57.04 ± 4.27	77.70 ± 1.04
CAE-RNN	speaker (x-vector LDA)	speaker	34.17 ± 0.20	46.16 ± 1.41	79.53 ± 2.11
	gender (self-pretrained)	gender	33.49 ± 1.12	56.55 ± 3.96	80.99 ± 1.79

experiments we found that more layers do not lead to a significant improvement in the AP score. The speaker and gender embeddings for each experiment were initialised with the methods that resulted in the highest AP score during development experiments.

In Table 3.7 we see the results for applying speaker and gender normalisation to the English test data. The *"Method"* column indicates which method has been used. Take note that the SP and GP scores when no normalisation is applied is much lower to those in the development results, but this is due to the English tests data containing a different set of speakers. For the AE-RNN all the speaker and gender normalisation methods, except speaker adversarial training, result in a marginally improved score upon the original model, seen in the first row. The highest AP score is from speaker conditioning, which is consistent with the development experiments.

All normalisation methods improved upon the AP score of the original CAE-RNN model, seen in the first row of the CAE-RNN results. The best results are from the speaker and gender conditioning methods, speaker conditioning lead to the highest AP score with lowest SP and GP scores, but gender conditioning, with the second highest AP score has the median SP and GP scores. Here the absolute improvement over the AP score for the best models is higher in the CAE-RNN than in the AE-RNN. This indicates that for the English data, the CAE-RNN model benefits most from the normalisation methods. The highest AE-RNN score is significantly less than the lowest CAE-RNN score, which shows that the correspondence training method is more effective in learning higher quality AWEs than our speaker or gender normalisation methods.

Table 3.7: Results from applying the speaker and gender normalisation methods, conditioning (cond.) and adversarial (adv.) training, on the English test data. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %

Model	Method	AP	SP	GP
AE-RNN	–	25.78 ± 0.48	57.83 ± 1.94	91.12 ± 1.31
	cond. speaker (self-pretrained)	$\mathbf{26.60 \pm 0.16}$	48.64 ± 1.91	83.85 ± 0.86
	cond. gender (self-pretrained)	26.45 ± 0.20	56.43 ± 2.99	87.79 ± 0.82
	adv. speaker	25.78 ± 0.17	49.73 ± 3.82	85.86 ± 0.76
	adv. gender	25.86 ± 0.38	54.79 ± 4.6	89.85 ± 1.54
CAE-RNN	–	31.78 ± 0.59	57.79 ± 1.2	91.66 ± 1.03
	cond. speaker (x-vector LDA)	$\mathbf{33.68 \pm 0.66}$	44.39 ± 0.70	79.00 ± 1.30
	cond. gender (self-pretrained)	33.28 ± 0.18	54.91 ± 3.44	87.05 ± 0.66
	adv. speaker	32.13 ± 0.75	50.47 ± 3.73	84.87 ± 1.91
	adv. gender	32.90 ± 0.27	56.39 ± 4.77	89.31 ± 2.03

The Xitsonga test results are seen in Table 3.8. Speaker conditioning, as in the English results, show the highest AP scores and also the lowest SP and GP scores for both the AE-RNN and CAE-RNN. In the AE-RNN results, we see that both adversarial training methods resulted in higher AP scores than gender conditioning. This is different to the English results, where speaker adversarial training resulted in no improvement upon the original model and gender adversarial training resulted in a marginal improvement that was, however, within one standard deviation of the score of the original model. We speculate that this difference in the adversarial training impact is due to more speakers in the Xitsonga dataset. In the CAE-RNN results, we see that speaker conditioning leads to a significant 7.96% absolute improvement over the original model's AP score, where

Table 3.8: Results from applying the speaker and gender normalisation methods on the Xitsonga test data. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %

Model	Method	AP	SP	GP
AE-RNN	–	12.46 ± 0.28	48.42 ± 1.08	91.26 ± 0.37
	cond. speaker (self-pretrained)	$\mathbf{13.81 \pm 0.46}$	38.92 ± 1.51	78.99 ± 1.53
	cond. gender (self-pretrained)	13.11 ± 0.35	42.08 ± 0.33	81.0 ± 1.45
	adv. speaker	13.28 ± 0.22	42.91 ± 1.31	88.25 ± 0.42
	adv. gender	13.20 ± 0.46	43.67 ± 0.66	89.51 ± 0.28
CAE-RNN	–	23.91 ± 0.7	48.08 ± 0.61	91.23 ± 0.51
	cond. speaker (x-vector LDA)	$\mathbf{31.87 \pm 0.95}$	33.88 ± 1.06	74.94 ± 0.61
	cond. gender (self-pretrained)	26.87 ± 0.4	39.15 ± 1.32	77.71 ± 1.79
	adv. speaker	25.39 ± 0.69	41.29 ± 1.19	89.45 ± 0.99
	adv. gender	25.52 ± 0.89	42.91 ± 0.73	89.82 ± 0.52

Table 3.9: Results from conditioning the Xitsonga models on x-vectors that have been encoded with an English-trained x-vector extractor model. Displayed is the average precision (AP) %, speaker predictability (SP) % and gender predictability (GP) %

Model	Conditioned	AP	SP	GP
AE-RNN	speaker	13.72 ± 0.55	38.87 ± 0.73	80.97 ± 1.16
	gender	12.94 ± 0.12	47.59 ± 0.87	83.35 ± 0.93
CAE-RNN	speaker	29.43 ± 1.87	34.46 ± 1.59	77.0 ± 1.79
	gender	26.95 ± 0.22	42.73 ± 1.28	79.2 ± 1.51

speaker conditioning in the English results only showed a 1.90% absolute improvement. Again, we speculate that this difference in impact is due to the higher number of speakers in the Xitsonga dataset. As with the the English results, the CAE-RNN benefits most from the speaker and gender normalising methods and the highest AP score from the AE-RNN results is also much lower than the lowest CAE-RNN score.

As an additional task, we investigate the crosslingual transferability from a English trained x-vector extractor model to Xitsonga data. In Table 3.9 we see the results for training the AE-RNN and CAE-RNN models on Xitsonga data with speaker and gender conditioning where the embeddings have been initialised with Xitsonga x-vectors that were encoded by an English trained x-vector extractor. Although the English dataset contains more speech, the models do not seem to benefit from the English trained x-vector extractor as the AP scores are all lower than the monolingual conditioning methods used in Table 3.8, except gender conditioning on the CAE-RNN, which shows marginal improvement.

3.4.3. Speaker and gender predictability analysis

We investigate the correlation between the AP and, SP and GP scores. In Figures 3.6 and 3.7 we see the scatter plots of the AP scores versus the predictability scores from applying the different speaker and gender normalising methods to models during our development experiments. Here speaker and gender predictability is shown in green and blue, respectively, and the different normalising methods are shown with different marker shapes. SP and GP scores from the same result lie on the same horizontal line. The lines of best fit have been drawn in grey and labelled with the correlation coefficient, r.

With the AE-RNN experiments, in Figure 3.6, we see that both SP and GP have a weak-moderate negative correlation with AP, where with GP it is slightly stronger. The best fit line for GP is also steeper than that of SP. This indicates that GP is a slightly better AP indicator than SP. From the graph it does not seem that there exists a linear relationship between SP and GP scores, but as expected, they do follow the same trend for most of the scores. When no normalising method is applied (marked with a circle) it results in the highest SP and GP scores, so all normalising methods do reduce SP and GP

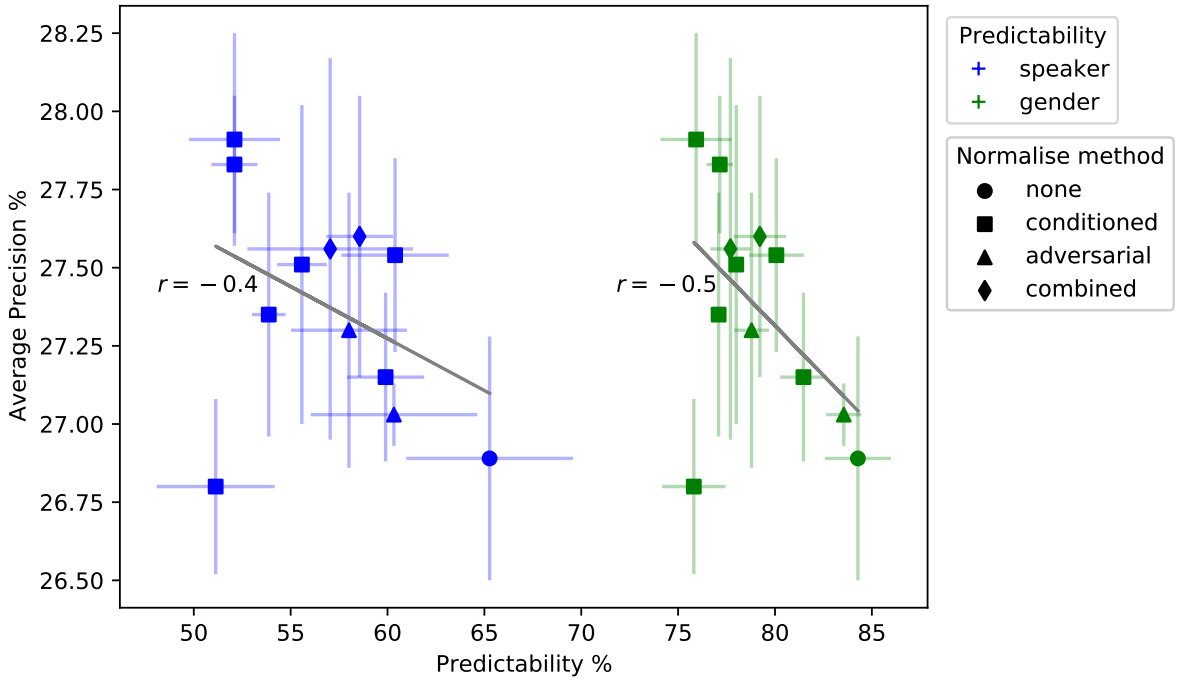

Figure 3.6: The predictability vs the average precision scores for different normalisation methods applied to the AE-RNN. The blue markers indicate speaker predictability and the green, gender predictability, where the standard deviations are shown with transparent lines. The lines of best fit are in grey and r is the correlation coefficient.

scores. The result with the lowest AP score also has the lowest SP and GP scores, but this seems to be an outlier as the other lower SP and GP scores seem to correlate with higher AP scores. We see that the conditioning normalise methods, shown with square markers, lead to the biggest reduction in SP and GP, but also the smallest reduction in SP. Adversarial training, shown with triangle markers, leads to the smallest reduction in GP.

With the CAE-RNN experiments, in Figure 3.7, we see that both the SP and GP scores have a moderate negative correlation with the AP scores. The GP scores have a marginally stronger correlation and their line of best fit is also steeper than that of the SP scores. Therefore the GP score is a slightly better indicator of the AP score and overall both the SP and GP scores are better AP predictors for AWEs from the CAE-RNN compared to those of the AE-RNN. The trends in the SP and GP scores are more dissimilar than in the AE-RNN experiments. Interestingly, the adversarial training methods and one of the conditioning methods, lead to a decrease in SP but an increase in GP. Conditioning and adversarial training combined, shown with diamond markers, lead to the biggest decrease in SP and conditioning leads to the biggest decrease in GP score.

3.5. Conclusion

We have trained AE-RNNs and CAE-RNNs with two different speaker and gender normalising approaches. In the first approach we conditioned the decoder components of the models

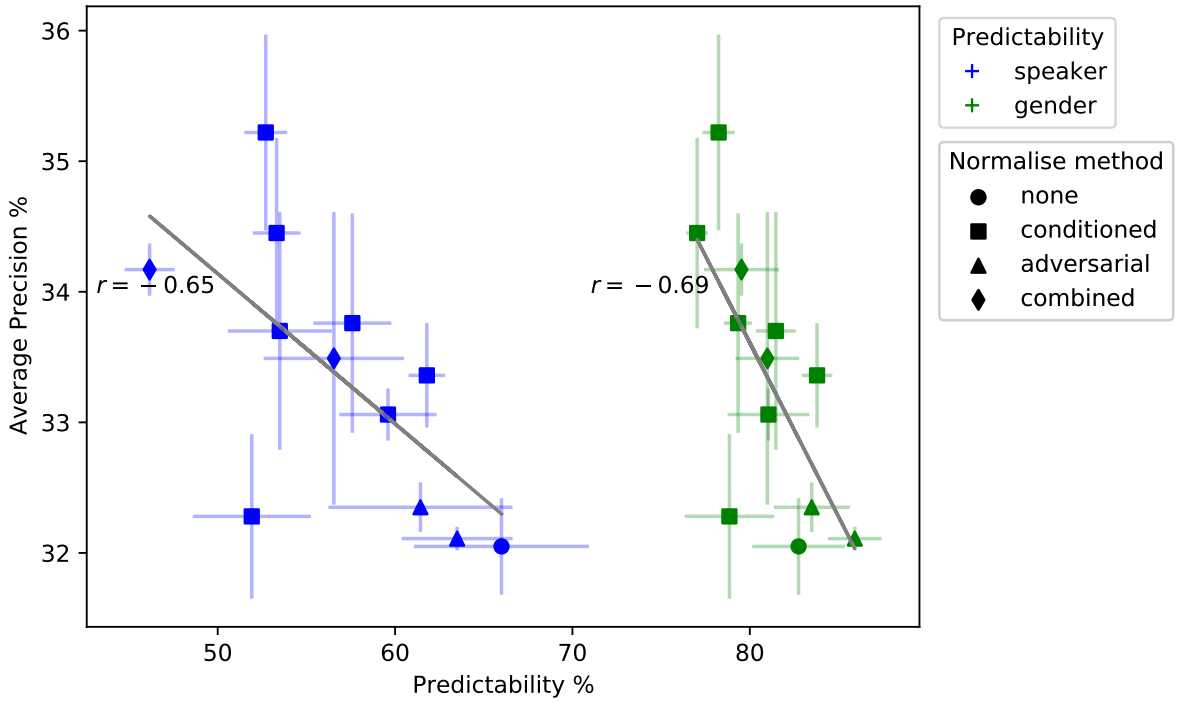

Figure 3.7: The predictability vs the average precision scores for different normalisation methods applied to the CAE-RNN. The blue markers indicate speaker predictability and the green, gender predictability, where the standard deviations are shown with transparent lines. The lines of bet fit are in grey and r is the correlation coefficient.

on speaker or gender embeddings. These embeddings are initialised in three different ways, randomly, with self-pretrained embeddings or with x-vectors. We find that self-pretrained embeddings leads to the best results for speaker and gender conditioning when applied to the AE-RNN and gender conditioning when applied to the CAE-RNNs. X-vectors that have been projected using LDA leads to the best results for speaker conditioning applied to the CAE-RNN. However, the AP scores for using differently initialised embeddings are quite close together and when weighing this up against the extra computation cost of pretraining the x-vectors or self-pretrained embeddings, we recommend using randomly initialised embeddings. In our second approach we use adversarial training to penalise the AE-RNN or CAE-RNN models for capturing speaker or gender information in the AWEs. We also performed experiments where the two approaches are used simultaneously, but we found that this does not offer a significant benefit. When trained on the English dataset, all normalising approaches marginally improve the test AP score and also reduces the SP and GP scores of the AWEs from both the AE-RNN and CAE-RNN models. Speaker conditioning leads to AWEs with the highest AP scores from both the AE-RNN and CAE-RNN models. The best result is from the CAE-RNN with a 5.98 % relative improvement. When trained on the Xitsonga dataset, all normalising approached also marginally improve the test AP score of the AWEs produced by the AE-RNN, but with the AWEs from the CAE-RNN, there is a bigger improvement. Again, speaker conditioning leads to

AWEs with the highest AP scores from both the AE-RNN and CAE-RNN models and the best result is also from the CAE-RNN with a significant 33.29% relative improvement. We speculate that the reason for the bigger improvement in the AP score of the Xitsonga trained models is that the Xitsonga dataset contains more speakers and these speakers overlap in the training and testing set.

In all the results, the AWEs from the CAE-RNN models consistently outperform those from the AE-RNN, even though the SP and GP scores are similar. This shows that the correspondence method used to trained the CAE-RNN is superior to our speaker and gender normalising approaches.

In an analysis of the SP and GP scores of the development experiments, we find that these scores have a weak-moderate negative correlation with the AP scores in the AE-RNN and a moderate negative correlation with those in the CAE-RNN.

The work presented in this chapter shows that normalising out some of the speaker and gender information contained in AWEs leads to at least marginal improvement.

Chapter 4

Frame-level speech representation learning for acoustic word embeddings

In Chapter 3 we investigated improving AWEs by augmenting models at the segment-level (specifically speaker and gender information). Here we will focus on the frame-level. Recent studies have had success with learning frame-level representations that perform well in phonetic category classification tasks [11, 37, 39]. We want to investigate if these learned representations that have been encouraged to capture phonetic contrasts will benefit AWEs if used as input to AWE models. Therefore, we will compare AWEs that have been produced with four different learned frame-level speech representations used as input to the CAE-RNN model. The models for learning these representations are listed below.

First, we consider a frame-level correspondence autoencoder (CAE) [37]. We find pairs of speech segments that are predicted to be of the same type using a UTD system [55]. DTW is then used to align frames between the pair of discovered speech segments. The frame-level CAE model is then trained to predict corresponding aligned frames from each other.

The next two approaches make use of predictive coding mechanisms. In contrastive predictive coding (CPC) [39], representations are learned by predicting the correct future frames from a set containing negative examples. In autoregressive predictive coding (APC)) [40], representations are learned by predicting future frames with an autoregressive model.

Finally, we combine the mechanisms of the CAE and APC representations to form a new representation learning approach, which we call CAPC. Here we use the same aligned pairs used with the CAE and train an autoregressive model to predict future corresponding frames from input frames.

We evaluate the different AWEs produced by each of the four learned representation types as well as by MFCCs in a word discrimination task which was discussed in Section 2.4.

Finally, the different representations and AWEs are submitted to various probing tasks for further analysis. Across two languages, English and Xitsonga, all four learned representations improve upon MFCCs, with the CPC representations consistently producing

the best results when used as input features to the AWE model. As an additional task, we perform crosslingual experiments, where we find that using frame-level representation models trained on a higher resourced language (English) to encode those of a low-resource language (Xitsonga), improves the AWEs of the latter.

The work is this chapter is based on a paper presented at the *IEEE Spoken Language Technology Workshop (SLT)* 2021 [50].

4.1. Related work

Of the most widespread used speech features for acoustic modelling are MFCCs [106], which are also used in previous AWE modelling research [12, 13, 48, 107]. Instead of using these conventional speech features, several zero-resource studies have focussed on learning frame-level speech representations where the linguistic information is emphasised and the nuisance factors like speech identity, emotion or noise are disregarded. These studies have become especially popular with the introduction of the *Zero Resource Speech Challenges* [20,21,108]. Many of the frame-level representation learning methods can be described as bottom-up learning, as the representations learn to capture meaningful contrasts like phone categories directly from lower-level features. In contrast, the CAE-RNN model (Section 2.3), relies on weak top-down constraints in the form of discovered words from a UTD system.

The strategy of ultimately combining the top-down and bottom-up learning methods when the frame-level representations are used as input to the CAE-RNN model relates to the combination of contextual (top-down) and sensory (bottom-up) processing executed in human perception [109].

Several studies have considered also using top-down constraints when training frame-level features; we discuss these approaches in Section 4.1.1. Recently, predictive coding approaches have also been introduced to train frame-level features and this is discussed in Section 4.1.2.

4.1.1. Top-down constraints for learning unsupervised frame-level features

On earlier unsupervised subword modelling methods, Jansen et al. [66] argued that the absence of top-down constraints are the reason for the lack of speaker independence exhibited. They proposed a subword modelling method that makes use of top-down constraints in the form of aligned frames. Here a GMM is trained on speech data and used to create a large UBM. Separately, a UTD system is used to predict pairs of similar speech segments of which the frames are then aligned using a DTW algorithm so that each frame has a corresponding frame from the other speech segment. Next, based on the idea that corresponding frames also correspond to the same subword, the UBM components

belonging to corresponding frames are clustered together to form UBM partitions which serve as acoustic models for the different subwords. They found that this approach improved upon the results of other bottom-up only methods.

Kamper et al. [37] used this same frame alignment idea for a DNN frame extractor which they call a *correspondence autoencoder (CAE)*. Here, for a pair of corresponding frames, the model is trained so that one frame is encoded into a latent variable which is then reconstructed into the other frame. The latent variables can then be used as frame-level representations of which the speaker and noise information has been reduced.

Renshaw et al. [110] enhanced the CAE model architecture and compared its produce representations with those of an AE and a de-noising AE. They found that representations produced by the CAE led to the best results in a phoneme discriminability task.

Thiolliere et al. [111] also used corresponding aligned frames as input to a DNN, but with a siamese network architecture. Here frames from the UTD segments are stacked so that there is one centre frame and six surrounding context frames. Pairs of frame stacks are then transformed by two identical networks to form phonetic representations; the representations are trained to be similar for stacks with corresponding frames. These representations performed better in a phoneme discriminability task than the CAE representations.

Zeghidour et al. [112] improved this model with a triamese architecture, which produces three representations at a time instead of two, which are then put through a triplet loss.

Last et al. [35] compared the triamese representations with those of the CAE when the two models have been trained on exactly the same UTD pairs. They found that the CAE representations led to better results in a same-different task and when the triamese model was trained to produce representations of the same smaller dimension as that of the CAE representations, better results were also achieved in a phoneme discriminability task.

4.1.2. Predictive coding for self-supervised frame-level features

In cognitive neuroscience, predictive coding is a popular framework used to explain how the brain efficiently processes sensory data [113, 114]. The framework postulates that the areas of the brain responsible for higher level processing predict the incoming sensory information rather than the information being registered and processed from lower level processing areas. Unsuccessful predictions lead to error signals in the lower level areas, which in return updates the predictions in the higher level areas [115].

Independent of neural predictive coding, predictive mechanisms are also used in the long-standing audio representation extraction method, linear predictive coding (LPC) [116, 117]. Here, the predictive mechanisms are present in the linear prediction operation that LPC is based on, where a future sample is estimated by the sum of linear mappings from past samples in a discrete audio signal.

Recently, research has considered using these predictive mechanisms in DNN architectures. Van den Oord et al. [39] proposed *contrastive predictive coding (CPC)*, a multi-domain representation extraction framework where representations are learned by predicting the correct future samples from a set containing negative examples. More specifically, an input sequence is mapped to context variables that are trained with the aid of a contrastive loss function [118] to only hold the discriminative information that is useful for predicting the future samples. They argue that by learning to predict across short-time context (as others have also done with word representation learning [119]) the representations learn to hold the underlying higher level information like phonemes and to disregard low-level noise. In their experiments they found that CPC features led to better phone classification results compared to MFCCs and other research has shown that using CPC features also improves results for supervised speech recognition [120].

Chung et al. [40] also utilised predictive coding mechanisms for speech representation learning in a DNN architecture which they call *autoregressive predictive coding (APC)*. Here an autoregressive model is trained to predict the future speech frames a number of steps ahead. The intuition here is the same as with CPC except that the model is not encouraged to disregard non-discriminant features. However, in a follow up study, Chung and Glass [121] argued that representation learning models should not encourage the disregard of certain information as it might be useful for downstream tasks. In their experiments, they found that the APC representations led to better phonetic classification results than those of CPC, but another study [122] found that CPC representations led to better results in a phoneme discriminability task.

4.2. Methodology

We use the different existing frame-level representation learning approaches discussed in Section 4.1 as input for an AWE model, we specifically consider the CAE, CPC and APC representations. Additionally, we combine the mechanisms of the CAE and APC to form a new representation learning approach, CAPC.

4.2.1. Frame-level correspondence autoencoder (CAE)

Unlike the the predictive coding representation learning models (that will be discussed in the following subsections) the CAE model does not have an autoregressive component and therefore the model is not encouraged to encode any temporally shared information. The model is rather encouraged to encode only the information that is shared between frames from different instances of the same (predicted) word. The intuition is that this encourages the model to normalise out noise and speaker information, since these properties could be different for the input and output frames.

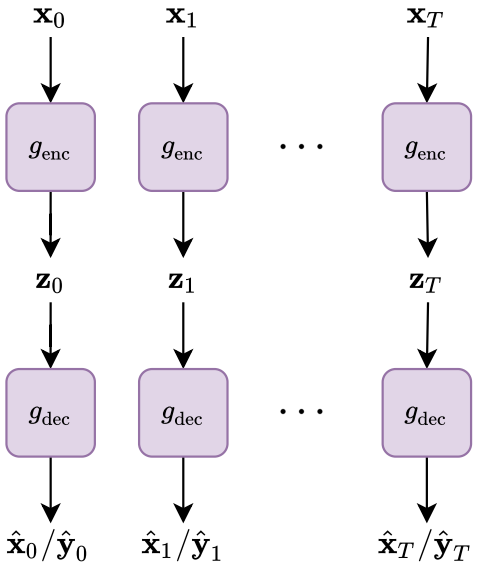

Figure 4.1: The CAE's weights are initialised from those of an AE. The AE is trained to reconstruct the input frames $\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T$ directly. The CAE is trained to reconstruct frames from another segment $\boldsymbol{y}_0, \boldsymbol{y}_1, ..., \boldsymbol{y}_T$, predicted to be a different instance of the same word as the input.

The CAE model produces the best results if its weights are initialised with those of a trained AE [35, 37]. The architectures of these two models are similar and are shown in Figure 4.1.

The AE takes a frame $\boldsymbol{x}_t$ as input. A function $g_{\text{enc}}^{(\text{AE})}$ encodes the input into a latent variable $\boldsymbol{z}_t$. This latent variable is then decoded by the function $g_{\text{dec}}^{(\text{AE})}$, yielding the model output $\hat{\boldsymbol{x}}_t = g_{\text{dec}}^{(\text{AE})}(\boldsymbol{z}_t)$. The target for this output is the input frame itself (from there the $\hat{\boldsymbol{x}}_t$ notation). For a batch of input frames X the AE is trained to minimise the MSE between the input and predicted frames as described by the loss function below:

$$L_{\text{AE}} = \frac{1}{|X|} \sum_{\boldsymbol{x}_t \in X} ||\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t||_2^2 \tag{4.1}$$

For the CAE model, we first have to create pairs of similar frames. Given an unlabelled speech collection, we use a UTD system to find speech segments which are predicted to be of the same unknown type [2]. Pairs of discovered word segments are then aligned using DTW, producing input-output frame pairs for the CAE. Since the segments may differ in length, some frames could be paired with multiple other more than one frame of the other segment.

The architecture of the CAE model is the same as that of the AE model, but instead of decoding $\boldsymbol{z}_t$ in order to predict the input frame itself, we use it to predict the corresponding frame in the pair. Formally, discovered speech segments (X, Y) are aligned to form input-output frame pairs $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, the model takes $\boldsymbol{x}_t$ as input, produces the latent representation $\boldsymbol{z}_t = g_{\text{enc}}^{(\text{CAE})}(\boldsymbol{z}_t)$, and then decodes $\boldsymbol{z}_t$ to obtain the predicted corresponding frame, $\hat{\boldsymbol{y}}_t = g_{\text{dec}}^{(\text{CAE})}(\boldsymbol{z}_t)$. For a batch of input-output frame pairs P, the model is trained to

minimise the MSE between the input and output frames as described by the loss function below:

$$L_{\text{CAE}} = \frac{1}{|P|} \sum_{(\boldsymbol{x}_t, \boldsymbol{y}_t) \in P} ||\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t||_2^2 \tag{4.2}$$

We use the latent variables $\boldsymbol{z}_t$ from the CAE model as the input representations to our AWE models.

4.2.2. Contrastive predictive coding (CPC)

The aim of CPC is to encode only the information that is shared between current and future acoustic observations [39]. This results in speech representations that better describe shared short-time information, like phone categories or speech intonation, depending on how far ahead future observations are. The original CPC study showed that it produced effective speech representation when trained on raw audio waveforms [39]. A more recent study [122] showed that it can also be successfully applied when predicting conventional frame-level acoustic features.

Figure 4.2 shows the architecture for learning CPC representations. A sequence of frames is received as input to the CPC model, with a single frame at time step t denoted as $\boldsymbol{x}_t$. The input frames are encoded by a function g_{enc} into latent variables, denoted as $\boldsymbol{z}_t$ at time step t. In our case, this encoder function consists of a sequence of linear layers. Next, the latent variables are encoded by an autoregressive function g_{ar} into context variables $\boldsymbol{c}_t$. We use a RNN layer for this purpose. Because this function is autoregressive it allows each $\boldsymbol{c}_t$ to be a summary of all $\boldsymbol{z}_{\leq t}$, such that $\boldsymbol{c}_t = g_{\text{ar}}(\boldsymbol{z}_{\leq t})$.

The next step is to determine a prediction score for every $\boldsymbol{c}_t$ at each prediction step. Let K be the chosen number of steps that we want to predict into the future. Then for

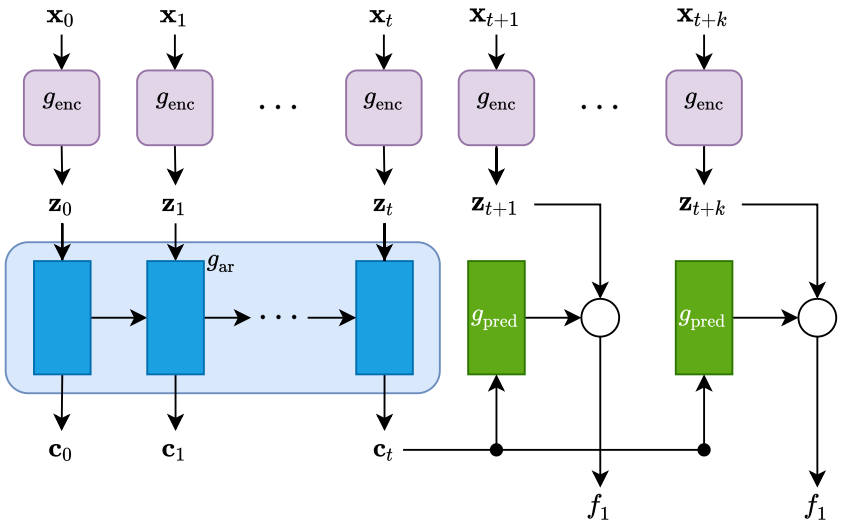

Figure 4.2: The CPC model is trained to compute a score from the context variables, $\boldsymbol{c}_0, \boldsymbol{c}_1, ..., \boldsymbol{c}_t$, and the future latent variables, $\boldsymbol{z}_{t+1}, ..., \boldsymbol{z}_{t+k}$.

each step $k \in [1, K]$ a function g^k_{pred} is used to transform $\boldsymbol{c}_t$. A log bilinear function is then used to calculate the score:

$$f_k(\boldsymbol{z}_t, \boldsymbol{c}_t) = \exp\left(\boldsymbol{z}^\top_{t+k} g^k_{\text{pred}}(\boldsymbol{c}_t)\right) \tag{4.3}$$

Let $\mathcal{Z}_t$ be a set that contains the true $\boldsymbol{z}_t$ along with $N-1$ negative samples of $\boldsymbol{z}_t$. We also calculate prediction scores for the negative samples. The model is then trained to maximise the score for $\boldsymbol{z}_t$ and minimise it for the negative samples. Concretely, the loss function used to do this is based on noise-contrastive estimation (NCE) [118]:

$$L_{\text{InfoNCE}}(\boldsymbol{x}_t, k) = -\log\left(\frac{f_k(\boldsymbol{z}_t, \boldsymbol{c}_t)}{\sum_{\boldsymbol{z}_i \in Z_t} f_k(\boldsymbol{z}_i, \boldsymbol{c}_t)}\right) \tag{4.4}$$

The final loss function applied to the CPC model for a sequence X of input frames can then be expressed as:

$$L_{\text{CPC}} = \frac{1}{K}\frac{1}{|X|} \sum_{k \in [1,K], \boldsymbol{x}_j \in X} L_{\text{InfoNCE}}(\boldsymbol{x}_j, k) \tag{4.5}$$

The NCE loss encourages the model to only encode information that is discriminatory amongst the set of sample frames, $\mathcal{Z}_t$. We take advantage of this by sampling negative frames from different utterances of only the same speaker as the true frame. This encourages the CPC model to normalise out speaker information, since it can't use this information to select the correct frame from among the negative examples [36].

Either $\boldsymbol{z}_t$ or $\boldsymbol{c}_t$ can be used as frame-level representations for a downstream task. But it is recommended to use $\boldsymbol{c}_t$ when extra context from the past is useful [39]. In our development experiments $\boldsymbol{c}_t$ did give better results, and we therefore use it as our input representations to the AWE models.

4.2.3. Autoregressive predictive coding (APC)

Similarly to CPC, the aim of APC is to encode only the information that is shared between current and future frames. The original APC paper [40] argues that when learned representations are encouraged to throw out nuisance information (like speaker identity or noise) there is a risk that useful information might also be lost. So instead of encouraging the model to normalise out non-discriminative features, as with the score maximisation of CPC, an autoregressive function is used to decode the predicted future frame from a latent variable containing general temporally-shared information.

Figure 4.3 shows the APC architecture. A sequence of frames are encoded by an autoregressive function $g_{\text{enc-ar}}$. In our case the autoregressive function is a stack of RNN layers. The last layer's hidden states at each time step is then used as the latent variables,

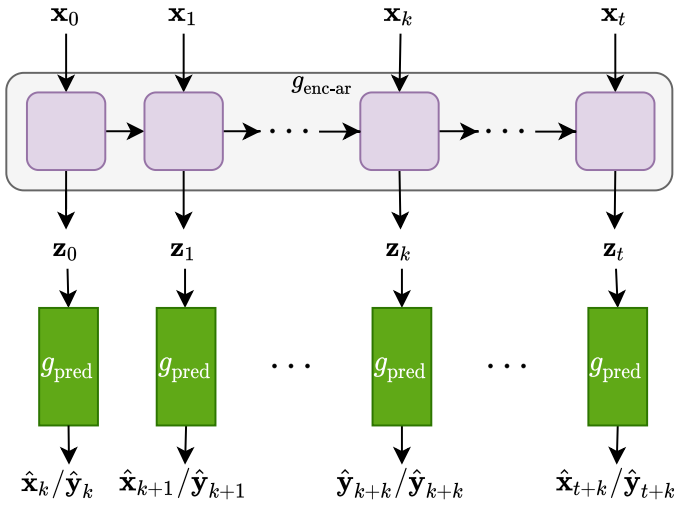

Figure 4.3: The APC model is trained to predict the frame k steps ahead of the input frame from a latent variable. The CAPC model is trained to predict the corresponding frame k steps ahead of the input frame.

denoted as $\boldsymbol{z}_t$ for time step t. Next, a prediction function g_{pred} transforms each $\boldsymbol{z}_t$ to the predicted input frame k steps ahead, such that it can be described as $\hat{\boldsymbol{x}}_{t+k} = g_{\text{pred}}(\boldsymbol{z}_t)$. For a sequence of input frames $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ the model is trained to minimise the mean absolute error (MAE) between the true and predicted future frames:

$$L_{\text{MAE}}(X) = \frac{1}{|X|} \sum_{\boldsymbol{x}_t \in X} ||\boldsymbol{x}_{t+k} - \hat{\boldsymbol{x}}_{t+k}||_1 \tag{4.6}$$

A follow-up study on APC proposed adding an auxiliary loss as a regularisation term [38]. This loss encourages the latent variables to also include information from previous frames in the sequence. Concretely, M different frames are chosen at random from X to use as anchors. An anchor at position m is denoted by $\boldsymbol{x}_{a_m}$. For each anchor we take a slice of X, denoted by A_m, that contains n frames that start s time steps behind a_m, such that $A_m = (\boldsymbol{x}_{a_m - s}, \boldsymbol{x}_{a_m - s + 1}, ..., \boldsymbol{x}_{a_m - s + n - 1})$.

Let $\mathcal{A}$ denote the set that contains every A_m sequence sliced from X. Then the auxiliary loss is given as the MAE loss for every A_m:

$$L_{\text{aux}}(X) = \frac{1}{M} \sum_{A_m \in \mathcal{A}} L_{\text{MAE}}(A_m) \tag{4.7}$$

In our development experiments we found that adding the auxiliary loss does result in a small improvement for the AWEs. The final loss function for our APC model is therefore

$$L_{\text{APC}} = L_{\text{MAE}} + \lambda L_{\text{aux}} \tag{4.8}$$

where λ is a hyper-parameter.

The hidden states from any of the layers of $g_{\text{enc-ar}}$ can be used as frame representations for downstream tasks. Previous research on autoregressive textual word embedding models showed that the information in hidden states are hierarchical across layers [123]. The original APC study [40] concluded that earlier layers in the autoregressive function contain more speaker information and later layers more phonetic information. Therefore we use the hidden states of the last layer $\boldsymbol{z}_t$ as speech representations in our AWE experiments.

4.2.4. Correspondence autoregressive predictive coding (CAPC)

As discussed previous subsections, APC produces representations that are encouraged to contain shared temporal information, whereas those of the CAE model are encouraged to contain information that is shared between frames from similar speech segments. We hypothesise that by combining the mechanisms of both these representation learning methods, the CAPC model will produce representations that carry the advantages of both encoding the higher level short-time information and normalising out speaker and noise information.

The CAPC model has the same architecture as the APC model, but instead of reconstructing the future frames of the input sequence, the model is trained to reconstruct the future frames of the corresponding speech segment, as illustrated in Figure 4.3. Here, the corresponding speech segments are the discovered UTD pairs where the frames have been aligned (as with the CAE model). The CAPC model uses corresponding frame pairs $(\boldsymbol{x}_t, \boldsymbol{y}_t)$ from pairs of aligned speech segment sequences (X, Y) as input-output pairs. Similarly to the APC model, an autoregressive function, $g^{(\text{CAPC})}_{\text{enc-ar}}$ encodes the input frames into a sequence of latent variables, Z. Then a prediction function $g^{(\text{CAPC})}_{\text{pred}}$ transforms each $\boldsymbol{z}_t \in Z$ to the predicted corresponding frame k steps ahead, such that it can be described as $\hat{\boldsymbol{y}}_{t+k} = g_{\text{pred}}(\boldsymbol{z}_t)$. For a sequence of pairs, P, made up of the aligned frames in (X, Y) the CAPC model is trained to minimise the MAE between the true and predicted future frame in Y that is k steps ahead:

$$L_{\text{CAPC}}(X) = \frac{1}{|P|} \sum_{(\boldsymbol{x}_t, \boldsymbol{y}_t) \in P} ||\boldsymbol{y}_{t+k} - \hat{\boldsymbol{y}}_{t+k}||_1 \tag{4.9}$$

Unlike the CAE model that has to be initialised with the weights of a trained AE model, the CAPC model can be initialised with random weights. We use the hidden states of the last RNN layer in $g^{(\text{CAPC})}_{\text{enc-ar}}$ as speech representations, like the APC representations.

4.2.5. Probing the different frame-level speech representations

For further analysis of the different frame-level speech representations and the AWEs produced by them, we perform five probing tasks. The goal is to investigate how the encoded information differs between the different types of representations and if there

are correlations between the increases or decreases in certain types of information and the intrinsic quality of the resulting AWEs. By probing the AWEs, as well as the frame-level representations, we can discover what information in the different frame-level representations is important to the AWE models. We perform two probing tasks on the produced AWEs and three on the frame-level representations as outlined in this section.

Speaker predictability of the AWEs

In Chapter 3, we used linear classifiers to determine the speaker and gender predictability of AWEs from which the models had been augmented in different ways. Other studies have also considered speaker classification as an analysis task on AWEs that were produced by different models [24, 29]. Here, we will use the a linear speaker classifier on AWEs that have been produced by the same model architecture (the CAE-RNN model) but with different types of speech representations as input. Similarly to Chapter 3, we use Equation (3.1) as the loss function for training the speaker classifier. As stated before, the intuition here is that a higher speaker classification accuracy is indicative of more encoded speaker information in the AWEs.

Utterance length predictability of the AWEs

Utterance length prediction has also been used in previous research as an analysis of AWEs [24,29] and x-vectors [98]. We want to investigate if the different types of frame-level representations result in more or less of the utterance length information encoded into the AWEs. Here we use a linear regression model to predict the number of frames that the original input utterance was made up of from the AWE. For a batch of N AWEs, the model is trained to minimise the MSE between the true, T, and predicted, $\hat{T}$, number of frames of the original utterances:

$$L_{\text{UL}} = \frac{1}{N}\sum_{n=1}^{N}(T_n - \hat{T}_n)^2 \tag{4.10}$$

The coefficient of determination (R^2) is then used to evaluate the predictions.

Speaker predictability of the frame-level representations

It is worth investigating if the amount of speaker information encoded into the AWEs correlate to the amount in the frame-level speech representations. Therefore, we also determine the speaker predictability of the representations directly. Another study has also performed speaker predictability probing of frame-level representations, specifically predictive coding representations, but they were interested in analysing the correlation with the training loss [124]. Again, we use a linear speaker classifier and minimise the loss

function of Equation (3.1), but here the model maps the first M frames of an utterance, spliced together, to the speaker probabilities.

Correct sequence order predictability of the frame-level representations

Since the predictive coding representation learning approaches are encouraged to encode shared temporal information, we are interested in investigating if these representations will perform better in temporal-related tasks and how they will compare to each other.

We perform a correct order of sequence prediction task with a linear binary classifier. Here, we extract the first M frames of an utterance and splice them together to form a vector $\boldsymbol{x}_{1:M}$, then we do the same with the next M frames of the utterance and form $\boldsymbol{x}_{M+1:2M}$. Next, we create true samples by splicing $\boldsymbol{x}_{1:M}$ and $\boldsymbol{x}_{M+1:2M}$ together, $\boldsymbol{x}_{1:2M}$ and we create negative samples by doing the reverse, $\boldsymbol{x}_{M+1:2M,1:M}$. The classifier then maps each sample to a probability, p, of it being in the correct order. Given a batch of N samples where at the n^{th} sample, $c_n = 1$ if it is in the correct order and $c_n = 0$ if it is not, the classifier is trained to minimise the following binary cross entropy loss function:

$$L_{\text{CO}} = \frac{1}{N} \sum_{n=1}^{N} c_n \log(p_i) + (1 - c_n) \log(1 - p_n) \tag{4.11}$$

The intuition here is that this task will achieve higher accuracy scores for representation approaches where frames share more information with surrounding frames.

Last frame predictability of the frame-level representations

We consider another probing task that relates to higher level information encoded into the representations. Here, we predict the last frame of an utterance from a set containing negative samples. Following the approach that others have used for language modelling [125, 126], we use linear prediction to form the predicted last frame. Given an utterance $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ and $T - 1$ context matrices, $(C_0, C_1, \ldots, C_{T-15})$, the predicted last frame $\hat{\boldsymbol{x}}_T$ is calculated as follows:

$$\hat{\boldsymbol{x}}_T = \sum_{t=1}^{T-1} C_t \boldsymbol{x}_t \tag{4.12}$$

Take note that the approach followed here is similar to that of CPC. With CPC the goal is also to predict the correct future frame from a set containing negative samples, but here we only try to predict the last frame. The autoregressive function in the CPC approach summarises over previous frames to form context variables which is transformed into the predicted future frame and here we simply sum over previous frames to form the predicted future frame.

Next, we calculate a score between $\hat{\boldsymbol{x}}_T$ and the true (or negative) last frame, $\boldsymbol{x}_T$. For a batch containing N utterances, where $\boldsymbol{x}_{n,T}$ is the last frame and b_n the bias of the n^{th} utterance, the score function, f_{lin}, is described as follows:

$$f_{\text{lin}}(\boldsymbol{x}_{n,T}, \hat{\boldsymbol{x}}_{n,T}) = \exp\left(\boldsymbol{x}_{n,T}^{\top} \hat{\boldsymbol{x}}_{n,T} + b_n\right) \tag{4.13}$$

Our model, g_{LF}, is made up of $T-1$ linear regression functions of which the weight matrices represent the context matrices, C_t, and the bias vectors, $\boldsymbol{b}_t$ will eventually form part of the bias term, b_n. Therefore, the dot product of the output of $g_{\text{LF}}(X_n)$ and $\boldsymbol{x}_{n,T}$ is $\log(f_{\text{lin}}(\boldsymbol{x}_{n,T}, \hat{\boldsymbol{x}}_{n,T}))$, which is made clear in the following derivation:

$$\begin{aligned}
g_{\text{LF}}(X_n) &= \sum_{t=1}^{T-1} C_t \boldsymbol{x}_{n,t} + \boldsymbol{b}_t \\
&= \hat{\boldsymbol{x}}_{n,T} + \sum_{t=1}^{T-1} \boldsymbol{b}_t \\
\boldsymbol{x}_{n,T}^{\top} g_{\text{LF}}(X_n) &= \boldsymbol{x}_{n,T}^{\top} \hat{\boldsymbol{x}}_T + \boldsymbol{x}_{n,T}^{\top} \left(\sum_{t=1}^{T-1} \boldsymbol{b}_t\right) \\
&= \boldsymbol{x}_{n,T}^{\top} \hat{\boldsymbol{x}}_T + b_n
\end{aligned}$$

As with CPC, we want to maximise the score of the true last frame and minimise the score of the negative samples. Therefore, here we also use NCE loss as past of the loss function used to train the model:

$$L_{\text{LF}} = \frac{1}{N} \sum_{n=1}^{N} -\log\left(\frac{f_{\text{lin}}(\boldsymbol{x}_{n,T}, \hat{\boldsymbol{x}}_{n,T})}{\sum_{\boldsymbol{x}_i \in \mathcal{X}_n} f_{\text{lin}}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_{n,T})}\right) \tag{4.14}$$

Again, we assume that the representation learning approaches that result in frames that contain shared information with surrounding frames, will perform better in this task. Additionally, considering that this method is similar to CPC, we want to investigate if the CPC representations, specifically, will lead to better performance in this task compared to the other representations.

4.3. Experimental Setup

4.3.1. Representation learning model implementations

We set up our frame-level AE and CAE models (Subsection 4.2.1) as in [35]. The encoder and decoder functions both consist of six 100-unit linear layers with a 39-dimensional latent variable in between. Through development experimentation we found that on the English datasets the best results are achieved if the AE model is trained for five epochs and

the CAE for ten epochs, and therefore again do the same on Xitsonga. A learning rate of $1 \cdot 10^{-3}$ is used for both models. All our neural networks are optimised using Adam [105].

For our CPC model (Subsection 4.2.2), we use an encoder of six 512-unit linear layers with layer normalisation and ReLU activation functions in between. A dropout layer with a rate of 0.5 is added after the third ReLU activation function. In development experiments we found that the dropout layer does not improve results, but it does stabilise training. We choose a long-short term memory (LSTM) layer as our summarising autoregressive function [127]. The dimensions of $\boldsymbol{z}_t$ and $\boldsymbol{c}_t$ are 64 and 256, respectively. For the contrastive loss in (4.4), based off development experiments, we choose 31 negative examples from the same batch and we predict three steps ahead. The model is trained with a learning rate of $1 \cdot 10^{-5}$.

Each batch contains nine utterances from nine different speakers. We train the English model for a maximum of 15k epochs, but stop at the epoch that produced the best results on validation data. We find that this happens when the model is at a training loss of around 0.93, and so we train the Xitsonga model until it reaches this loss value.

The autoregressive encoder of our APC model (Subsection 4.2.3) consists of a stack of three gated recurrent unit (GRU) [54] layers with a hidden state dimensionality of 512, which is thus also the dimensionality of $\boldsymbol{z}_t$. The predictor function is one linear layer and based on development experiments, predicts two steps ahead. For the auxiliary loss (4.7), based on development experiments, we choose twelve anchors that we use to create sequences of seven frames from 14 steps back and we predict five steps ahead. We use an auxiliary loss weight of $\lambda = 0.1$. In development experiments we found the number of epochs that produces the best AWEs on the English validation data to be 50, and also use this many epochs on the Xitsonga data. We use a learning rate of $1 \cdot 10^{-3}$.

Our CAPC model (Subsection 4.2.4) has the same architecture as the APC model, however, the predictor function only predicts one step ahead and we do not use an auxiliary loss. As with APC, we train the model for 50 epochs with a learning rate of a learning rate of $1 \cdot 10^{-3}$ on both English and Xitsonga.

4.3.2. Unsupervised AWE model implementation

We compare MFCCs, CAE, CPC, APC and CAPC representations (Section 4.2) as input features to the unsupervised CAE-RNN AWE model of Section 2.3. This model is pre-trained as an AE-RNN using (2.1) before switching to the CAE-RNN loss of (2.2). We follow the model setup of [13] as we did in Section 3.3. Due to the different learned representations having larger dimension sizes than that of the MFCC, we increase the dimensionality of the layers in the AE-RNN and CAE-RNN, so that the encoder and decoder functions each consist of a stack of three GRUs with a hidden state dimensionality of 512. For the English dataset, we train the AE-RNN for 150 epochs and the CAE-RNN

for 25 epochs and use early-stopping on validation data. For the Xitsonga dataset, we do not have validation data, so we average the number of epochs that it takes to produce the best AWEs on the English validation data for each of the different types of input representations.

4.3.3. Evaluation

To evaluate the intrinsic quality of the AWEs we use the same-diff task discussed in Section 2.4.

As an AWE baseline, we use downsampling [9, 15]. In this method, we choose ten equally spaced frame representations from a sequence and interpolate them to form an AWE. We obtain downsampled AWEs from each of the representation learning methods considered. For the validation of the CPC model, we also use downsampling and determine the AP score of the downsampled AWEs.

As an additional way of performing the same-different task, we consider using DTW over the speech segments, each segment represented using a sequence of the frame-level representations that are under consideration. Therefore, this approach has access to the full sequences without any compression.

4.3.4. Probing tasks

We split the English validation data 80:20 into training and development sets. All the models are then trained on this training set at a learning rate of $1 \cdot 10^{-2}$ with a bath size of 50 using the Adam optimiser. In order, the speaker prediction on AWEs (SE), utterance length prediction (UL), speaker prediction on representations (SR), corect order prediction (CO), and last frame prediction (LF) tasks are trained using 300, 100, 200, 200 and 300 epochs with early stopping. All the models are linear with the same input dimension as the size of the respective representations. The SE and SR task models have an output dimension of the number of speakers, which is eight. The UL and CO task models have an output dimension of one and for numerical stability, we add a sigmoid layer to the CO prediction results just before the loss function is applied. The LF task model has an output dimension equal to that of the input.

4.4. Experiments

Our main research question is whether improved self-supervised frame-level feature learning is beneficial when used in combination with a segment-level model for producing AWEs. The MFCC, CAE, CPC, APC and CAPC frame-level representations are therefore used as input to AWE models and the results on the English and Xitsonga test data are reported

Table 4.1: AP (%) results on the English test data for DTW and the CAE-RNN and downsampling AWE approaches.

	DTW	Downsampling	CAE-RNN
MFCC	35.90	19.40	30.18 ± 0.34
CAE	$\mathbf{41.49 \pm 1.97}$	21.27 ± 1.51	31.31 ± 1.17
CPC	16.03 ± 0.03	25.38 ± 0.48	$\mathbf{36.83 \pm 0.92}$
APC	30.68 ± 0.26	20.48 ± 0.08	33.55 ± 1.03
CAPC	36.89 ± 0.31	24.42 ± 0.11	35.62 ± 0.62

in Section 4.4.1. Further analysis of the different types of representations are performed on the English validation data of which the results are discussed in Section 4.4.2.

4.4.1. Results

We compare MFCC, CAE, CPC, APC and CAPC representations when used as input representations to the CAE-RNN model of Section 2.3. As a baseline comparison, we also use these representations to create downsampled AWEs. Additionally, we run DTW over all the representations for a direct same-different task evaluation. All experiments are performed three times and we report the AP scores along with the standard deviation.

Table 4.1 shows the AP scores of these evaluations on the English test data. First focusing only on the AWE results (downsampling and CAE-RNN columns), we see that in both cases all the learned representations improve upon MFCCs. In both AWE approaches, best results are achieved when using the CPC representations. The CAPC representations results in higher AP scores than the CAE and APC AWEs, indicating that the combination of the correspondence and predictive learning have been complementary on the English dataset.

Somewhat surprisingly, when the representations are used directly to do the same-different task (DTW column), the only representations to outperform MFCCs are the frame-level CAE and CAPC representations with the CAE representations resulting in the highest AP score overall. Moreover, for both the CPC and APC representations, the corresponding CAE-RNN outperforms its DTW counterpart (e.g. 29.52% vs. 34.76% for the CPC representations). This is despite DTW having access to the full sequences while the CAE-RNN needs to compress the sequences into AWEs. One reason for this could be that the top-down constraints used in both the frame-level CAE and CAPC model are the same as those used in the CAE-RNN model (obtained from the UTD system), and therefore does not provide any additional signal. In contrast, the self-supervision signal for the CPC and APC models are obtained in a bottom-up fashion which is different from that of the CAE-RNN—the top-down signal used in the CAE-RNN seems to be complementary to the bottom-up approach of CPC and APC. This reason could also

Table 4.2: AP (%) results on the Xitsonga test data for DTW and the CAE-RNN and downsampling AWE approaches.

	DTW	Downsampling	CAE-RNN
MFCC	28.15	18.36	22.52 ± 0.29
CAE	$\mathbf{48.32 \pm 1.96}$	26.01 ± 0.33	29.61 ± 3.21
CPC	7.65 ± 0.32	18.66 ± 1.24	$\mathbf{40.93 \pm 0.77}$
APC	18.65 ± 0.42	16.16 ± 0.22	38.96 ± 0.74
CAPC	30.62 ± 0.30	24.38 ± 0.81	34.89 ± 0.64

explain why the DTW result of the CAPC representations is closer to the corresponding CAE-RNN result compared to the frame-level CAE results.

Some similar trends are observed on the Xitsonga data in Table 4.2, but here the affects are even more pronounced. Again, the best AWE approach is when the CPC representations are used as input to the CAE-RNN. Note that, here the APC representations as input to the CAE-RNN results in a higher AP score compared to using the CAPC representations. The reason that CAPC representations lead to worse performance than the APC representations on the Xitsonga data is possibly that the CAPC representations are more sensitive to validation tuning.

The DTW system using CPC representations performs substantially worse than the DTW system using MFCCs, with the CAE representations performing best of the DTW systems. And again the CPC and APC representations as input to the CAE-RNN lead to substantially better performance compared to the corresponding DTW systems (e.g. 38.96% vs. 18.65% for the APC representations). Interesstingly, here the CAPC representations as input to the CAE-RNN lead to a higher AP score compared to the corresponding score of DTW.

Finally, we are interested to see if the learned representations can be used across languages. This is related to previous studies applying frame-level features learned on one language to another [128, 129]. However, here we are specifically interested in the resulting AWEs, which has not been considered before. We train the frame-level representations on English data and then use the trained models to encode the Xitsonga data which is then used as input to the CAE-RNN.

Table 4.3 shows the cross-lingual test results of the resulting AWEs. Again, the CPC representations perform best. Surprisingly, the representations learned on English perform better than using those trained on Xitsonga (Table 4.2) (except for the CAPC representations). The reason for this is likely due to the English dataset containing more speech data. Therefore there is potential for even larger improvements by using more substantial amounts of unlabelled data to train the learned representations.

4.4.2. Further analysis

We want to investigate the information that is encoded into the different frame-level representations and resulting AWEs. We discuss the results of the probing tasks setup as described in Section 4.2.5. All tasks are repeated three times and the average scores and standard deviations are reported.

First, we consider the information that is encoded in the resulting AWEs. In Table 4.4 we see the accuracy and R^2 scores of the speaker classification and length prediction tasks, respectively. From the speaker classification results (in the *Speaker* column), we see that all the AWEs produced by the learned representations, except those of the CAPC representations, have reduced speaker classification accuracy scores compared to those of the MFCCs. The lowest speaker classification score is from the AWEs produced by the CAE representations. This is maybe indicative that the frame-level CAE model is a stronger speaker normalising method compared to CPC where the model is encouraged to only encode frame-level information that is discriminatory within utterances of the same speaker. It is unexpected that the CAPC AWEs have the highest speaker classification score, since both the CAE and APC representations lead to AWEs with reduced speaker classification scores.

Focussing on the utterance length prediction scores (in the *Utt. Length* column), we see that all the AWEs produced by learned representations have reduced R^2 scores compared to those of the MFCCs. Interestingly, all the representations that make use of predictive coding lead to the lowest scores.

Next, we consider the probing tasks performed directly on the representations, these results are seen in Table 4.5. In the left-most column is the speaker classification accuracy scores. The scores of the predictive coding representations, when compared to each other, correlate to the trend of their corresponding acoustic word embedding speaker accuracy scores. However, here the representations of the CAE has the lowest score and the MFCCs the second lowest score. The dimensionality of the predictive coding representations are much larger than that of the MFCCs and CAE representations (e.g., the MFCCs are 13-dimensional and the APC representations are 512-dimensional). It is possible that the linear speaker classifier can better fit the speaker information in the larger representations.

Table 4.3: AP (%) results on Xitsonga when training the frame-level representations on English before applying it to the Xitsonga data to train a CAE-RNN AWE model.

	AP (%)
CAE	34.25 ± 1.61
CPC	$\mathbf{41.79 \pm 0.60}$
APC	40.07 ± 1.13
CAPC	31.88 ± 2.06

In the middle column are the accuracy scores for the correct order binary classification. Here we see that all the learned representations have higher scores than the MFCCs and surprisingly, the CPC representations achieved a 100 % accuracy in all three the repeated experiments. The CAE representations have the second highest score. This seems to counter our initial intuition that representations that share more information with surrounding frames will lead to higher accuracy scores, since the CAE representations are not encouraged to encode temporally shared information. However, the speaker and noise normalising of the CAE could result in the classifier being better able to fit to the natural English language order information in the frame representations.

In the right-most column are the accuracy scores for the last frame prediction. This task shares similarities with the CPC approach and as expected, the CPC representations lead to the highest score. These results seem to correspond with our initial intuition that frames that have more mutual information with surrounding frames will lead to higher scores as the top three scores are from the predictive coding representations.

Finally, we investigate the correlation between the various probing tasks and AP scores of the validation data. Figure 4.4 shows a lower triangular matrix of the correlation coefficients between the different tasks, where the correlation coefficient between tasks i and j are in the i^{th} row and j^{th} column. Take note that for each task there are five different types of representations of which there are three results each. Therefore there are only 15 data values per task and the reader should be wary to draw strong conclusions about the correlations until further investigation with more data.

The strongest positive correlation is between the average precision (AP) and last frame (LF) prediction scores. The strongest negative correlation is between the speaker classification of the representations (SR) and the utterance length prediction (UL). The weakest correlation is between the correct order prediction (CO) and speaker classification of the AWEs (SE). All the tasks, except for SE, have at least a moderate correlation with the AP score.

Table 4.4: The scores for the prediction tasks performed on the final English validation AWEs. On the left is the speaker classification accuracy (%) and on the right is the R^2 score for the utterance length prediction.

	Speaker	Utt. Length
MFCC	64.05 ± 3.89	0.95 ± 0.01
CAE	51.43 ± 1.95	0.94 ± 0.01
CPC	57.83 ± 1.65	0.90 ± 0.02
APC	60.76 ± 3.20	0.90 ± 0.03
CAPC	67.70 ± 4.15	0.88 ± 0.02

Table 4.5: The scores for the prediction tasks performed on the different final English validation representations. The columns from left to right show the speaker classification accuracy (%), correct order binary classification accuracy (%) and last frame prediction accuracy (%).

	Speaker	Order	Last Frame
MFCC	32.72 ± 2.47	83.03 ± 1.24	90.68 ± 0.54
CAE	23.95 ± 0.83	88.06 ± 0.63	85.50 ± 1.45
CPC	53.69 ± 0.60	100.0 ± 0.00	99.15 ± 0.34
APC	57.83 ± 1.64	85.80 ± 1.01	94.09 ± 0.62
CAPC	66.91 ± 0.90	87.93 ± 1.38	92.81 ± 0.31

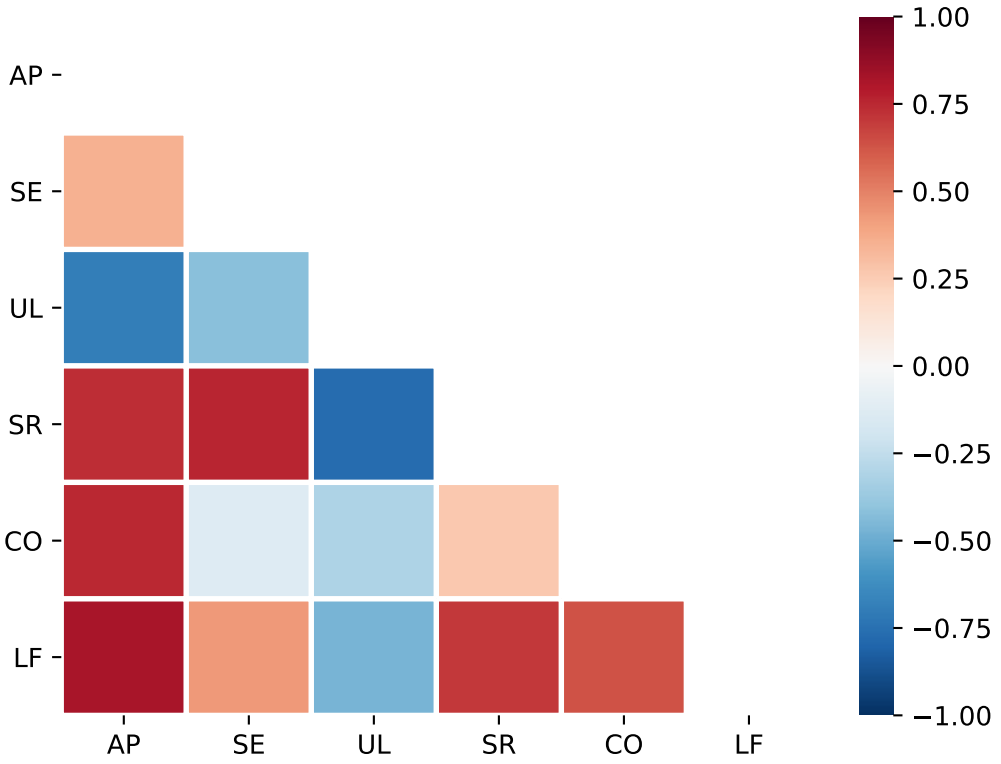

Figure 4.4: The lower triangle matrix of correlation coefficients between the results of the different probing experiments. From left to right on the x-axis is: average precision score (AP), speaker classification of the AWEs (SE), utterance length prediction (UL), speaker classification of the representations (SR), correct order prediction (CO) and last frame prediction (LF).

4.5. Conclusion

In this chapter we considered how AWEs from an unsupervised model can be improved by using frame-level representations from self-supervised approaches as input. Concretely, we

compared CAE, CPC and APC when used as input to the CAE-RNN. We also introduce and compare a new type of representation learning approach, CAPC, that combines the mechanisms of the CAE and APC.

In a word discrimination task on two languages, all AWEs from learned representations outperform those from MFCCs with CPC consistently achieving the best results. On the English data, the CAPC results outperformed the CAE and APC representations, showing that the combined approach was complimentary, but unfortunately this results is not consistent in the Xitsonga experiments.

Different trends were observed when using the features to perform the task directly using DTW: in this case, the AWEs produced by CPC representations performed worst while those of the frame-level CAE performed best.

In a crosslingual experiment we found that that using the larger English dataset to train the representation learning models and then to encode the Xitsonga representations results in better quality AWEs, compared to the Xitsonga only method, for all learned representations except those of the CAPC approach.

We also performed various probing tasks of the different learned representations and MFCCs.

The work presented in this chapter shows that the quality of AWEs can be improved by using better (learned) frame-level representations as input to the AWE model.

Chapter 5

Speaker-based training strategies for acoustic word embeddings

In Chapter 3 we investigated improving AWEs by augmenting the models and in Chapter 4 we investigated improving the frame-level input features. Here, we will consider the training strategy used for both the AE-RNN and CAE-RNN.[1] Inspired by the way that humans are exposed to speech input and curriculum learning strategies that have been used for other DNN models, we investigate if training the AE-RNN and CAE-RNN with a strategy where the number of speakers are gradually increased is beneficial for the resulting AWEs. This strategy is compared to other strategies, including, training on input from a single speaker, from multiple speakers and gradually decreasing speakers. The different strategies are evaluated by evaluating the resulting AWEs. We find that the different training strategies do not have a significant impact on the quality of the AWEs and that the strategies that allows input from multiple speaker in the first few epochs lead to faster convergence.

5.1. Intuition and related work

A lot of acoustic models are trained on speech from the same set of speakers for all iterations and outside of the low-resource setting these sets often consist of thousands of speakers. However, when humans learn to process and recognise speech, they are not introduced to the speech of a large set of speakers immediately. Instead, in the early stages of life, humans likely only hear speech from a small number of speakers (like from household members and caretakers) and as they get older they are exposed to more and more speakers. In Section 5.1.1 below we discuss work that has investigated the impact of the number of speakers at different stages of life.

It is worth investigating if a training strategy that more closely resembles the way that humans are exposed to different speakers will have an impact on AWEs. Other deep learning models used for a variety of tasks have benefited from a training strategy where

[1]The work in this chapter formed part of a collaboration between Stellenbosch University and the University of Edinburgh.

the difficulty of the learnt task is gradually increased and more on this is discussed in Section 5.1.2.

5.1.1. The impact of the number of speakers on speech perception

A range of studies related to human speech perception have lead to results that suggest different levels of impact of the number of speakers a human is exposed to on their performance in speech perception tasks.

Bergmann and Christia [130] did an experimental investigation on the effect of the number of speakers that provide input (three groups of ages four, six and 12 months) to infants on their ability to discriminate native vowels. The data that they gathered indicate that there is likely not a link between the number of speakers and the ability of infants to discriminate between native vowels.

Schatz et al. [131] introduced a phonetic model that successfully resembles observations of infant phonetic discriminability [132] when the model is trained on dataset of roughly evenly distributed speech from more than 20 different speakers with an equal number of male and female speakers. The conditions of their dataset don't match those observed for infants (empirical evidence based on North American children) [133] where the speaker distribution skews heavily towards female speakers and infants don't receive an even amount of speech from all available speakers. Li et al. [134] investigated whether variation in input to this model will have an impact on the results, this included comparing models trained on multiple speakers versus a single speaker. They found that models trained on either a balanced set of multiple or single speaker(s) resemble the observations of the phonetic discriminability of infants, but those trained on a single speaker (which more closely matches the set of speech that infants receive as input [133]) have a slightly closer resemblance.

The before mentioned studies all relate to speech perception in infants, however, other studies suggest that the number of speakers can have a greater impact at later stages in life. Li-Ari [135] investigated the relationship between social network size and speech perception. They performed an empirical experiment where adult participants (between the ages of 20 and 57) had to note the number of speakers that a participant interacted with via speech for in a week and were then scored on a number of speech perception tasks. An analysis of the results revealed that participants that were exposed to more speakers were better at identifying vowel sounds embedded in noise. To find the reason for this correlation, they performed a simulation similar to the experiment to investigate the different variables and they found that the the increase in variability of speech from an increased number of speakers leads to better phoneme categorisation. In a second simulation experiment, they found that at the early stages of learning a larger number of speakers does not have the same impact. The boundaries between native phonetic

categories are not yet known at the early stages of learning and the variation in these categories from different speakers can make the learning task more difficult or not have an impact [133, 135, 136]. However, when the learner is able to discriminate between different categories, a larger number of speakers can aid the learner in robustness against speaker variation [135, 137].

5.1.2. Curriculum learning for DNNs

In behaviour analysis, the term *shaping* refers to a human or animal training strategy where the learner is gradually guided towards a target behaviour by successive reinforcements of increasingly accurate target responses [138, 139]. Research in machine learning has investigated if training models with a similar strategy can be advantageous [140–142]. Bengio et al. [41] formalised this strategy for training machine learning models with increasingly difficult tasks and called it *curriculum learning*. They used this training strategy on various different machine learning models and found that it can result in significant improvement in generalisation compared to training with no curriculum learning strategy. One of the machine learning models that they considered is a DNN language model [143] that is trained to predict the the next word in a sentence. They use text from Wikipedia as their training dataset and implement curriculum learning by gradually increasing the vocabulary size of the input data. During the first iteration, only word that are within the top 5 000 most frequent words are used as input and this is increased with the next 5 000 most frequent words with every iteration.

Other studies have also since taken advantage of curriculum learning [42–46] including those focussing on acoustic modelling. Braun et al. [47] investigated an approach that they call *accordion annealing* to improve noise robustness in automatic speech recognition. In this approach the training of their model is divided into stages. In the first stage the model is trained on speech with low signal-to-noise ratios and in the next stages the range of signal-to-noise ratios is gradually expanded to include higher ratios. They found that this approach showed improved performance over a model that is trained on wide range signal-to-noise ratio speech from the start. This approach seems counter-intuitive and different from other curriculum learning methods as the model starts with training on the most noisy speech (a more difficult task than training on clean or less noisy speech) but [47] argues that this allows the model to explore a wider parameter space in the first stages and they further show that this approach also outperforms a model that is instead first trained on high signal-to-noise ratio speech. This strategy has also been used for a denoising AEs [43].

5.2. Methodology

We will implement and compare various speaker number based training strategies on the AE-RNN and CAE-RNN models. This includes a curriculum learning strategy, discussed in Section 5.1.2, where we will first train the models on a single speaker and then gradually increase the difficulty by incrementing the number of speakers in the training set. Incidentally, the way in which we train the AE-RNN and CAE-RNN consecutively can also be viewed as a curriculum learning strategy: first the AE-RNN is trained to reconstruct the input speech segment, an easy task, and then with the same model parameters, the CAE-RNN is trained to reconstruct the speech segment in the same pair as the input, a more difficult task.

We setup four different training strategies and they are discussed below:

Multiple speaker learning (multi): At each iteration the model is trained on speech segments from all the speakers in the training dataset. This is the same strategy followed for the AE-RNN and CAE-RNN models in Chapters 3 and 4 and in most other AWE studies [29].

Single speaker learning (single): The model is trained on speech segments from one speaker. We choose the speaker with the most speech segments or with the most same speaker UTD pairs, for the AE-RNN and CAE-RNN, respectively. A multi and single strategy were also compared to each other in [134] in a phoneme discrimination task, where they found that both strategies result in similar trends. It is expected that the multi strategy will result in better quality AWE, but it will be interesting to see what the impact of speaker variation is by comparing it to the AWEs from the single strategy.

Incremental speaker learning (incr): At each stage/iteration the model is trained on speech segments only from j selected speakers from the training set; initially $j = 1$ and then increments with every subsequent stage. The j speakers are always the top j speakers with the most speech segments. We assume that this will be an adequate approximation for the top j speakers with the most UTD pairs consisting of speech segments from only the j speakers. This is a curriculum learning strategy where difficulty is linked to the variation of speakers.

Decremental speaker learning (decr): As in the incr strategy, the model is trained on speech segments from only j selected speakers at each stage, but here j is initially equal to the total number of speakers in the training set and then decrements with each stage. Again, the j speakers are always the top j speakers with the most speech segments. This is the opposite of a curriculum learning strategy as the model is first trained on the most difficult task and then the difficulty is decreased at each stage. It will be interesting to compare the results of this strategy to the incr strategy.

If it proves to be true that the gradual increase in difficulty aids the models to produce better quality AWEs, then it is expected that the decr strategy should have a detrimental impact on the quality of the AWEs.

5.3. Experimental setup

The AE-RNN and CAE-RNN models follow the setup in [13], described in Section 3.3.3 and the AWEs are again evaluated using the same-diff task (Section 2.4).

For the incremental and decremental training strategies, we want there to be a sufficient number of stages to train the models effectively. Therefore, for the experiments in this chapter we use the Hausa dataset (of which more details were discussed in Section 2.5) which contains 83 different speakers which is considerably more than the English and Xitsonga datasets with 12 and 24, respectively. The Hausa training set is also very unbalanced with the speaker with the most speech segments containing about x % of the total segments, which means that the single, first stage of the incr and last stage of the decr training strategies should hopefully allow a sufficient amount of input for the models. For the multi strategy and stages of the incr and decr strategies where speech segments from multiple speakers are selected we choose speech segments so that there is somewhat speaker uniformity to prevent the models from having a bias towards the most frequent speakers. Therefore, at each stage with j speakers with the j^{th} having N_j speech segments or pairs consisting of speech segments from the j speakers only, the number of speech segments or pairs to choose is defined as follows:

$$\max\left(\min_{i\in[1,j]}\left(N_i\right), N_{\min}\right) \tag{5.1}$$

$N_{\min}$ is 200 for the AE-RNN and 1 400 for the CAE-RNN, which was picked based off of development experiments.

We experiment with two different methods of updating the stages in the incr and decr training strategies. One option is to update a stage after one epoch, like in [41] where the difficulty is increased after every iteration. In this we train the AE-RNN and CAE-RNN on an additional 20 and 10 epochs, respectively, after the last speaker has been added to the training set. The other option is to exhaustively train the model at each stage and then load the model parameters which resulted in the best validation score for the next stage, like in [47]. Here we train the models for 120 and 15 epochs at each stage for the AE-RNN and CAE-RNN, respectively. Needless to say, this option requires considerably more training time compared to the first option with total epoch counts of 9960 and 1245 versus 103 and 93 for the AE-RNN and CAE-RNN, respectively.

5.4. Experiments

We report results on the Hausa dataset. There are a number of training strategy explorations to consider, which we discuss in Section 5.4.1 and the test results of the best training strategies are discussed in Section 5.4.2.

5.4.1. Development experiments

We explore the following questions: (1) What is the impact of training AWEs on multiple speakers? (2) What is the best stage update method? (3) Does speaker based curriculum learning have a positive effect? (4) Does the opposite of a curriculum learning strategy have a negative effect?

First we compare AWEs that have been trained on multiple speakers vs on a single speaker. These results are shown in Table 5.1. The subsequent AE-RNN and CAE-RNN can be trained with the same or different strategy and the respective columns show which was used for each model.

From the results in the table we see that training on multiple speakers results in AWEs with significantly higher AP scores, shown in the first row, compared to training on a single speaker only, shown in the second row. We know from previous work [13] that training the CAE-RNN without initialising with the pretrained AE-RNN parameters leads to poor results. However, training on a single speaker is an easier task than training on multiple speakers and we investigate if training the CAE-RNN on randomly initialised parameters with a single speaker will also lead to poor results. In the third row, we see that this does indeed also result in a very poor AP score (worse than the AE-RNN trained on a single speaker).

Next, we investigate which stage update method is better for the incr strategy. For now, we only compare two different AE-RNN and CAE-RNN combinations of strategies and these results are shown in Table 5.2. The first method is to exhaustively train the model at each stage and load the best parameters at the next stage, denoted by BM. The second methods is to update the stage after one epoch, denoted by OE.

Table 5.1: Development results for the multi versus the single curriculum learning strategies. In the AE-RNN and CAE-RNN columns are the strategies used for each and the AP scores are shown in the AP columns.

AE-RNN	AP (%)	CAE-RNN	AP (%)
multi	25.81 ± 0.38	multi	41.18 ± 0.64
single	14.08 ± 0.38	single	20.52 ± 0.40
none	-	single	6.70 ± 0.76

Table 5.2: Development results for using the one epoch till next stage (OE) and best parameters till next stage (BP) stage update methods. In the AE-RNN and CAE-RNN columns are the strategies used for each and the AP columns shows the scores of the final AWEs.

AE-RNN	AP (%)	CAE-RNN	AP (%)
single	14.08 ± 0.38	$\text{incr}^{(\text{BM})}$	29.92 ± 0.11
single		$\text{incr}^{(\text{OE})}$	31.16 ± 0.59
$\text{incr}^{(\text{BM})}$	19.03 ± 1.5	multi	32.53 ± 2.69
$\text{incr}^{(\text{OE})}$	22.46 ± 0.94	multi	$\mathbf{39.53 \pm 0.67}$

From the results in the table we see that the OE method consistently leads to higher AP scores compared to the BM counterparts, e.g. 31.16 versus 29.0.11 in the CAE-RNN column of the top two rows and 22.46 versus 19.03 in the AE-RNN column of the two bottom rows. It is possible that the BM method guides the model parameters into a local minima before the last stage which leads to worse performance.

In Table 5.3 are the results for different incr strategy combinations. There are a large number of combinations that can be investigated, but we consider only the four where the AE-RNN strategy is as easy or easier than the strategy used for the CAE-RNN (the difficulty order is: none, single, incr, multi).

From the results we see that training the AE-RNN with the incr strategy leads to the best CAE-RNN results. However, the multi-multi strategy in Table 5.1 still achieves a slightly higher AP score. This shows that the incr strategy, unfortunately, has slight negative or no impact on the AWEs. Here the CAE-RNN initialised without the parameters of a pretrained AE-RNN produces AWEs also with an AP score significantly lower than the CAE-RNN initialised with the parameters of an AE-RNN trained with the single strategy (like in Table 5.1).

Finally, we investigate following a training strategy that is the opposite to a curriculum learning strategy. The results for different combination of the decr strategy are shown in Table 5.4. Again, there is a large number of possible combinations but here we only consider the three where the strategy for the AE-RNN is more difficult or as difficult than the strategy used for the CAE-RNN (order of difficulty is single, decr, multi).

Table 5.3: Development results for different combinations of the incr strategy. In the AE-RNN and CAE-RNN columns are the strategies used for each and the AP columns shows the scores of the final AWEs.

AE-RNN	AP (%)	CAE-RNN	AP (%)
single	14.08 ± 0.38	incr	31.16 ± 0.59
none	-	incr	23.57 ± 1.52
incr	22.46 ± 0.94	multi	$\mathbf{39.53 \pm 0.67}$
incr		incr	38.82 ± 0.97

Table 5.4: Development results for different combinations of the decr strategy. In the AE-RNN and CAE-RNN columns are the strategies used for each and the AP columns shows the scores of the final AWEs.

AE-RNN	AP (%)	CAE-RNN	AP (%)
multi	25.81 ± 0.38	decr	$\mathbf{41.11 \pm 0.29}$
decr	22.10 ± 0.36	single	33.93 ± 1.58
decr		decr	38.51 ± 0.38

Table 5.5: Test results for the top three training strategy combinations. In the AE-RNN and CAE-RNN columns are the strategies used for each and the AP columns shows the scores of the final AWEs.

AE-RNN	AP (%)	CAE-RNN	AP (%)
multi	21.56 ± 0.27	multi	34.16 ± 0.74
multi		decr	34.20 ± 0.37
incr	20.91 ± 0.39	multi	33.59 ± 0.14

From the results we see that the highest AP scores are from the AWEs produced by the CAE-RNN trained with the decr strategies, in the first and third row. Interestingly, AP score from the multi-decr strategy is higher than any of the incr strategy combinations and within one standard deviation of the score from the multi-multi strategy. Therefore the opposite of a curriculum learning approach does not have a negative impact and it seems that there is actually slightly better performance if the model is introduced to multiple speakers at an early stage.

Overall, the results seem to indicate that the specific training strategy has very little or no impact as long as both the AE-RNN and CAE-RNN are trained on multiple speakers (this includes the multi, incr and decr strategies).

5.4.2. Hausa test results

The test AP scores for the top three development training strategy combinations (multi-multi from Table 5.1, incr-multi from Table 5.3 and multi-decr from Table 5.4) are shown in Table 5.5. The highest AP score is from the multi-decr strategy AWEs in the second row, however, all three scores are within one standard deviation of the multi-multi AP score in the first row. From the development and test experiment results, it is evident that the selected speaker-based strategy has very little or no impact as long as both the AE-RNN and CAE-RNN models are exposed to input from multiple speakers.

5.4.3. Further analysis

We are interested to see how the AP scores of the different strategies at different epochs compare, especially to those with increasing and decreasing speaker numbers. The epoch

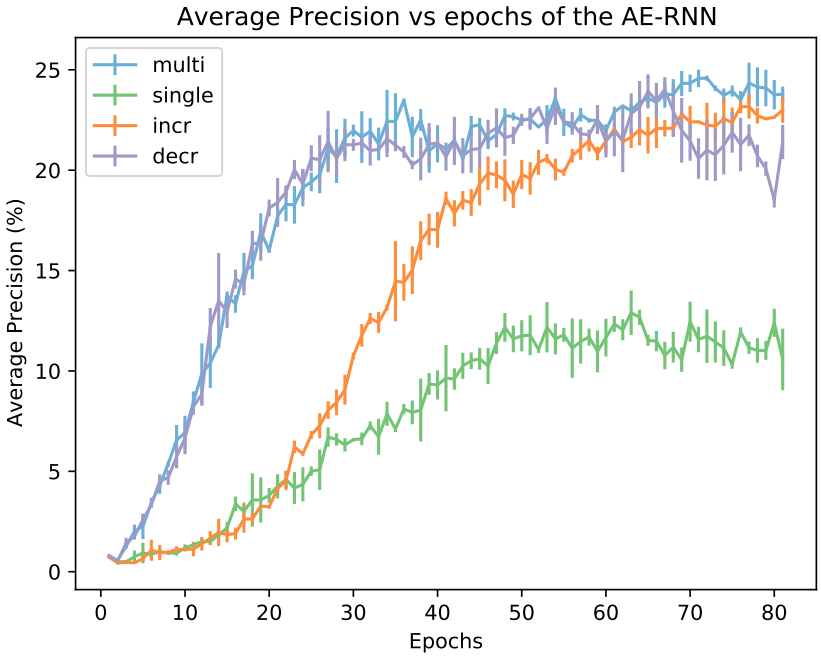

Figure 5.1: The average precision score at every epoch for the different training strategies applied to the AE-RNN. The vertical bars indicate the standard deviation.

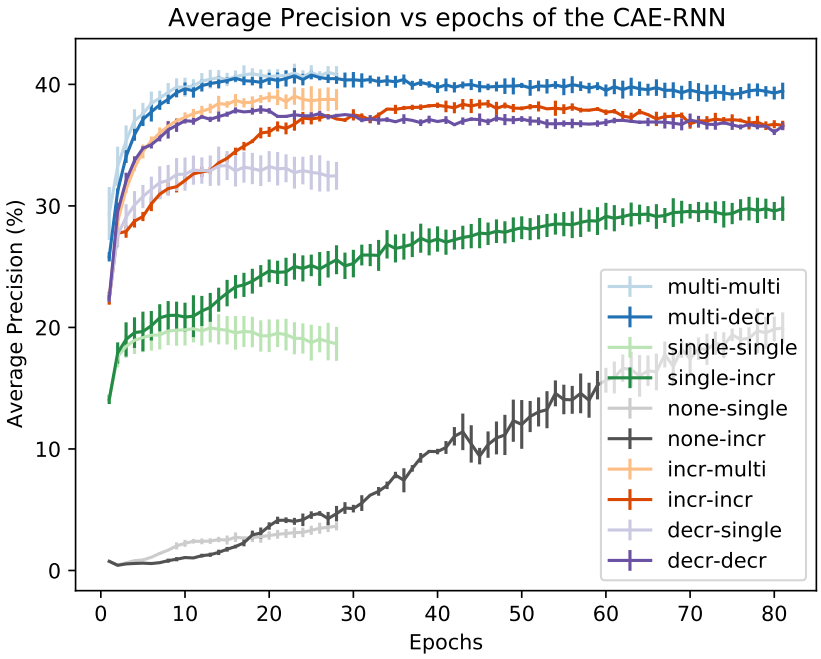

Figure 5.2: The average precision score at every epoch for the different training strategies applied to the CAE-RNN. The vertical bars indicate the standard deviation

number versus the AP scores of various strategies reported on in Section 5.4.1 are shown in Figures 5.1 and 5.2 for the AE-RNN and CAE-RNN, respectively. We show the first 83 epochs which is up until all speakers have been added in the incr strategies or removed in

the decr strategies. Take note that in Figure 5.2 all the light colour lines end at 30 epochs and this is because the CAE-RNN trained with the single and multi strategies are only trained for so many epochs.

The different strategies seem to follow the same rate of increase in AP score across models. In the figure we see that there is a fast increase in the AP scores for the multi (blue line in Figure 5.1 and light blue and light orange in Figure 5.2) and decr (purple line in Figure 5.1 and dark blue and dark purple in Figure 5.2) strategies. Although, [41] found that curriculum learning training strategies lead to faster convergence, here, the similar trends in the multi and decr strategies shows that many speakers in the early stages are what drives faster AP score convergence where the incr strategies (orange line in Figure 5.1 and dark green, dark grey and dark orange lines in Figure 5.2)) instead show a slower increase in AP score. Also notice that after about 20 speakers have been introduced to the AE-RNN with the incr strategy the increase in AP score is accelerated, again highlighting that many speakers drive faster AP score increase.

5.5. Conclusion

We have investigated various speaker number based training strategies for the AE-RNN and CAE-RNN. This includes training on multiple speakers at all epochs, like in Chapters 3 and 4 on a single speakers. Inspired by curriculum learning, we train the models on an increasing number of speakers, to better mimic the way that humans are exposed to different speakers. To further investigate the different number of speakers at each stage, we train the models with the opposite of a curriculum learning strategy, by decreasing the number of speakers at every epoch.

Unfortunately, the curriculum learning inspired strategy does not improve on the quality of the AWEs trained on multiple speakers. In fact, we find that as long as both the subsequently trained AE-RNN and CAE-RNN receive input from multiple speakers at some stages, there is very little difference between the results. Additionally, in an analysis we find that input from many speakers in the early epochs lead to faster convergence of the measure of the AWE quality.

There are two reasons for the absence of AWEs improvement to consider. Firstly, it could be that an increase in speaker number is not a task well suited for a curriculum learning strategy as other types of tasks have achieved significant success [41]. Secondly, there might be a too small amount of input data. The experiments in this chapter were conducted in a zero-resource setting (as it is the focus in this thesis) with a limited number of hours of speech, but infants are exposed to a considerable number of hours of speech [133] and other studies that have successfully modelled infant speech trends [131, 134] have taken this into the consideration.

Chapter 6

Conclusion

6.1. Thesis summary

We have investigated various methods at an attempt to improve the intrinsic quality of unsupervised AWEs produced by both the AE-RNN [11] and CAE-RNN [13] models.

In Chapter 3 we considered two approaches to normalise out speaker and gender information, speaker and gender conditioning and adversarial training, and investigated if a reduction in this information does indeed result in improved AWEs.

In the conditioning approach, the decoder component of either the AE-RNN or CAE-RNN is conditioned on speaker or gender embeddings. Conventionally, embeddings are initialised randomly, but some studies have had success with initialising embeddings with other trained embeddings like i-vectors [86] or self-pretrained embeddings [103]. Therefore, we also compared using different initialisations: from a random uniform distribution, x-vectors (a DNN approach for i-vectors) and AE-RNN and CAE-RNN self-pretrained embeddings. The self-pretrained and x-vectors initialisations do lead to better results, but it is still very close to the results of the randomly initialised embeddings. In the adversarial training approach, the AE-RNN or CAE-RNN is penalised with a negative loss term if the speaker identity or gender can be predicted from the AWEs. We find that the best of the two approaches is conditioning.

On the English dataset the best result is from applying speaker conditioning to the CAE-RNN with the speaker embeddings initialised from LDA projected x-vectors with a 5.98 % relative improvement compared to the original CAE-RNN. We see a more significant improvement on the Xitsonga dataset with speaker conditioning on the CAE-RNN having a 33.29% relative improvement. However, the bigger improvement in the Xitsonga dataset is likely due to the Xitsonga training set having more speakers (24 versus 12 in the English training set) and overlapping speakers between the training and test sets.

Additionally, we analysed the speaker and gender information captured in the AWEs by training and evaluating the final AWEs of all the different methods on a speaker or gender classifier. We find that these method did indeed reduce the speaker and gender information captured by the AWEs, but only slightly. Furthermore, we find that there is a correlation between AP and the amount of speaker or gender information. This suggests

that if more of this information can be reduced it can lead to more improved AWEs, but this is evidently a harder problem than previously thought.

In Chapter 4 we implemented different frame-level speech representation types, three of them are from previous studies, the frame-level CAE model [37], CPC [39] and APC [38] and a fourth is our own, CAPC, which is a combination of the frame-level CAE and APC.

The frame-level CAE uses top-down information from pairs of speech segments that are predicted to be of the same type using a UTD system [55]. The model is then encouraged to only encode information that is shared between aligned frames and disregard what is not, like speaker and noise information. In a probing task, we do find that the frame-level CAE representations contain the least amount of speaker information and also result in AWEs with the least amount of speaker information. However, although the frame-level CAE representations result in the best DTW AP score, AWEs from these representations result in the smallest improvement of all the learned representations used as input when compared to MFCCs. The reason for this could be that the CAE-RNN uses the same top-down information as the frame-level CAE and so the combination is perhaps not complementary. This reason can be confirmed in future work by using other frame-level representation learning methods that uses the same top-down information as the frame-level CAE [35].

Both the CPC and APC models learn representation by being trained to predict future frames, this encourages the models to encode shared temporal information into the learned representations, like higher level phoneme information. More specifically, the CPC model learns representations by being trained to predict the correct sequence of future frames from a set containing negative examples from the same speaker. This encourages the CPC representations to additionally only encode information that is discriminatory between frames of the same speaker, therefore speaker information will be disregarded. The APC model uses an autoregressive function to predict future frames and in contrast to the CPC model, it is not encouraged to disregard any information. Both of these representations used as input to the CAE-RNN lead to improvement in AP compared to using MFCCs as input, with the CPC representations leading to the highest score. Interestingly, using DTW directly on both of these representation types lead to worse performance than their CAE-RNN AWE counter parts. This is despite the DTW approach having full access to the representations, indicating that the CAE-RNN used together with these representations that have been trained in a bottom-up way is complementary.

For the CAPC model, we use the same architecture as that of the APC model, but we use the input-output aligned frames that was used with the frame-level CAE. We hope that by combining these two different mechanisms, we can gain the benefit of encoding shared temporal information as well as disregard speaker and noise information. In our experiments we find that on the English data, these representations as input to the CAE-RNN does outperform both the frame-level CAE and APC representations used as

input, but not on the Xitsonga data set. This indicates that the CAPC model is perhaps more sensitive to validation tuning than the APC model.

Taking together all the results from the different learned representations, we find that AWEs from all four different learned representations outperform those from MFCCs, with CPC representations consistently achieving the best results on English and Xitsonga data. Compared to the CAE-RNN trained on MFCCs, AWEs trained on CPC representations showed a 18.02 % relative improvement on the English data and a 81.75 % relative improvement on the Xitsonga dataset. We also performed a crosslingual experiment, where we train the speech representation models on the larger English dataset and then use these trained models to encode Xitsonga representations. This results in even better quality AWEs, compared to the Xitsonga only method, with the CPC AWEs having a 85.57 % relative improvement compared to the MFCC AWEs. Additionally, we also performed a number of probing tasks on both the different AWEs and frame-level representations directly.

In Chapter 5 we investigated different training strategies based on the variance of speakers in a training batch. Inspired by the fact that human infants are first only exposed to speech from a limited number of speakers which gradually increases, we use a curriculum learning strategy [41] where the level of difficulty is determined by the number of different speakers in the batch. This strategy is compared to three other strategies, training on multiple speakers, on a single speaker and with a reverse curriculum learning strategy. These strategies are applied in different combinations to both the AE-RNN and CAE-RNN. We find that the training strategy has very little to no impact on the quality of the AWEs as long as there are multiple speakers at some stage of the training. Additionally, in an analysis we find that input from many speakers in the early epochs lead to faster convergence of the AP score.

In summary, we conclude that normalising out some of the speaker and gender information contained in AWEs leads to at least marginal improvement in AWE quality. Using self-supervised frame-level speech representations can lead to significant improvement in AWE quality and different training strategies do not contribute towards improved AWE quality.

6.2. Future work

There are a number of areas that should be focussed on in future work, this includes extensions to our presented approaches for improving AWEs and other ventures.

In Chapter 3 we found that conditioning the AE-RNN and CAE-RNN on speaker and gender embeddings leads to some improvement in AWE quality. These are only two types of nuisance properties and future work could consider looking into other types like for example: age, accent, method of recording and level of noise. Previous studies have had

success with training acoustic models to be noise invariant [102] and in a multilingual study, language conditioning has been used for the CAE-RNN [24].

In Chapter 4 we performed a multilingual experiment where representation learning models were trained on the larger English dataset and then used to encode Xitsonga representations which was used as input to the CAE-RNN. This resulted in AWEs with higher AP scores as those from models trained on Xitsonga-only representations. We focused on a zero-resource setting where a only a limited amount of data is available in this thesis and future work should consider doing this same multilingual experiment with a much larger initial training set for the representation models.

In Chapter 5 we took inspiration from human infant speech learning for a curriculum learning training strategy based on the number of speakers. However, infants are usually exposed to many hours of speech [133] and other studies that have tried to mimic infant learning in acoustic modelling have taken this into account [131, 134]. Therefore future work should consider implementing our same experiments on a much larger corpus to confirm if the curriculum learning strategy does not have an impact on the quality of AWEs.

In an analysis study on the current evaluation metrics used for AWEs, including the AP score, Algayres et al. [23] found that these current metric scores do not necessarily correspond to how the AWEs will fair in downstream tasks. Therefore, future work should include using our AWEs in downstream tasks like query-by-example search and full speech segmentation. Additionally, future research should focus on coming up with evaluation metrics for AWEs that are task independent.

Although we focussed on the AE-RNN and CAE-RNN, none of our proposed methods are specific to these models (except speaker and gender conditioning which is dependent on an encoder-decoder architecture). Future work should include applying these methods to other AWE models to see if they have the same effects, like the Siamese RNN [12, 48] and the correspondence variational autoencoder [22].

Another area that should be focused on to improve unsupervised AWEs is the method used to retrieve the similar pairs of speech segment used as input-output pairs for the CAE-RNN. In this thesis and other work [13, 35, 37, 66] a UTD system [55] is used to discover similar segments of speech in an unsupervised way. This system has to be finely tuned for each dataset, which means it would be a laboursome task to use this system for the many other zero-resource languages. Also, the CAE-RNN trained on supervised word pairs achieves much better results compared to training on UTD pairs. Therefore, if we can find an approach to retrieve better pairs of similar speech segments, the quality of the AWEs from the CAE-RNN will be improved.

Bibliography

[1] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proc. Interspeech*, 2017.

[2] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans., Audio, Speech, Language Process.*, vol. 16, no. 1, pp. 186–197, 2008.

[3] Y. Yuan, C.-C. Leung, L. Xie, H. Chen, B. Ma, and H. Li, "Learning acoustic word embeddings with temporal context for query-by-example speech search," in *Proc. Interspeech*, 2018.

[4] Y. Hu, S. Settle, and K. Livescu, "Acoustic span embeddings for multilingual query-by-example search," in *Proc. SLT*, 2021.

[5] H. Kamper, K. Livescu, and S. Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," *Proc. ASRU*, 2017.

[6] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Comput. Speech Lang.*, vol. 46, pp. 154–174, 2017.

[7] H. Kamper, A. Jansen, and S. Goldwater, "Unsupervised word segmentation and lexicon discovery using acoustic word embeddings," *IEEE Trans., Audio, Speech, Language Process.*, 2016.

[8] L. Rabiner, A. Rosenberg, and S. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Trans., Audio, Speech, Language Process.*, vol. 26, no. 6, pp. 575–582, 1978.

[9] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. ASRU*, 2013.

[10] S. Bengio and G. Heigold, "Word embeddings for speech recognition," in *Proc. Interspeech*, 2014.

[11] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio Word2Vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Proc. Interspeech*, 2016.

[12] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. ICASSP*, 2016.

[13] H. Kamper, "Truly Unsupervised Acoustic Word Embeddings Using Weak Top-down Constraints in Encoder-decoder Models," in *Proc. ICASSP*, 2019.

[14] Y.-A. Chung and J. Glass, "Speech2Vec: A sequence-to-sequence framework for learning word embeddings from speech," in *Proc. Interspeech*, 2018.

[15] N. Holzenberger, M. Du, J. Karadayi, R. Riad, and E. Dupoux, "Learning Word Embeddings: Unsupervised Methods for Fixed-size Representations of Variable-length Speech Segments," in *Proc. Interspeech*, 2018.

[16] D. M. Eberhard, G. F. Simons, and C. D. Fennig, "Ethnologue: Languages of the World," https://www.ethnologue.com/, 2021.

[17] K. Levin, A. Jansen, and B. Van Durme, "Segmental acoustic indexing for zero resource keyword search," in *Proc. ICASSP*, South Brisbane, Queensland, Australia, 2015, pp. 5828–5832.

[18] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. Rose, M. Seltzer, P. Clark, I. McGraw, B. Varadarajan, E. Bennett, B. Borschinger, J. Chiu, E. Dunbar, A. Fourtassi, D. Harwath, C. Lee, K. Levin, A. Norouzian, V. Peddinti, R. Richardson, T. Schatz, and S. Thomas, "A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition," in *Proc. ICASSP*, 2013.

[19] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results," in *Proc. SLTU*, 2016.

[20] E. Dunbar, X. N. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The zero resource speech challenge 2017," in *Proc. ASRU*, 2017.

[21] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2020: Discovering discrete subword and word units," in *Proc. Interspeech*, 2020.

[22] P. Peng, H. Kamper, and K. Livescu, "A correspondence variational autoencoder for unsupervised acoustic word embeddings," in *Proc. NeurIPS-SAP*, 2020.

[23] R. Algayres, M. Zaiem, B. Sagot, and E. Dupoux, "Evaluating the reliability of acoustic speech embeddings," in *Proc. Interspeech*, 2020.

[24] H. Kamper, Y. Matusevych, and S. Goldwater, "Improved acoustic word embeddings for zero-resource languages using multilingual transfer," *arXiv preprint arXiv:2006.02295*, 2020.

[25] H. Kamper, Y. Matusevych, and S. Goldwater, "Multilingual acoustic word embedding models for processing zero-resource languages," in *Proc. ICASSP*, 2020.

[26] C. Jacobs, H. Kamper, and Y. Matusevych, "Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation," in *Proc. SLT*, 2021.

[27] O. Räsänen, "Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions," *Speech Communication*, vol. 54, no. 9, pp. 975–997, 2012.

[28] M. Elsner and C. Shain, "Speech segmentation with a neural encoder model of working memory," in *Proc. EMNLP*, 2017.

[29] Y. Matusevych, H. Kamper, and S. Goldwater, "Analysing autoencoder-based acoustic word embeddings," in *Proc. BAICS*, 2020.

[30] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. Interspeech*, 2011.

[31] C. R. Pernet and P. Belin, "The role of pitch and timbre in voice gender categorization," *Front. Psychol.*, vol. 3, 2012.

[32] M. J. Sjerps, N. P. Fox, K. Johnson, and E. F. Chang, "Speaker-normalized sound representations in the human auditory cortex," *Nature Communications*, vol. 10, no. 1, pp. 2465, 2019.

[33] L. Badino, A. Mereta, and L. Rosasco, "Discovering discrete subword units with binarized autoencoders and hidden-Markov-model encoders," in *Proc. Interspeech*, 2015, p. 5.

[34] T. Tsuchiya, N. Tawara, T. Ogawa, and T. Kobayashi, "Speaker invariant feature extraction for zero-resource languages with adversarial learning," in *Proc. ICASSP*, 2018.

[35] P.-J. Last, H. A. Engelbrecht, and H. Kamper, "Unsupervised Feature Learning for Speech Using Correspondence and Siamese Networks," *IEEE Trans., Audio, Speech, Language Process.*, vol. 27, pp. 421–425, 2020.

[36] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the ZeroSpeech 2020 challenge," in *Proc. Interspeech*, 2020.

[37] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. ICASSP*, 2015.

[38] Y.-A. Chung and J. Glass, *Improved Speech Representations with Multi-Target Autoregressive Predictive Coding*, 2020.

[39] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[40] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. R. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. Interspeech*, 2019.

[41] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009.

[42] V. Avramova, *Curriculum Learning with Deep Convolutional Neural Networks*, Ph.D. thesis, KTH Royal Institute of Technology, 2015.

[43] K. J. Geras and C. Sutton, "Scheduled denoising autoencoders," in *Proc. ICLR*, 2015.

[44] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," in *Proc. ICML*, 2019.

[45] R. Lotfian and C. Busso, "Curriculum learning for speech emotion recognition from crowdsourced labels," *IEEE Trans. Audio Speech Lang. Process.*, p. 1, 2018.

[46] C. Wang, Y. Wu, S. Liu, M. Zhou, and Z. Yang, "Curriculum Pre-training for End-to-End Speech Translation," in *Proc. AMACL*, Online, 2020, pp. 3728–3738.

[47] S. Braun, D. Neil, and S.-C. Liu, "A curriculum learning method for improved noise robustness in automatic speech recognition," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Kos, Greece, 2017, pp. 548–552.

[48] S. Settle and K. Livescu, "Discriminative acoustic word embeddings: Recurrent neural network-based approaches," in *Proc. SLT*, 2016.

[49] L. van Staden and H. Kamper, "Improving unsupervised acoustic word embeddings using speaker and gender information," in *SAUPEC/RobMech/PRASA Conference*, 2020.

[50] L. van Staden and H. Kamper, "A comparison of self-supervised speech representations as input features for unsupervised acoustic word embeddings," in *Proc. SLT*, 2021.

[51] G. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 5, pp. 530–549, 2003.

[52] J. Bourgain, "On lipschitz embedding of finite metric spaces in Hilbert space," *Israel J. Math.*, vol. 52, no. 1, pp. 46–52, 1985.

[53] M. Belkin and P. Niyogi, "Laplacian eigenmaps for fimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[54] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[55] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011.

[56] I. Szoke, F. Metze, L. Javier, J. Proenca, A. Buzo, M. Lojka, X. Anguera, and X. Xiong, "Query by example search on speech at Mediaeval 2015," in *In Proc. CEUR Workshop*, 2015, p. 3.

[57] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009.

[58] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU*, 2009.

[59] Y. Yuan, C. Leung, L. Xie, H. Chen, and B. Ma, "Query-by-example speech search using recurrent neural acoustic word embeddings with temporal context," *IEEE Access*, vol. 7, pp. 67656–67665, 2019.

[60] W. Shen, C. M. White, and T. J. Hazen, "A Comparison of Query-by-Example methods for spoken term detection," 2009, p. 4.

[61] P. Li, J. Liang, and B. Xu, "A Novel Instance Matching Based Unsupervised Keyword Spotting System," in *Proc. ICICIC*, 2007, pp. 550–550.

[62] Y. Zhang and J. Glass, "A Piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping," in *Proc. Interspeech*, 2011, p. 4.

[63] A. Jansen and B. Durme, "Indexing raw acoustic features for scalable zero resource search," *13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012*, vol. 3, pp. 2465–2468, 2012.

[64] G. Mantena and X. Anguera, "Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 8515–8519.

[65] M.-L. Sung, *Unsupervised Spoken Term Discovery on Untranscribed Speech*, Ph.D. thesis, The Chinese University of Hong Kong, 2020.

[66] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 8091–8095.

[67] C.-T. Chung, C.-a. Chan, and L. Lee, "Unsupervised discovery of linguistic structure including two-level acoustic patterns using three cascaded stages of iterative optimization," in *Proc. ICASSP*, 2013.

[68] C.-y. Lee, T. O'Donnell, and J. Glass, "Unsupervised Lexicon Discovery from Acoustic Input," *Transactions of the Association for Computational Linguistics*, vol. 3, 2015.

[69] O. Walter, T. Korthals, R. Haeb-Umbach, and B. Raj, "A hierarchical system for word discovery exploiting DTW-based initialization," in *Proc. ASRU*, 2013.

[70] M. A. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, "The Buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability," *Speech Communication*, vol. 45, no. 1, pp. 89–95, 2005.

[71] N. Vries, M. Davel, J. Badenhorst, W. Basson, E. Barnard, and A. de Waal, "A smartphone-based ASR data collection tool for under-resourced languages," *Speech Communication*, vol. 56, pp. 119–131, 2014.

[72] T. Schultz, N. T. Vu, and T. Schlippe, "GlobalPhone: A multilingual text & speech database in 20 languages," in *Proc. ICASSP*, 2013.

[73] P. Ladefoged and D. E. Broadbent, "Information Conveyed by Vowels," *The Journal of the Acoustical Society of America*, vol. 29, no. 1, pp. 98–104, 1957.

[74] C. Tang, L. S. Hamilton, and E. F. Chang, "Intonational speech prosody encoding in the human auditory cortex," *Science*, vol. 357, no. 6353, pp. 797–801, 2017.

[75] J.-L. Gauvain and Chin-Hui Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans., Audio, Speech, Language Process.*, vol. 2, no. 2, pp. 291–298, 1994.

[76] S. Ahadi and P. Woodland, "Combined Bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 11, no. 3, pp. 187–206, 1997.

[77] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[78] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[79] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *EUROSPEECH*, 1995.

[80] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. Mori, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," in *Proc. ICASSP*, 2006.

[81] M. Siniscalchi, J. Li, and C.-H. Lee, "Hermitian based hidden activation functions for adaptation of hybrid HMM/ANN models.," in *Proc. Interspeech*, 2012.

[82] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans., Audio, Speech, Language Process.*, vol. 3, no. 5, pp. 357–366, 1995.

[83] L. Lee and R. Rose, "Speaker normalization using efficient frequency warping procedures," in *Proc. ICASSP*, 1996.

[84] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 7942–7946.

[85] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU*, 2013.

[86] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast Adaptation of Deep Neural Network Based on Discriminant Codes for Speech Recognition," *IEEE Trans., Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1713–1725, 2014.

[87] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B.-H. Juang, "Speaker-Invariant training via adversarial learning," in *Proc. ICASSP*, 2018.

[88] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans., Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[89] J. Grzybowska and M. Ziółko, "I-vectors in gender recognition from telephone speech," in *Proc. NCAMBM*, 2015, p. 7.

[90] M. Wang, Y. Chen, Z. Tang, and E. Zhang, "I-vector based speaker gender recognition," in *Proc. IAEAC*, 2015, pp. 729–732.

[91] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, p. 6.

[92] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[93] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Proc. ASRU*, 2015.

[94] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, Florence, Italy, 2014, pp. 4052–4056.

[95] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," in *Proc. Interspeech*, 2017.

[96] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015.

[97] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017.

[98] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the Information Encoded in X-Vectors," in *Proc. ASRU*, 2019.

[99] J. Schmidhuber, "Learning Factorial Codes by Predictability Minimization," *Neural Computation*, vol. 4, no. 6, pp. 863–879, 1992.

[100] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, p. 9.

[101] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015.

[102] D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, and Y. Bengio, "Invariant representations for noisy speech recognition," *ArXiv*, vol. abs/1612.01928, 2016.

[103] T. Kocmi and O. Bojar, "An Exploration of word embedding initialization in deep-learning tasks," in *arXiv:1711.09160 [Cs]*, 2017.

[104] A. Kanagasundaram, S. Sridharan, G. Sriram, S. Prachi, and C. Fookes, "A study of x-vector based speaker recognition on short utterances," in *Proc. Interspeech*, 2019, pp. 2943–2947.

[105] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[106] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans., Audio, Speech, Language Process.*, vol. 28, no. 4, pp. 357–366, 1980.

[107] W. He, W. Wang, and K. Livescu, "Multi-view Recurrent Neural Acoustic Word Embeddings," *arXiv:1611.04496 [cs]*, 2016.

[108] M. Versteegh, R. Thiollière, T. Schatz, X.-N. Cao Kam, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *Proc. Interspeech*, 2015.

[109] J. L. McClelland and D. E. Rumelhart, "An interactive activation model of context effects in letter perception: I. An account of basic findings," *Psychological Review*, vol. 88, no. 5, pp. 375–407, 1981.

[110] D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater, "A comparison of neural network methods for unsupervised representation learning on the Zero Resource Speech Challenge," in *Proc. Interspeech*, 2015, p. 5.

[111] R. Thiolliere, E. Dunbar, G. Synnaeve, M. Versteegh, and E. Dupoux, "A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling," p. 5.

[112] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, "Joint Learning of Speaker and Phonetic Similarities with Siamese Networks," in *Proc. Interspeech*, 2016.

[113] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[114] K. Friston, "The free-energy principle: A unified brain theory?," *Nat Rev Neurosci*, vol. 11, no. 2, pp. 127–138, 2010.

[115] L. de-Wit, B. Machilsen, and T. Putzeys, "Predictive Coding and the Neural Response to Predictable Stimuli," *J Neurosci*, vol. 30, no. 26, pp. 8702–8703, 2010.

[116] J. Makhoul, "Linear prediction: A tutorial review," in *Proc. IEEE*, 1975, vol. 63, pp. 561–580.

[117] D. O'Shaughnessy, "Linear predictive coding," *IEEE Potentials*, vol. 7, no. 1, pp. 29–32, 1988.

[118] M. Gutmann and A. Hyvarinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," p. 8.

[119] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector Ssace," in *Proc. ICLR*, 2013.

[120] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, 2019.

[121] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *Proc. ICASSP*, 2020.

[122] M. A. C. Blandón and O. Räsänen, "Analysis of predictive coding models for phonemic representation learning in small datasets," in *Proc. ICML Workshop Self-Supervision in Audio and Speech*, 2020.

[123] M. Peters, M. Neumann, L. Zettlemoyer, and W.-t. Yih, "Dissecting Contextual Word Embeddings: Architecture and Representation," in *Proc. EMNLP*, 2018.

[124] Y.-A. Chung, Y. Belinkov, and J. Glass, "Similarity Analysis of Self-Supervised Speech Representations," *arXiv:2010.11481 [cs, eess]*, 2020.

[125] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proc. ICML*, 2007.

[126] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in *Proc. ICML*, 2012.

[127] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735–80, 1997.

[128] A. Conneau, A. Baevski, R. Collobert, and M. Auli, *Unsupervised Cross-Lingual Representation Learning for Speech Recognition*, 2020.

[129] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *Proc. ICASSP*, 2020.

[130] C. Bergmann and A. Cristia, "Environmental Influences on Infants' Native Vowel Discrimination: The Case of Talker Number in Daily Life," *Infancy*, vol. 23, no. 4, pp. 484–501, 2018.

[131] T. Schatz, N. H. Feldman, S. Goldwater, X.-N. Cao, and E. Dupoux, "Early phonetic learning without phonetic categories: Insights from large-scale simulations on realistic input," in *Proc. NAS*, 2020.

[132] J. F. Werker and R. C. Tees, "Cross-language speech perception: Evidence for perceptual reorganization during the first year of life," *Infant Behavior and Development*, vol. 7, no. 1, pp. 49–63, 1984.

[133] E. Bergelson, M. Casillas, M. Soderstrom, A. Seidl, A. Warlaumont, and A. Amatuni, "What Do North American Babies Hear? A large-scale cross-corpus analysis.," *Developmental science*, 2019.

[134] R. Li, T. Schatz, Y. Matusevych, S. Goldwater, and N. H. Feldman, "Input matters in the modeling of early phonetic learning," in *Proc. ACCSS*, 2020.

[135] S. Lev-Ari, "The influence of social network size on speech perception," *Quarterly Journal of Experimental Psychology*, vol. 71, no. 10, pp. 2249–2260, 2018.

[136] G. C. Rost and B. McMurray, "Finding the signal by adding noise: The role of noncontrastive phonetic variability in early word learning," *Infancy*, vol. 15, no. 6, 2010.

[137] M. Sumner, "The role of variation in the perception of accented speech," *Cognition*, vol. 119, no. 1, pp. 131–136, 2011.

[138] B. F. Skinner, "Reinforcement today," *American Psychologist*, vol. 13, no. 3, pp. 94–99, 1958.

[139] G. B. Peterson, "A day of great illumination: B. F. Skinner's discovery of shaping," *Journal of the Experimental Analysis of Behavior*, vol. 82, no. 3, pp. 317–328, 2004.

[140] J. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, pp. 71–99, 1993.

[141] K. A. Krueger and P. Dayan, "Flexible shaping: How learning in small steps helps," *Cognition*, vol. 110, no. 3, pp. 380–394, 2009.

[142] D. L. Rohde and D. C. Plaut, "Language acquisition in the absence of explicit negative evidence: How important is starting small?," *Cognition*, vol. 72, no. 1, pp. 67–109, 1999.

[143] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. ICML*, 2008.