Deep Learning for Prominence Detection in Children's Read Speech

Submitted in partial fulfillment of the requirements
of the degree of

Dual Degree (B.Tech + M.Tech)

by

Mithilesh Vaidya
(Roll No. 17D0700011)

Supervisor:
Prof. Preeti Rao

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
October 2021

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: 17/10/2021

Mithilesh Vaidya

Roll No. 17D070011

Abstract

Expressive reading, considered the defining attribute of oral reading fluency, comprises the prosodic realization of phrasing and prominence. The detection of perceived prominence in speech can serve as a proxy for gauging the speaker's comprehension of the text in an oral reading assessment task. Hand-engineered acoustic features extracted at the word level from a set of suprasegmental attributes such as pitch and intensity contours have been popular in previous works. We first review the performance of a previously proposed Random Forest classifier operating on a compact set of hand-crafted acoustic features, which is considered as the baseline for this work. With the advent of deep learning, automatic feature learning from acoustic contours has shown promising results. We summarise the key findings of our prior work on experimenting with a previously-proposed convolution neural network. A CNN is trained on low-level acoustic contours while word-level context is explicitly incorporated by augmenting each frame with a positional encoding. Though this model involves minimal hand-engineering, it fails to surpass the performance of hand-crafted acoustic features presented in the baseline work, despite various enhancements to the architecture such as a sequence model for modelling temporal dependencies. We then present an end-to-end deep learning pipeline which operates directly on segmented speech waveforms to learn acoustic features relevant to prominent word detection. In the chosen Convolutional Recurrent Neural Network (CRNN) framework, the CNN module extracts acoustic features from waveform segments at word level while the RNN module exploits utterance-level dependencies. The CNN module is further found to benefit from the replacement of unconstrained convolution filters at the input with perceptually motivated Sinc filters. The linguistic association between the prosodic events of phrase boundary and prominence can be further exploited by multi-task learning. By sharing the Sinc filters and conditioning the prominence branch on phrase boundary prediction, a noticeable improvement in performance is observed, thereby exceeding the previously reported performance on the same dataset of a random forest ensemble predictor trained on carefully chosen hand-crafted acoustic

features. We also present a justification for the improvement in performance due to Sinc filters by visualising their frequency response and comparing it with the standard unconstrained convolution layer. We further evaluate the possibly complementary information between acoustic features and pre-trained word embeddings. A drastic improvement in performance confirms the effectiveness of such non-acoustic features. Finally, the performance of the model as a function of various word segment properties is discussed. Based on an analysis of these results and existing literature, we propose a number of promising future directions which can be pursued.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Apart from the phonemic content, extensive para-linguistic information is conveyed in human speech. This information can convey additional meaning, help interpret nuanced differences in meaning or convey emotion. Conflict estimation [3], emotion recognition [4, 5, 6] and detection of various disorders [7, 8, 9] are some of the tasks which can be carried out by analysing these para-linguistic features. They manifest themselves via changes in various acoustic properties such as pitch, rhythm and intensity.

The prosodic structure of speech carries important information in terms of the syntax and the meaning, both of which are critical to a listener's ease of comprehension of the spoken message [10, 11, 12]. Phrase boundaries embed sentence syntax through word grouping while prominence or emphasis on specific words signals new information or highlights a contrast.

In this work, we focus specifically on prominence prediction in children's read speech. Before we proceed, it is important to distinguish between two forms of stress: word stress (also called pitch accent) and sentence stress (referred to as prominence in this work). Word stress is defined for every word in English and occurs on one syllable, as indicated in the dictionary using the accent symbol ‘. For example, the word 'dictionary' has a dictionary pronunciation of *dìk-sh-ner-ē*. The word stress for this word is always expected for the syllable 'dik'. It is

also called pitch accent because in English, the stress is acoustically realised by modifying the pitch of the stressed syllable. Other languages may utilise other forms of acoustic manipulation to stress the syllable. Sentence stress, on the other hand, is heavily influenced by the frequency of the word. More specifically, if the word imparts any new information or contradicts existing information, the speaker is expected to stress on the word and make it stand out in it's local context via changes in the suprasegmental attributes such as duration, F0, intensity and spectral shape[13]. Local context includes not only the phones in the word but also several neighbouring words. Hence, the word 'dictionary' may or may not be sentence stressed as it depends on the context in which it is used. The two are not independent: when a speaker decides to make a specific word prominent, the expected syllable to be stressed in that word (as mentioned in the dictionary) must be chosen to carry the sentence prominence. Therefore, knowledge of word stress (i.e. which syllable should be stressed in a given word) is important for a good speaker.

Prominence detection has applications in scoring of spoken language fluency [14, 15] and text-to-speech synthesis [16]. Moreover, we plan to use the prominence predictions as an auxiliary input for a model tasked with end-to-end speech scoring which assigns a fluency score to the entire utterance.

Traditionally, various aggregates of the sampled acoustic parameters (e.g. pitch and intensity) across the word segment including mean and variance, contour shape descriptors, and differences in these quantities across neighbouring words are computed. These are referred to as 'word-level prosodic features' [17, 18] which are then used to train a conventional supervised classifier, possibly in combination with lexico-syntactic information [19, 20, 21]. In the baseline work [1], a large set of features were computed across the distinct suprasegmental attributes of speech and Recursive Feature Elimination with Cross Validation (RFECV) was employed to derive a compact set of interpretable features for speaker-independent boundary and prominence detection on a children's oral reading dataset. It was observed that apart from the expected pitch, duration and intensity based aggregates, the acoustic cues to prominence included a number of spectral shape functionals while the phrase boundary prediction was dominated by pause-based features.

Hand-engineered features require extensive domain-knowledge. Moreover, they can miss out on important patterns in data since the space of such features is very large. To overcome these shortcomings, end-to-end deep learning models operate on minimally-processed input

and learn all task-related features through large datasets. Thanks to an explosion in compute capabilities and size of datasets, such models have dominated the landscape for the last decade. A brief summary of deep learning models for prominence detection is discussed next.

Rosenberg et al. [22] used a large number of acoustic-prosodic features and aggregates at word level derived from their previous AuToBI work [17, 23]. A bidirectional RNN classifier was used to model the word sequence context as opposed to that explicitly provided in the feature vector. They observed a small improvement ($< 1\%$ absolute) in boundary and pitch accent detection over a baseline conditional random forest classifier. Wu et al. [24] also used similar aggregated acoustic features with an LSTM to find an improvement over the use of an SVM classifier. Lin et al. [25] used a hierarchical BLSTM network to aggregate features across phone, syllable and word to model contextual information at multiple scales in the joint detection of boundaries and prominence.

Motivated by the demonstrated potential of convolutional neural networks to learn discriminative patterns and replace any feature engineering, recent research has focused on extracting CNN based feature representations from low-level acoustic-prosodic contours to obtain the word-level detection of pitch accents in an utterance [26, 27]. Stehwien et. al. [27, 28] used a CNN on sampled acoustic parameters (energy, F0, loudness, voicing probability, zero crossing rate and harmonic-to-noise ratio), together with a context window of two neighbouring words to optimally learn the word-level aggregated features. With word position indicators provided in the input segment, they report an improvement of 1-3% absolute over Rosenberg [17] on lexical stress and phrase boundary detection on the BURNC corpus. On the spectrum of hand-engineered features at one end and models operating on raw waveform at the other, the above model lies somewhere in the middle. It, however, suffers from the same issues mentioned above, albeit to a lesser degree as compared to the RF classifier operating on word-level aggregates.

Both local acoustic features and longer, more global contexts spanning several words and possibly different sentences across the utterance are important in the perception of prominence. Hence, architectures combining low-level feature aggregation with sequence models were realized with the same contour-learned features input to an LSTM classification layer [26, 29]. Attention mechanism can help attend to various character embeddings in a BLSTM-CRF model [30].

As mentioned previously, feature learning via end-to-end neural network systems trained

on speech waveforms is being increasingly viewed as the optimal approach to complex classification tasks [3, 31]. Such systems have achieved performances close to, but not always exceeding, those of classifiers with task-specific hand-crafted features. This is because such models are prone to overfitting due to the large number of parameters. This hints at a need for the introduction of reasonable constraints, or additional domain knowledge in the network architectures, especially in the widely encountered data-constrained scenarios. This trade-off between allowing the model to learn everything from scratch and incorporating domain knowledge is an important consideration in today's deep learning landscape.

In this work, we explore precisely such variations for the prominence detection task from segmented speech waveforms starting from a straightforward CRNN model. This is the first case of prosodic event detection from speech waveforms that we are aware of. The first variation involves replacing the CNN layer at the input with tunable bandpass filters called Sinc filters [31]. These are motivated by the traditionally used Mel Filterbanks which emulate low-level auditory processing. SincNet has been applied in frame-level speaker identification where its hyperparameters have been found to be critical, although sometimes counter-intuitive, to achieved task accuracy [32].

Next, we try to exploit the linguistic association between phrase boundaries and prominent words with multi-task learning. The presence of phrase-finality increases the perceived prominence of the word [33, 34] and can potentially contribute to the feature representation for prominence. Recent work on the joint prediction of boundary and prominence is promising. In [25], the boundary predictions were computed from the final layer output of a 3-layer BLSTM network while prominence predictions are made at the penultimate layer. In another attempt, prosodic event classification is viewed as a 4-class problem [35]. Building on the the above theme, we explore various multi-task architectures for prominence detection that incorporate information about the (typically more reliably predicted) phrase boundary status of the word.

We also examine the possibly complementary information present in the hand-crafted acoustic features. Finally, given the importance of lexical information in prominence detection [36, 37], we report the performance of the combination of acoustic features and pre-trained word embeddings.

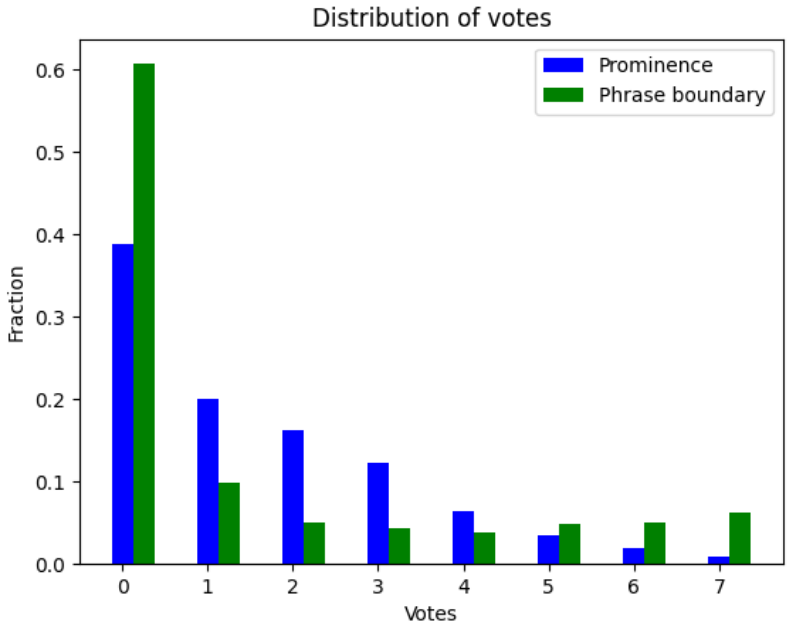


Figure 1.1: Distribution of votes for prominence and phrase boundary. The dataset contains a total of 41,286 words across 790 utterances by 35 speakers, recorded at 16 kHz sampling rate. Each word is labelled for both prominence and boundary by 7 naive listeners.

1.1 Dataset

The children's oral reading dataset used in this work comprises recordings of grade-appropriate text read aloud by selected middle school students with reasonable word decoding ability in English (as second language) but widely varying levels of prosodic skill [1]. The individual utterances are story paragraphs comprising between 50-70 words, each word labelled separately for the presence/absence of prominence and phrase boundary by 7 naive listeners using the RPT methodology [38]. This is reduced to a net rating per word based on number of votes (out of 7), further scaled down to the range 0-1, to obtain the 'degree' of prominence (boundary) per word. The utterances are available segmented at word level by forced alignment with the manual transcript.

Z-score normalisation is applied to each audio recording. Moreover, all hand-crafted acoustic features are also z-score normalised, which is a standard practice in machine learning to tackle features on different scales.

The distribution of votes is given in Figure 1.1. This may be compared with data sets of native English speech with perceived phrase breaks reported to be 14.4% and prominent words to be 26.1% of the total number of words [39]. If we consider prominence labelling based on 3 or more votes, we obtain a comparable figure of 25% words considered prominent in our dataset. Since the inter-rater agreement for phrase boundary is higher than that for prominence, thresholding on 4 or more votes gives 19.9% words as phrase boundaries.

1.2 Our Contributions

- The first end-to-end deep learning model for the task of prominence detection which operates on waveforms segmented at word level.
- Demonstrate a performance improvement by replacing the unconstrained 1D convolution at the input with a Sinc convolution in the context of prosodic event detection.
- Analysis of various multi-task learning paradigms for joint prediction of prominence and phrase boundary detection.

1.3 Publication

The work presented in this report has been submitted as:
Mithilesh Vaidya, Kamini Sabu and Preeti Rao, "Deep Learning for Prominence Detection in Children's Read Speech"
at the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Chapter 2

Word-level Features

2.1 Hand-crafted acoustic features

For the task of prominence detection, 34 hand-crafted features (A34) were proposed in the baseline [1]. These were obtained after applying RFECV to a total of 2524 features, consisting of aggregates of pitch, energy, intensity and spectral band energies computed across the word segment and varying neighbourhoods of upto 2 words on each side and speaker normalised versions of the same. They also include some AuToBi baseline features [23]. A short description of the chosen set of 34 features can be found in Table 2.1.

Similar to A34, which have been optimised for the task of prominence detection, A27 refers to 27 features extracted using RFECV for the task of phrase boundary detection in [1]. These are described in Table 2.2

Sr. No.	Feature Set (No. of features)	Description
1.	F0 (8)	min, max, span of F0 contour in context; Differences of max and span with previous word
	F0 contour shape (3)	Slope of line fitted to semitone F0 contour of word; std of instantaneous slope contour across word; Similarity of word F0 contour with valley contour
2.	Intensity (6)	mean, span of intensity in context; Differences of mean and span with previous word; Slope of line fitted to energy contour across context window
3.	Spectral Shape (12)	Mean for bands: 0 to 500Hz and 400Hz to 2kHz; Span across temporal context for intensity in bands: 2kHz to 5kHz, 5kHz to 8kHz; Slope of band 400Hz to 2kHz intensity contour across word; Difference of specific band energy between current and next word: (bands 0 to 500 Hz, 60 Hz to 400 Hz, 1kHz to 2kHz, 2kHz to 4kHz, 5kHz to 8kHz)
4.	Duration (5)	Max-min of SN vowel durations in the word normalised by local tempo in pause-separated window; Max vowel duration*; Max syllable duration; Difference of max syllable duration with that of previous word*; Max-min difference for phone duration with that of previous word*; Note: * are Speaker Normalised by phone rate

Table 2.1: 34 hand-crafted features optimal for prominence prediction obtained from RFECV and a Random Forest classifier. The *context* mentioned in the table could be any of the 8 word neighbourhoods. Refer to Section 4.2.2 and Table 6 in [1] for more details.

Sr. No.	Feature Set (No. of features)	Description
1.	F0 (4)	Min, mean of F0 contour; Differences of above with previous word
	F0 contour shape (6)	Slope of line fitted to F0 contour of word; Difference of F0 slop with next word; Word-level max of SN instantaneous F0 slope contour Difference of slope mean between current and next word Similarity of word F0 contour with rising and falling contour
2.	Intensity (6)	Min, span, slope of intensity in context; Differences of above with previous word;
3.	Spectral Shape (5)	Word-level span for energy in band 5kHz to 8kHz; Difference of band energy for 1kHz to 2kHz between current and next word; Difference in across-word fitted slope of the same between current and next word; Word-level min of spectral tilt
4.	Duration (3)	Max syllable duration SN by articulation rate; Mean phone duration context normalised by phone rate; Difference in longest vowel duration within a word with previous word (both SN by articulation rate)
5.	Paurse (3)	Difference between SN post-word pauses of current and next word; Difference between pre-word and post-word pauses; SN version of above

Table 2.2: 27 hand-crafted features optimal for phrase boundary prediction obtained from RFECV and a Random Forest classifier. *SN* denotes speaker-normalised version of the feature. The *context* mentioned in the table could be any of the 8 word neighbourhoods. Refer to Section 4.2.2 and Table 4 in [1] for more details.

2.2 Lexical features

Prominence is linked to the text syntax and semantics with content words such as proper nouns expected to receive prominence most of the time, followed by adjectives, nouns, adverbs, and verbs respectively [40]. To exploit this dependency, each word in the canonical text is categorised into one of 12 part-of-speech (PoS) tags. For words which are inserted by the speaker erroneously, an additional binary variable is set to 1. In addition to these 13 features, the number of phones and number of syllables in the word are also considered. This gives a total of 15 features, together termed as PoS+. Moreover, based on the top-down expectations of the listener, each word is labelled as either *prominent*, *optional* or *not prominent* (similarly for phrase boundary). This gives a total of 6 binary features per word. These are referred to as information structure or IS.

The field of natural language processing has witnessed tremendous progress in the last decade due to large powerful models trained on huge corpora [41, 42, 43, 44]. A multitude of tasks can be solved by freezing the feature extraction layers of such models and adding classification layers on top, which are trained for the task at hand. Numerous word embeddings have been developed over the years, which capture semantic similarities in high-dimensional space. Lexical features in the form of word embeddings can implicitly capture parts-of-speech information. For our task, we test two popular word embeddings: GloVe [45] and BERT [46]

Their usage in the context of prosodic event detection has been explored in [37]. In this work, we extract 100-dimensional GloVe embeddings pre-trained on Wikipedia using the gensim package. For BERT, we use the *bert-base-uncased* pre-trained model available in the *HuggingFaces* library. The 768-dimensional BERT embedding is extracted from the Encoder of the Transformer model.

The embedding is passed through a dropout layer followed by a linear layer whose dimension is a hyperparameter [37]. The intuition behind this step is to ensure that the lexical and acoustic features lie in similar spaces (in terms of dimensionality and abstractness) before concatenation. After tuning, we found that a dropout layer of probability 0.3 and a fully-connected layer of dimension 300 gave the best performance.

Chapter 3

Acoustic Contours

Before moving on to waveform-based CNN models, we explored the architecture proposed by Stehwien et. al. [27] in more depth since it proposes to combine the best of both worlds: domain knowledge is incorporated by extracting contours such as F0 while deep learning's potential is exploited by training a CNN to detect patterns in such contours. We briefly review our previous work [2] in this chapter.

3.1 Feature sets

The word-level features in the baseline work [1] were computed from word (and sub-word) aligned contours corresponding to the time-varying acoustic parameters of F0, intensity and spectral shape, computed at 10 ms intervals. We wish to investigate CNN-based automatic learning of word-level features from the same low-level acoustic contours. A brief description of the contours is given in Table 3.1.

Sr. No.	Feature Set (No. of contours)	Contour description
1.	Pitch (4)	Pitch in Hz and semitones; z-score normalised versions of both
2.	Intenity (4)	Energy and Intensity; z-score normalised versions of both
3.	Spectral shape (7)	HNR; Spectral Tilt; Sonorant Band (300-2300 Hz) Energies in the bands: 60–400 Hz, 400–2000 Hz, 2000–5000 Hz, 5000–8000 Hz

Table 3.1: 15 low-level acoustic contours which are fed to a CNN for feature extraction, as discussed in Section 3.1.

3.2 CRNN

A CNN is tasked with feature extraction from acoustic contours. We have a matrix of acoustic features for each utterance (call it *feat*). It's dimensions are T x 15 where T is the total number of frames in the utterance. From the word boundaries (obtained using forced alignment), we extract a slice for each word by also including it's immediate neighbourhood in the feature matrix i.e. if s_k and e_k denote the start and end frames respectively of the k^{th} word in the utterance, the following slice is extracted for the i^{th} word: $feat[s_{i-1} : e_{i+1}]$. Note that pauses between words are also captured in this feature slice.

By including the neighbourhood, the CNN can detect patterns during word transitions, which can be especially helpful for modelling pauses. However, CNNs are invariant to translation (by design) and hence cannot distinguish between frames of the current word and it's neighbours. This can be tackled by concatenating a positional encoding to each frame. Extending the 1-bit encoding proposed in [27], we concatenate a 3-bit encoding which is 001 for the frames of the previous word, 010 for the current and 100 for the following word (Figure 3.1). With separate encodings for the previous and the following word, the CNN can further distinguish between pre-word pauses and post-word pauses, which would not be possible with the

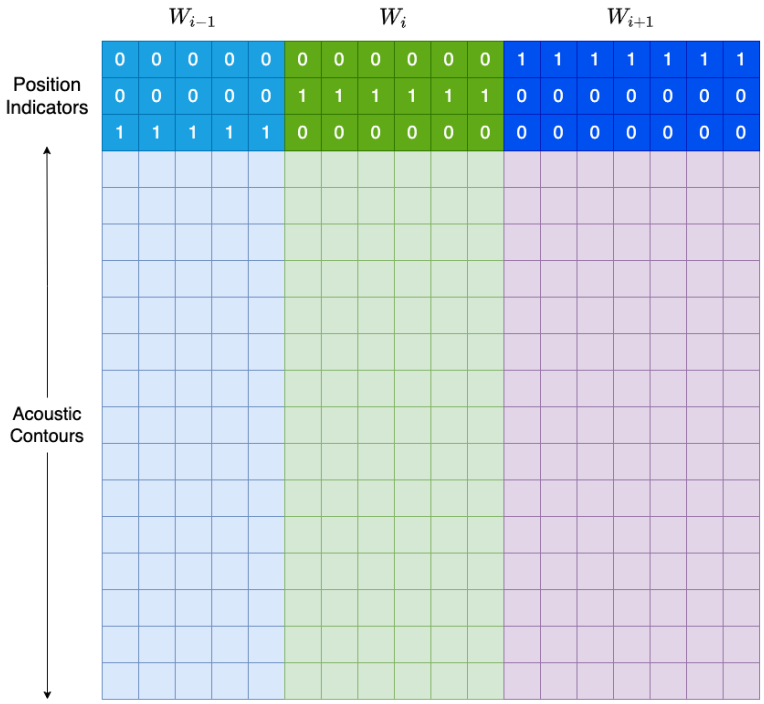


Figure 3.1: Final feature matrix for the i^{th} word after augmenting the contours with a 3-bit positional encoding.

1-bit encoding in [27].

Given the previously observed speaker-dependence of the relative importances of the different prosodic attributes, we also experiment with an architecture which has separate CNN filter banks for each feature set. The final output is concatenated to obtain the final embedding. The 1D convolution output of each of the CNN filters is max-pooled across time to get a scalar value per filter per filter bank. We experiment with various kernel sizes to capture sub-phone, phone, syllable and word-level context. Each filter bank has N filters, each with k kernel sizes, resulting in a kN-dimensional feature encoding for each feature group corresponding to a word.

The CNN embeddings for each word are fed to a sequence classifier (such as GRU), which can dynamically model utterance-level dependencies as compared to the static word neighbourhoods involved in the computation of hand-crafted features. The final architecture is depicted in Figure 3.2.

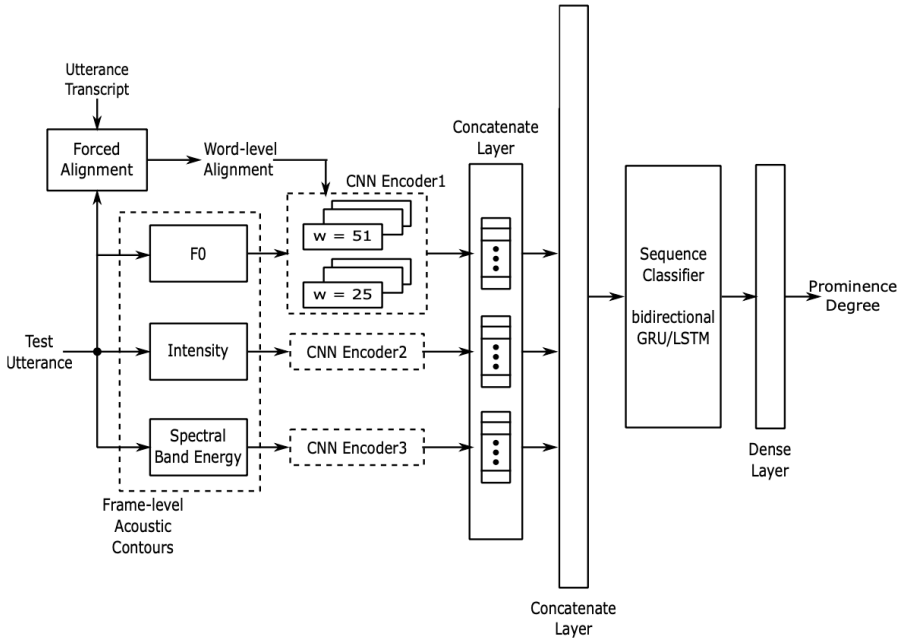


Figure 3.2: The proposed CRNN framework for extracting features from acoustic contours. Figure reproduced from [2]. Note that the CNN depicted in the figure has separate filter banks for each feature set.

3.3 Results

The results for the first set of experiments are presented in Table 3.2. We replaced the Random Forest classifier with sequence models such as LSTM and GRU (and their bidirectional equivalents) and varied the number of layers and hidden units. A 2 layer, 256-dimensional bidirectional LSTM gave a modest improvement of 0.02 (absolute) in Pearson correlation over the baseline Random Forest classifier using 34 acoustic hand-crafted features (discussed in detail in

Table 3.2: Performance of various sequence models with 34 hand-crafted acoustic features. (* indicates sd < 0.01)

Model	# layers	# hidden units	Pearson correlation
RFC	-	-	0.69*
GRU	2	96	0.68
LSTM	2	256	0.69
BGRU	2	96	0.70
BLSTM	2	256	0.71*

Table 3.3: Performance of a single CNN operating on all feature groups as a function of different sets of kernel widths, with the corresponding literature reference indicated. (* indicates s.d. $<$ 0.01)

Kernel widths	Pearson correlation
3, 5, 11, 25, 51	0.67
5, 11, 25, 51 [47]	0.67
25, 51	0.67*
11, 25, 51	0.67*
11 [26]	0.65

Section 2.1), thereby proving the importance of modelling utterance-level dependencies instead of treating each word independently.

Next, we tune the CNN architecture by varying the CNN kernel widths for capturing patterns in the contours at various time scales. Results are presented in Table 3.3. We observe that the syllable and word width kernel sizes (25 and 51 frames or 250 and 510 ms respectively) helps the performance while including other widths does not change it. Hence, we finalise these widths. The number of filters of each kernel size is set to 8 since additional filters did not give any improvement in performance.

On training a separate filter bank for each feature set instead of a single CNN, the performance further jumped to 0.69. Note that this is the best Pearson correlation obtained so far using acoustic contours. In summary, a sequence model (BLSTM) which accepts hand-crafted acoustic features attains a Pearson correlation of 0.71, which is still higher than that obtained after moving from hand-crafted features to acoustic contour-based learning. This gap motivated us to go one level deeper and explore the possibility of extracting features tailor-made for the task of prominence detection from waveform in an end-to-end fashion.

This page was intentionally left blank.

Chapter 4

Waveform-based Models

Replacing manual feature extraction with models which can automatically learn relevant features from data has been the main promise of deep learning. A brief survey of the above trend in the context of feature extraction from waveforms is discussed next.

Among the earliest works, [48] (2013) used a conventional CNN pipeline consisting of 1D convolution, max-pooling and Tanh activation for the task of phoneme recognition on TIMIT. The classification accuracy for a CNN trained on raw waveform was only 2% less (absolute) than a CNN trained on MFCCs (a CRF decoder is used in both cases). [49] (2015) improved upon the architecture with various initialisation schemes for the CNN weights, a different non-linearity for training stability purposes and LSTM layers for better sequence modelling. CNN kernels of width 35 ms were able to match the performance of Log-Mel filterbanks for Large Vocabulary Continuous Speech Recognition (LVCSR). However, these models were trained on 2000 hours of data, pointing towards the need for large datasets in order to approach (and surpass) the performance of hand-engineered features. [50] (2016) demonstrated a superior model for the task of emotion recognition (and summarise their findings with a rather amusing title). A convolutional recurrent neural network (CRNN) operating on waveform was found to outperform various handcrafted feature sets such as ComParE [51] and eGeMAPS [52]. WaveNet [53]

(2016), an autoregressive model, was a major leap in the field of raw audio generation. Causal dilated convolutions for an exponential increase in the receptive field, residual skip connections for improved gradient flow and gated activation units demonstrated superior performance on diverse tasks such as text-to-speech, music generation and speech recognition. For speaker verification, [54] (2018) extracted a fixed-dimensional embedding from raw waveform using 9 CNN layers, followed by an LSTM for exploiting temporal dependencies. A learnable 2-parameter pre-emphasis layer (tuned with a lower learning rate) and a strided convolution layer at the input outperformed Mel-Filterbank features. Complex Gabor filters were proposed in [55] (2020) for replacing the usual CNN weight kernels to fully take advantage of its optimal time-frequency resolution. LEarnable Audio Frontend (LEAF), proposed in [56] (2021), combines Gabor filters with learnable pooling and learnable per-channel compression to allow the model to learn all pre-processing parameters from scratch.

Inspired by the success of previous works, we propose a model which takes the ASR-decoded word segments as input and predicts the degree of prominence.

4.1 CNN

Convolution Neural Networks (CNN), which were originally developed for pattern recognition in images, are now ubiquitous in deep learning. The translation-invariant filters are efficient in terms of number of parameters and do an excellent job at extracting patterns from any data.
A typical CNN consists of a number of layers. Each layer consists of:

1. Convolution layer: Responsible for extracting patterns from the input signal. We use 1D convolutions in our case since the waveform is a 1D signal.

2. Batch Normalisation [57]: Important for training stability. Similar to input data normalisation, Batch Normalisation normalises the activations of the hidden layers, thereby reducing the internal covariate shift. This not only improves performance but also leads to training stability.

3. Activation function: Non-linearities are crucial for the extensive expressive power of any neural network. Since Sigmoid and Tanh suffer from vanishing gradients, they have been

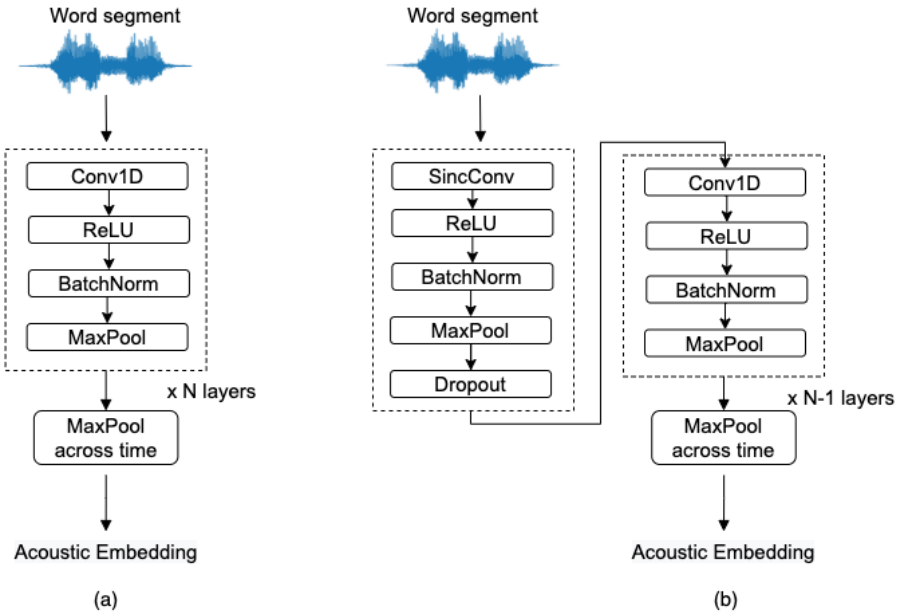


Figure 4.1: CNN architectures: (a) Standard CNN layers (b) Sinc replacing the first unconstrained layer.

replaced by ReLU and other activation functions.

4. Max Pooling: Pooling the activations of the convolution filters has two benefits. Firstly, by aggregating the most important features, the model becomes robust to perturbations and noise. Secondly, by reducing the time dimensionality, the filters in deeper layers can extract patterns at coarser scales.

Many such layers help in the hierarchical extraction of information. The final CNN output is max-pooled across time to get a fixed-dimensional embedding for each word (Figure 4.1(a))

4.2 SincNet

Mel Filterbanks are very popular since they emulate the low-level auditory filtering carried out by our ears. Although they have served the speech community well for decades, the advent of deep learning has questioned the notion of a fixed filter bank for *all* speech tasks. Instead, can we extract task-specific filters motivated by Mel Filterbanks? SincNet [31] is a promising answer to this question. By replacing the input CNN layer with Sinc convolution (SincConv), an improvement was observed for the task of speaker recognition.

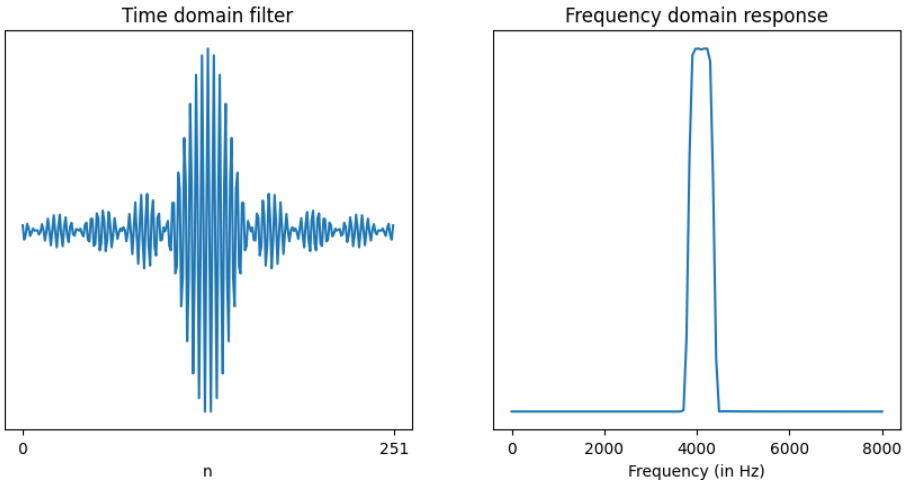


Figure 4.2: Time domain representation and its equivalent frequency domain response of a Sinc filter of size 251 samples. The filter is centred at 4000 Hz with a bandwidth of approximately 500 Hz.

These filters are parameterized by only 2 scalars: the lower and upper cutoff frequencies of the band-pass filter. Both parameters are learned end-to-end via backpropogation. Figure 4.2 depicts a Sinc filter in both time and frequency domain.

They are initialised with the centre frequencies of Mel Filterbanks so as to speed up training. Theoretically, a standard convolution layer, with more parameters, has more expressive power than SincConv. However, these unconstrained filters suffer from overfitting. On the other hand, SincConv generalises well with just two parameters and has an elegant interpretation in terms of its frequency response. In this work, we replace the first unconstrained convolution layer with a Sinc layer, as shown in Figure 4.1(b).

4.3 Sequence Models

Context is important for predicting prominence [58]. In the manual extraction of word-level features, the local neighbourhood is explicitly included in the feature computation stage e.g. in the baseline work, a neighbourhood of upto 2 words (on each side) is considered in computing the aggregates.

In deep learning, the word context that is critical in speech prosody perception can be

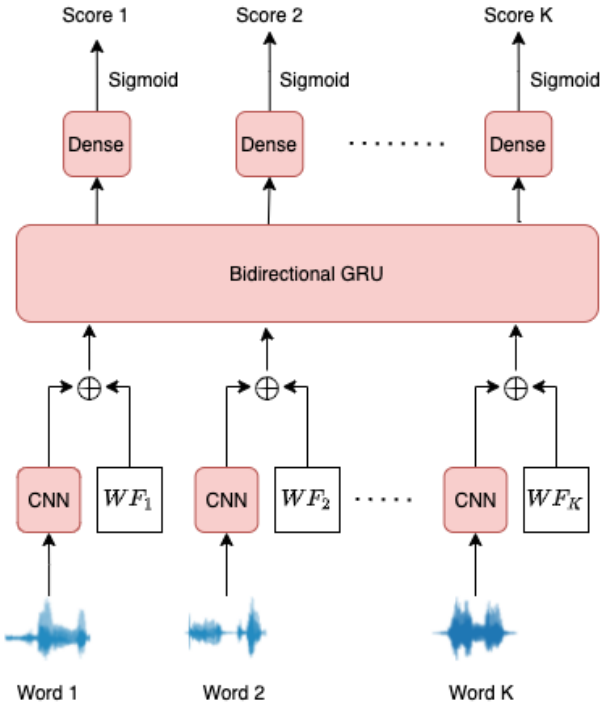


Figure 4.3: Overall CRNN architecture. The CNN block refers to any one of the two proposed architectures in Figure 4.1. WF_t denotes word-level features such as hand-crafted acoustic or lexical features. $\oplus$ denotes concatenation.

modelled by sequence models. The relevant local context is learned from the sequence of words spanning the entire utterance. Owing to their popularity in various sequence tasks, we use a Gated Recurrent Unit (GRU) [59] in our architecture. At each time step, the GRU takes as input a fixed-dimensional vector corresponding to each word. This could be the CNN embedding extracted from the word segment, hand-crafted word-level features or a concatenation of the two. Since the entire utterance is available offline, a bidirectional GRU can further benefit from both past and future context.

The output of the GRU at each time step is fed to a dense network consisting of two fully-connected (FC) layers. The first FC layer, consisting of 128 neurons, is followed by ReLU activation and a dropout layer. The second FC layer has one neuron, which is followed by sigmoid activation to get the final score.

The final CRNN architecture is depicted in Figure 4.3.

This page was intentionally left blank.

Chapter 5

Multi-task Learning

Studies have shown that the presence of phrase-finality increases the perceived prominence of the word [33, 34]. This linguistic association between phrase boundaries and prominent words can be exploited using a multi-task learning (MTL) framework. An extensive survey on the motivation behind MTL and its variants can be found in [60, 61].

Large deep learning models are prone to overfitting. In some cases, in addition to the labels of the task at hand, we also have access to labels of a related task. In such cases, we can benefit from improved generalisation by training the model on an additional auxiliary task. This can be thought of as implicit regularisation. We are aware that prominence and phrase boundary are related tasks and hence the two can potentially benefit from shared representations. Such sharing suppresses the noise present in the dataset by reducing overfitting since the same parameters must learn useful representation for both tasks.

Recent work incorporating both prominence and phrase boundary predictions is reviewed next. In [25], the boundary predictions are computed from the final layer output of a 3-layer BLSTM network while prominence predictions are made at the penultimate layer. This difference could be attributed to the longer context which is typically required for the task of phrase boundary prediction. In another attempt, prosodic event classification is viewed as a 4-class

problem [35].

Based on the MTL variants proposed in [62], we try out various combinations of feature sharing and conditioning.

5.1 Parameter Sharing

In parameter sharing, a carefully chosen section of the architecture is shared among the two tasks. In most deep learning models, the layers close to the input are responsible for feature extraction while those at the output carry out classification. For example, in object detection, the first few CNN layers are responsible for low-level processing such as edge detection while latter layers combine these edges to detect more complicated patterns.

In our case, the CNN is responsible for extracting acoustic information (such as pitch, duration and energy) from waveform. It is reasonable to hypothesise that these low-level features are then combined by the subsequent GRU for predicting prominence or phrase boundary. Hence, we explore the benefit of sharing either the Sinc layer or both Sinc and the CNN layers.

5.2 Conditional MTL

In conditional MTL, the prediction(s) of the auxiliary task(s) is supplied as input to the main task. This is particularly helpful when the final prediction of the auxiliary task serves as a strong signal for the prediction of the main task.

An important consideration in these architectures is the location where the auxiliary prediction is fed as input to the main branch. For our architecture, the phrase boundary prediction can either be concatenated before the dense layer (with output of GRU) or before the GRU (with the CNN embedding). Ideally, feeding it as early as possible should benefit all subsequent layers. Hence, we supply the phrase boundary prediction at the GRU stage of the prominence detection branch. This is because the CNN is operating on 1-dimensional waveforms while the GRU accepts a fixed-dimensional embedding, which is an utterance-level feature which ideally

encapsulates all acoustic properties relevant to the task of prominence detection.

We then explore the possibility of both conditioning and parameter sharing in a hybrid architecture. Lastly, we study the performance of an 'oracle' model, in which the prominence branch has access to the ground truth boundary labels. All MTL architectures are depicted in Figure 5.1.

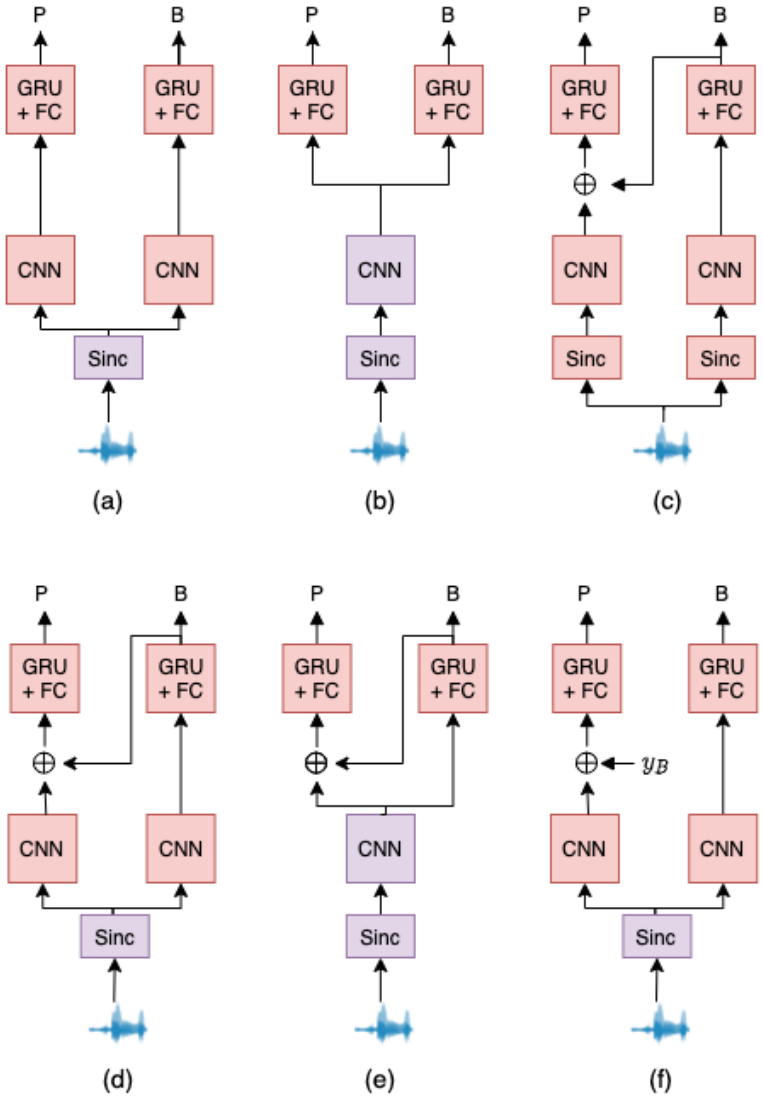


Figure 5.1: MTL architectures: (a) Shared Sinc layers, followed by separate CNNs and GRU heads for prediction of prominence (P) and boundary (B). (b) Both Sinc and CNN layers are shared. (c) Conditioned MTL with separate CNNs and GRUs for P and B predictions. (d) Shared Sinc combined with conditioning by boundary prediction. (e) Shared Sinc and CNN combined with conditioning. (f) The oracle version of (d), where y_B denotes the ground truth label for boundary. $\oplus$ denotes concatenation.

Chapter 6

Experimental Setup

In this chapter, we discuss the experimental setup. This includes the train-test split, testing procedure using ensembling, loss function used for training and various hyperparameter considerations.

6.1 Methodology

Since the dataset is limited, we use cross-validation style testing instead of a fixed train-val-test split to report performance. Following the procedure in the baseline work [1], the dataset is split into 3 equal folds with no speaker overlap. We carry out 3-fold testing by training and validating models on 2 of the folds and testing these models on the remaining fold. For training and validating on the 2 folds, we further split the 2 train folds into 4 sub-folds. This is done to ensure that the model gets to see 75% of the data during training and 25% during validation (as opposed to 50% for both train and validation if we do not split the two folds and train just 2 models). The 4 trained models are then used for inference and their predictions are averaged to generate results on the corresponding unseen test set. Refer to Table 6.1 for a

Model	Train	Val	Test
1a	1, 2, 3	4	5, 6
1b	2, 3, 4	1	5, 6
1c	3, 4, 1	2	5, 6
1d	4, 1, 2	3	5, 6
2a	3, 4, 5	6	1, 2
2b	4, 5, 6	3	1, 2
2c	5, 6, 3	4	1, 2
2d	6, 3, 4	5	1, 2
3a	1, 2, 5	6	3, 4
3b	2, 5, 6	1	3, 4
3c	5, 6, 1	2	3, 4
3d	6, 1, 2	5	3, 4

Table 6.1: Breakdown of sub-folds used for training, validation and testing.

detailed breakdown of the folds used for training, validation and testing. Increasing the number of folds might result in a more reliable cross-validation result (low standard deviation) but time taken for each experiment will also increase. Since the standard deviation for all waveform-based models was observed to be less than 0.01, the benefit of increasing the number of folds (beyond 3) might turn out to be marginal.

The results for the prominence degree prediction are reported in terms of Pearson correlation between the predicted output and the degree of prominence from the RPT rater votes which serves as ground truth. Although one can obtain a binary prediction by thresholding and compute the F-score by also thresholding the ground truth votes (e.g. >2 votes is labelled as prominent), Pearson correlation is preferred because our task involves the prediction of *degree* of prominence.

We use ensembling for reducing model bias: the predictions of 1a, 1b, 1c and 1d are averaged to get the final prediction for the test folds 5 and 6. Similarly, the models 2a, 2b, 2c, 2d and 3a, 3b, 3c, 3d give predictions for test folds 1, 2 and 3, 4 respectively. The mean and standard deviation of these three Pearson correlation values are finally reported.

6.2 Loss function

The Mean Squared Error (MSE) between the ground truth and the model prediction (both in range 0-1) is minimised:

$$L_{MSE} = \sum_{n=1}^{B} \sum_{i=1}^{w_n} (\hat{y}_i^n - y_i^n)^2 \tag{6.1}$$

where B is the batch size, w_n refers the number of words in the n^{th} utterance while $\hat{y}_i^n$ and y_i^n are the predicted and ground truth values respectively for the i^{th} word in the n^{th} utterance.

For MTL, the final loss is a convex combination of the prominence MSE loss ($L_{prominence}$) and phrase boundary MSE loss ($L_{boundary}$) i.e.

$$L_{total} = \alpha L_{prominence} + (1 - \alpha) L_{boundary} \tag{6.2}$$

where α is a hyperparameter which controls the trade-off between performance on the main task and the auxiliary task. For the single-task experiments, $\alpha = 1$.

6.3 Hyperparameter settings

The hyperparameters for the CNN model are tuned for performance on the validation sets. Table 6.2 lists the range of hyperparameters which were tried out, along with some intuition behind their choice. We found the optimal configuration to be: 4 CNN layers, each consisting of 32 filters of kernel width 51, stride 1 and max pooling with kernel size of 3.

For the GRU, it was found that a 3-layer, 256-dimensional bidirectional GRU with dropout of 0.5 at each output layer (except the last layer) consistently gave the best performance. The hyperparameters of the GRU and the subsequent FC layers are fixed for all experiments.

For training, Adam [63] optimizer is used with a learning rate of 0.001. Batch size is

Table 6.2: Range of various hyperparameters for tuning.

Hyperparameter	**Min**	**Max**	**Intuition**
Total CNN layers	2	8	Deep networks can extract sophisticated patterns
Number of CNN/Sinc filters	16	128	More filters increases expressive power of CNN at the cost of overfitting
CNN/Sinc Kernel width	7	151	Influences the receptive field
Pooling width	2	4	Temporal resolution reduces with higher pooling but noise is also suppressed
Stride	1	2	Reduces dimensionality and overfitting. Higher pooling preferred over stride since there is loss of information for stride > 1

set to 64 and the model is trained on a single NVIDIA GeForce GTX 1080. Early stopping is used on the validation set with patience set to 12 epochs i.e. if the Pearson correlation does not increase by more than 0.002 for 12 epochs, we stop training and choose the model with the best performance observed so far for testing. The random seeds for *PyTorch*, *Numpy* and the default *Random* library in Python are manually set for reproducibility purposes.

Chapter 7

Results and Discussion

This chapter presents the results of all the experiments carried out in this work. We begin with single-task learning architectures and observe a performance improvement due to Sinc convolution. Next, multi-task architectures and their results are discussed. Finally, we test the effectiveness of lexical information and their complementarity with acoustic features.

7.1 Single-task learning

We first examine the performance of 34 word-level acoustic-prosodic features (A34, discussed in Section 2.1) which are computed on the acoustic contours of pitch, intensity and spectral shape versus time as well as various segmental durations including pauses. These are obtained by a two-stage feature selection procedure, as detailed in [1]. The performance of the baseline system, which uses a Random Forest classifier on A34, is reported in row 1 of Table 7.1. We also test the same A34 set of features with a bidirectional GRU (BGRU) to study whether the implicit 'learning' of word context can further boost performance (row 2). We note that the performance jump from row 1 to row 2 indicates a clear improvement with the BGRU model,

Table 7.1: Performance of different architectures for single-task learning. Standard deviation $<$ 0.01 for all models.

No.	Input	Acoustic model	Layer 1 (type, width, stride)	Pearson correl.
1.	A34	RFC	-	0.696
2.	A34	GRU	-	**0.726**
3.	Wav	CRNN	Standard, 51, 1	0.692
4.	Wav	CRNN	Sinc, 51, 1	0.712
5.	Wav	CRNN	Sinc, 31, 2	**0.721**
6.	A34 and Wav	CRNN	Sinc, 31, 2	0.735

indicating the value of learned context over that explicitly represented within the A34 feature computations.

Next, we evaluate the performance of the CRNN model operating directly on the speech waveform. For CRNN models, the number of filters in all CNN and Sinc layers is 32 while pool size is 3. For layers 2, 3 and 4, the hyperparameters are fixed (as stated in Section 6.3) while Layer 1 variations are reported here in terms of Pearson correlation.

From row 3, we can conclude that the performance of the CNN model is close to that of the RF classifier, which is an encouraging starting point. A rise in the Pearson correlation can be seen with the Sinc layer replacing the (unconstrained) first convolutional layer (while keeping the number of filters, filter width and stride unchanged). A reduction of the Sinc filter widths to 31 samples (a window of approximately 2 ms) gave a further improvement, especially when the stride was concurrently changed to 2 samples from 1 sample. This is consistent with the observations of previous work that smaller filter widths in the Sinc layer are superior in the context of speaker recognition [32]. While this seems counter-intuitive given that auditory filter impulse responses at lower centre frequencies are of duration well over 10 ms, the reduced frequency resolution due to the apparent truncation does not seem to harm the performance. It is encouraging to see that the gap between waveform-learned and the hand-crafted A34 features has been almost bridged with this tuned Sinc version.

Further, to check for any complementary information in the two representations, we con-

Table 7.2: Performance of various multi-task learning architectures and additional features. (Pearson correlation s.d. < 0.01)

No.	MTL variant and additional features	Pearson correl.
1.	Tuned Sinc (without MTL)	0.721
2.	Fig 5.1(a) (shared Sinc)	0.727
3.	Fig 5.1(b) (shared CNN)	0.726
4.	Fig 5.1(c) (conditional MTL)	0.727
5.	Fig 5.1(d) (shared Sinc + conditional MTL)	**0.740**
6.	Fig 5.1(e) (shared CNN + conditional MTL)	0.724
7.	Fig 5.1(f) (oracle)	0.747
8.	MTL Fig 5.1(d) + A27	0.747
9.	MTL Fig 5.1(d) + A34/A27	0.757

catenate the A34 features with the 32-dimensional CNN embedding before feeding it to the BGRU. The resulting performance that exceeds that of either. Ablation studies were carried out by splitting A34 into 4 categories of features: pitch, energy, duration and spectral features. It was observed that no specific contribution can be termed as dominant. This indicates the future potential for better waveform-based feature learning, possibly with a larger training dataset.

7.2 Multi-task learning

The MTL experiments reported in Table 7.2 were carried out after first tuning α. After a preliminary grid search, we found that the best performance across configurations is obtained when α is set to 0.95 in equation 6.2 (after scaling the MSE of each to bring them into the same range).

The best single-task performance is reported in row 1 for reference. We note from row 2 and row 3 that sharing either the Sinc layer or both Sinc and CNN gives a slight improvement over single-task learning. On the other hand, an increase in performance is seen with conditioned MTL when only the Sinc layer is shared while the CNN layers remain task-dependent (row 5). This is consistent with our expectation that the lowest level features extracted from the

input waveform correspond to the basic suprasegmental attributes fundamental to all prosodic event detection. Therefore, the parameters of the constrained Sinc layer benefit from improved geenralisation in the multi-task setup. Neither conditioned MTL without any sharing of parameters (row 4) nor conditioned MTL with a shared Sinc and CNN (row 6) demonstrate any substantial improvement in performance. This indicates that parameter sharing should be done with careful consideration of the task at hand.

In row 7, the performance of the 'oracle' model is reported. Since ground truth labels are fed to the prominence branch (instead of the phrase boundary prediction), we observe the expected jump in performance. The gap between 0.740 and 0.747 can be attributed to imperfect phrase boundary predictions. Hence, the value 0.747 can also be interpreted as an upper bound on the performance of the conditioned MTL architecture with a shared Sinc layer.

In row 8 of Table 7.2, we report performance on the concatenation of the hand-crafted acoustic features (A27) with the generated CNN embeddings at the GRU input. We expect the boundary task to benefit from these features and in turn improve the performance of prominence prediction. The observed improvement (from 0.74 in row 5) confirms the presence of complementary acoustic information which is not captured by the purely waveform-based architecture. Lastly, we also concatenate the A34 features at the GRU stage in the prominence branch of the model. The resulting Pearson correlation of 0.757 is the best acoustic model result we have attained.

7.3 Additional features

The results in Table 7.3 demonstrate the power of standalone lexical features. Both PoS+ (row 1) and IS (row 2) outperform single-task waveform-based models discussed in Table 7.1. Moreover, a further increase in performance is observed on concatenating the two (row 3), implying the presence of complementary information in the two representations. Their performance is close to that of the best acoustic MTL model (row 9 in Table 7.2). Although BERT has replaced GloVe embeddings in a wide variety of NLP tasks and demonstrated good performance for prominence detection [36], it did not give any clear improvement over GloVe in our task (row 4 and row 5). This could be attributed to very simple story texts without semantic ambiguities

Table 7.3: Performance of lexical features.

Sr.	Features + Model	Pearson correl.
1.	PoS+	0.747
2.	IS	0.732
3.	PoS+ and IS	0.752
4.	BERT	0.761
5.	GloVe	0.76
6.	GloVe and PoS+ and IS	0.765
7.	MTL Fig 2(d) and A34/A27 and GloVe	0.813

that may benefit from contextualized embeddings. We note a big jump in performance in the final row of Table 7.3, where the GloVe features are concatenated with the corresponding CNN embedding and hand-crafted features in each branch of the best MTL model, emphasising the importance of lexical features in the task of prominence detection. Although this is at odds with the expectation that beginning readers do not necessarily realize prominence correctly, it supports the important role of the top-down expectations in raters' perceptions.

This page was intentionally left blank.

Chapter 8

Analysis

In this chapter, we examine some properties of the proposed architecture. Firstly, a comparison of Sinc filters and unconstrained convolution filters is presented. We also examine the benefit of sharing the Sinc layers in an MTL architecture. Next, we study the model predictions as a function of various word properties to look for potential trends. Lastly, we study the effect of batch size and learning rate, two important hyperparameters, on the training routine.

8.1 Filter Visualisation and Comparison

The weights of the first CNN/Sinc layer can be interpreted by either simply plotting the kernel (for the time domain representation) or computing a DFT to obtain the equivalent frequency domain response. For comparing Sinc filters with standard convolution filters, we use the single-task learning models from row 3 and row 4 of Table 7.1. For fair comparison, both convolutions have a kernel size of 51 and stride of 1.

To start with, we plot the frequency response of all 32 Sinc filters (Figure 8.1) and 5 unconstrained standard convolution filters (Figure 8.2) (response of more than 5 filters makes it

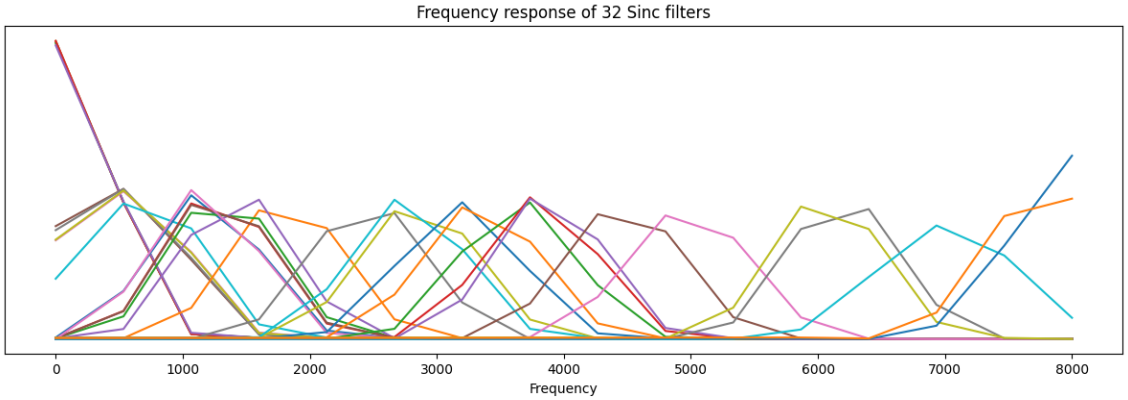


Figure 8.1: Frequency domain response of all 32 Sinc filters. They are not ideal bandpass filters due to truncation of filters in the time domain. Since they are initialised with Mel Filterbank centre frequencies, the density of filters is high at low frequencies while they become sparse as we move towards higher frequencies.

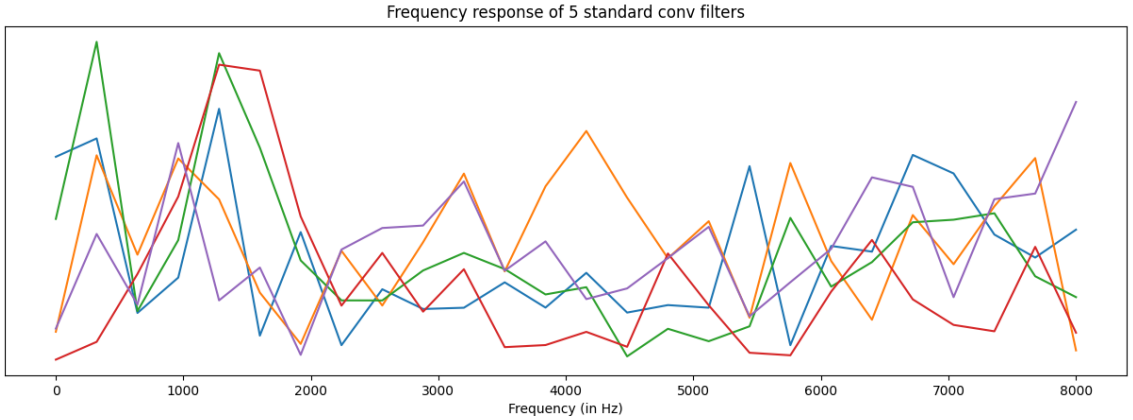


Figure 8.2: Frequency domain response of 5 randomly chosen standard convolution filters. There is no discernible pattern in their frequency response.

difficult to trace any particular filter).

While Sinc filters can be ordered by their centre frequency, no such natural ordering exists for unconstrained convolution layers. Hence, we randomly pick one filter out of the 32 and visualise their time and frequency domain representations in Figure 8.3 and Figure 8.4.

Based on these plots, it is difficult to draw any conclusions based on the representations (both time and frequency) of individual unconstrained convolution filters. Instead, we study their overall behaviour by plotting the cumulative frequency response (CFR) of the learned filters in order to understand the regions of the spectrum which are being selectively modelled.

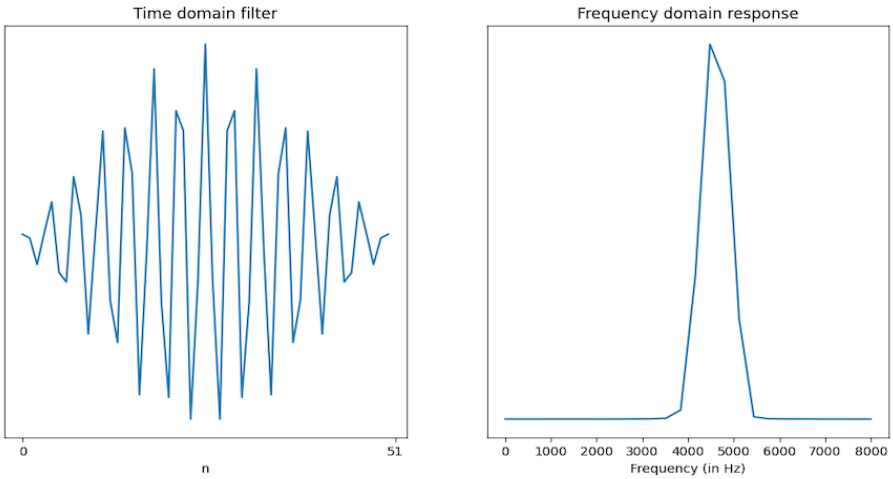


Figure 8.3: Time and frequency domain representation of a randomly chosen Sinc filter. It is centred at around 4158 Hz and has a bandwidth of approximately 568 Hz.

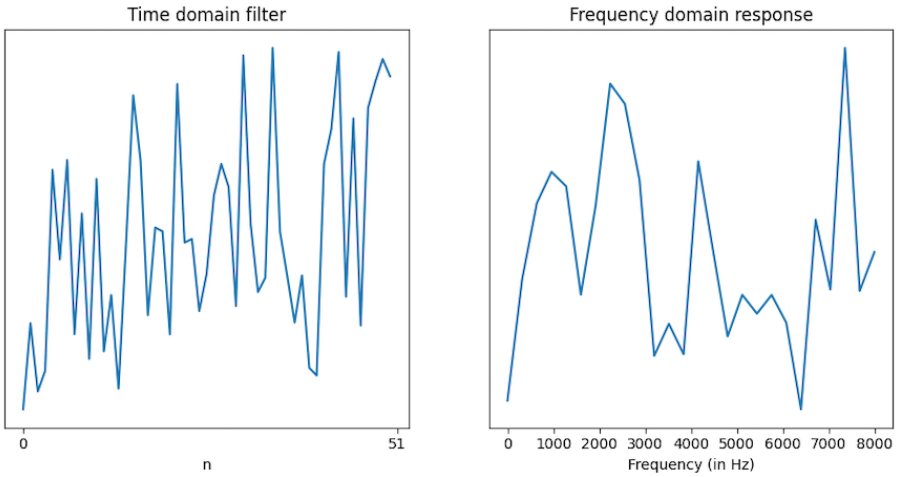


Figure 8.4: Time and frequency domain representation of a randomly chosen standard convolution filter. The plot is not very informative since it is difficult to draw any concrete conclusion from such a response.

We use a procedure similar to [64] for computing the CFR (F_{cum}):

$$F_{cum} = \sum_{k=1}^{N_F} \frac{\mathscr{F}_k}{\|\mathscr{F}_k\|_2} \tag{8.1}$$

where N_F is the total number of filters (32 in all our models) and $\mathscr{F}_k$ is the magnitude spectrum of filter f_k for $k = 1, 2, ..., N_F$.

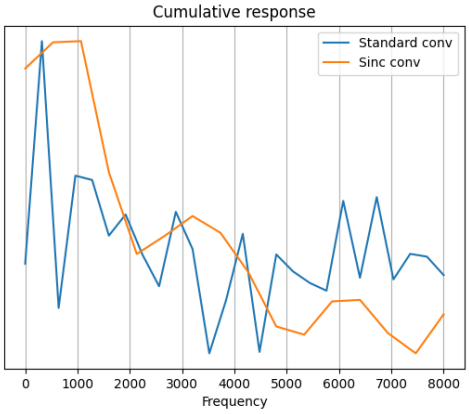
8.1.1 Single-task comparison

The first experiment involves a comparison between the Sinc convolution against the unconstrained standard convolution layer. Similar to the previous experiment, we use models from row 3 and row 4 of Table 7.1. The filter response on test sets 5 and 6 for each of the 4 cross-validation models is given in Figure 8.5.

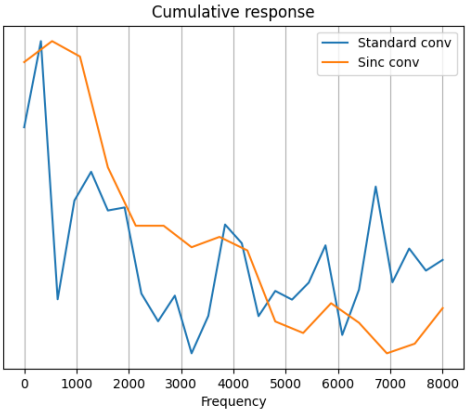
As compared to the the noisy cumulative response of standard 1D convolution, the Sinc filters have a smooth response. We see a strong response in the range (100 - 400 Hz) for both Sinc and standard convolution, which corresponds to pitch. The first formant, around 1100 Hz, is also captured by both models. Thereafter, the frequency response for Sinc drops smoothly (with the exception of a peak between 3000 - 4000 Hz, which corresponds to the second formant). On the other hand, the standard conv response is very noisy. With more parameters, the standard unconstrained convolution layer is more prone to overfitting, which is evident in the plots.

8.1.2 Multi-task comparison

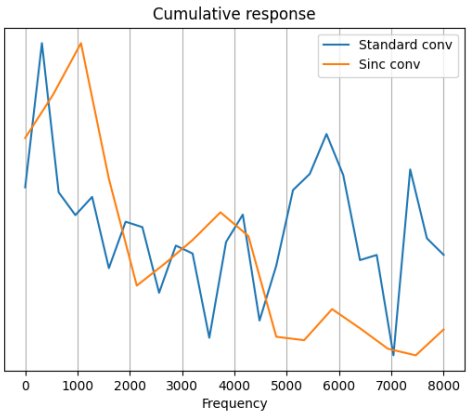
In Section 7.2, we observed an improvement on sharing only the Sinc layer in a conditioned MTL architecture. The cumulative frequency response provides a potential justification. More specifically, we compare the CFR of the Sinc filters in Figure 5.1(c) (conditioned MTL with *separate* Sinc, CNN and GRU for prominence and boundary) vs Figure 5.1(d) (conditioned MTL with *shared* Sinc and separate CNN and GRU for prominence and boundary). The filter



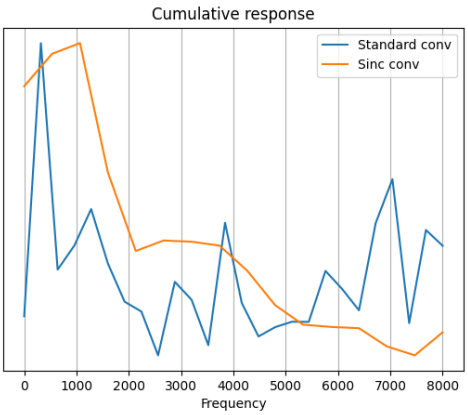
(a) Model 1a from Table 6.1



(b) Model 1b from Table 6.1

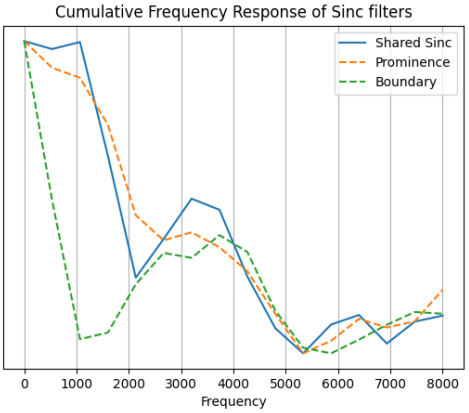


(c) Model 1c from Table 6.1

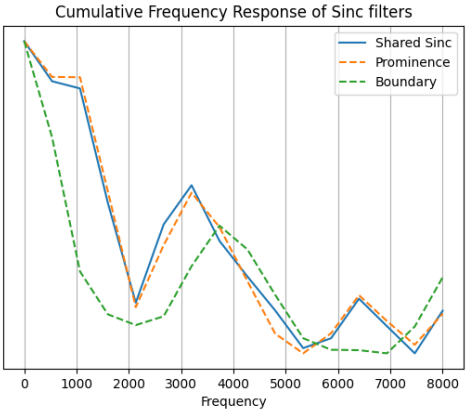


(d) Model 1d from Table 6.1

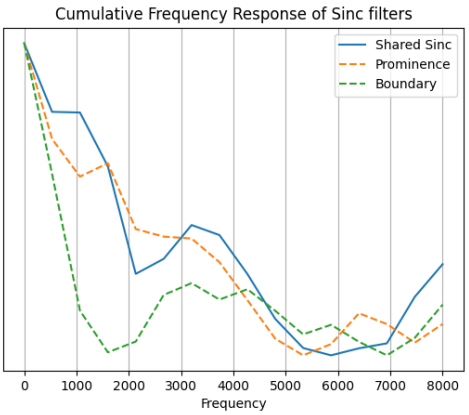
Figure 8.5: Comparison of the cumulative frequency response of Sinc and standard convolution filters of width 51 and stride 1 for single-task learning. (a), (b), (c) and (d) correspond to each of the four models trained using 4-fold cross-validation on sub-folds 1, 2, 3 and 4. The blue contour depicts the CFR of the standard convolution layer while the yellow contour corresponds to the Sinc CFR.



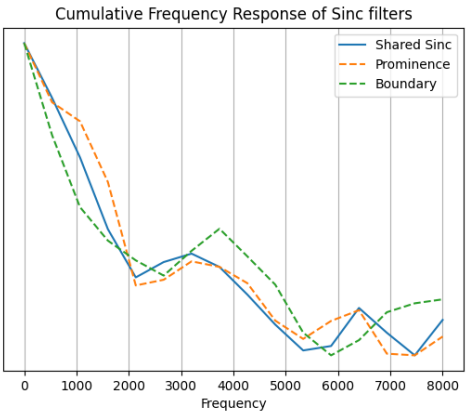
(a) Model 1a from Table 6.1



(b) Model 1b from Table 6.1



(c) Model 1c from Table 6.1



(d) Model 1d from Table 6.1

Figure 8.6: Comparison of the cumulative frequency response of shared Sinc filters and separate Sinc filters for prominence and boundary. (a), (b), (c) and (d) correspond to each of the four models trained using 4-fold cross-validation on sub-folds 1, 2, 3 and 4. The width of Sinc filters in this case is 31 samples (optimal as discussed in Section 7.1). Dashed lines indicate the CFR of Sinc filters which are unique to the two tasks (yellow for prominence, green for phrase boundary) while the solid line is the CFR of shared Sinc filters.

response on test sets 5 and 6 for each of the 4 cross-validation models is given in Figure 8.6. It can be seen that the shared Sinc response (solid blue line) closely follows the response of a Sinc layer trained for just prominence (dashed yellow line). However, the Sinc CFR trained for phrase boundary (dashed green line) lacks a peak in the range 100 - 400 Hz (corresponding to pitch) and around 1100 Hz (corresponding to the first formant). Although it is difficult to explain why such a response was learned through backpropogation in the first place, it is not surprising that the performance of separate Sinc filters is sub-optimal because we know from baseline work that pitch and intensity features are crucial for the task of phrase boundary prediction.

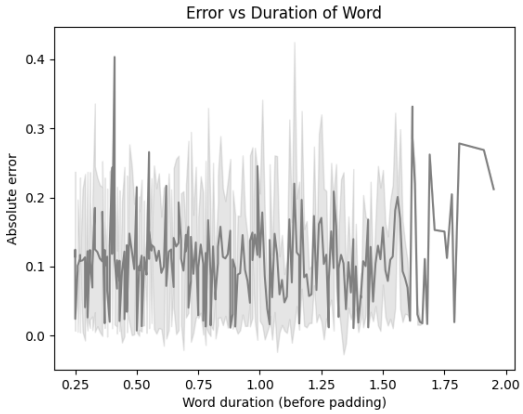
8.2 Model predictions

The model predictions of the best acoustic model in Table 7.2 (row 9) on test folds 5 and 6 are analysed in this section. For each word, the absolute error (L1 norm) between ground truth and prediction is considered as the model prediction error.

In Figure 8.7(a), absolute error as a function of word duration is analysed. There seems to be no discernible global trend: both short and long words have similar error ranges. A slight increase in error is seen as we approach very long words (> 1.7 seconds). A potential explanation could be that the last max-pool layer (for both Sinc and non-Sinc CNNs) results in a loss of temporal information for very long words.

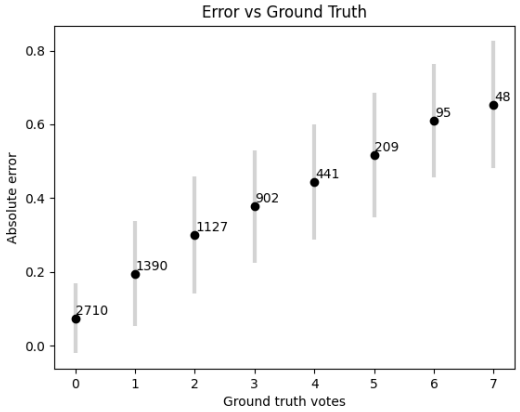
In Figure 8.7(b), we plot error as a function of the location of the word in the utterance. There seems to be a slight upward trend near location 70. There are two possible explanations: either the GRU fails to model long-term dependencies (leading to the possibility of replacing it with more powerful models such as Transformers [65]) or the error is not reliable because few utterances have more than 70 words.

Figure 8.7(c) depicts error as a function of the ground truth. The pattern is clear: error increases with votes. Due to the skewed distribution of ground truth votes, the model primarily focuses on non-prominent words (votes < 2). This could be alleviated by weighting the MSE loss for each word with a weight which is inversely proportional to the count of its ground truth label. This is especially important if we need a strong decision boundary (especially near votes 2 or 3 depending on the inter-rater agreement) so as to threshold the output and output a binary



(a) Prediction error vs Duration

(b) Prediction error vs Location of Word



(c) Prediction error vs Ground Truth

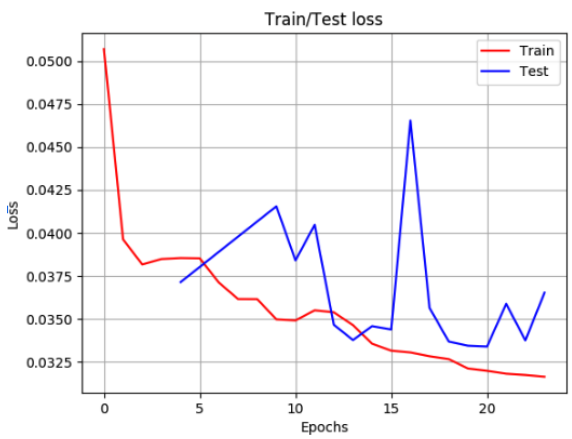
Figure 8.7: Analysis of the model prediction error as a function of various word attributes such as duration (a), location in utterance (b) and ground truth label (c) is presented. In (a) and (b), the dark contour represents the mean error while the light shaded extension depicts the standard deviation. In (c), the black dots correspond to mean error while bars denote standard deviation. The annotation at each point in (c) denotes the count of ground truth labels for each of the votes. The best acoustic model with MTL, A34 and A27 features (row 9 of Table 7.2) is used.

prediction (prominent or not prominent).

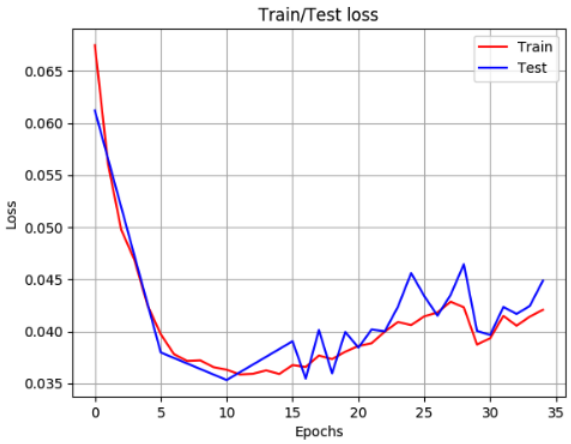
8.3 Loss curves

Learning rate and batch size are empirically-tuned hyperparameters which significantly influence the performance of the model. Low learning rates (and/or higher batch size) slows down training and could prevent the model from escaping local minima on the loss surface. On the other hand, larger rates (and/or lower batch size) can lead to unstable training due to noisy gradients. Presented next are some train-validation loss curves at different learning rates and batch sizes for a single-task learning model without Sinc (row 3 of Table 7.1). These were obtained to stabilise training before further improving the model architecture.

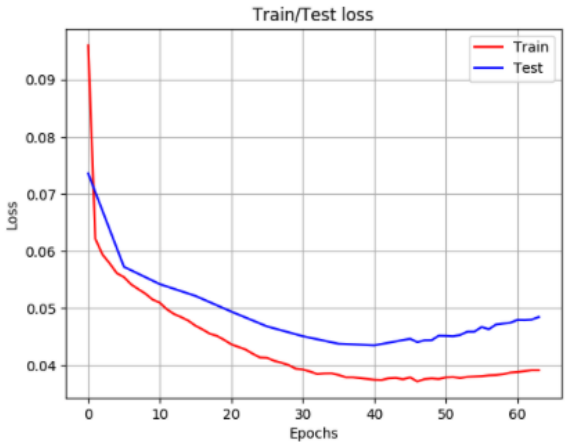
In Figure 8.8(a) and (b), the learning rate is 0.01 while batch size is 8 (each figure corresponds to training on two different folds). It is observed that both training and validation (referred to as test in the figure) curves are very noisy. The batch size is increased by stepping the optimizer after every 64 samples (each batch on the GPU is still limited to 8 samples due to memory constraints). The curve in Figure 8.8(c) is smoother but the loss quickly starts diverging. To counteract this, the learning rate is reduced by a factor of 10 to 0.001. The resulting curves in Figure 8.8(d) are smooth and satisfactory. The validation loss decreases upto a certain point and shows an upward trend, which is an expected sign of overfitting.



(a) LR: 0.01, BS: 8



(b) LR: 0.01, BS: 64



(c) LR: 0.001, BS: 64

Figure 8.8: Train and validation loss curves obtained on varying the learning rate (LR) and batch size (BS) for the single-task learning standard convolution model (row 3 of Table 7.1). In (a), LR is high while BS is low, leading to unstable training. As we increase the batch size (b), the plots become smoother but performance on validation oscillates. On decreasing the learning rate (c), the plots are as desired (smooth drop in validation loss followed by gradual increase due to overfitting).

Chapter 9

Summary

In this work, we first reviewed the baseline which involves extensive hand-engineering for deriving a compact set of acoustic features. We then summarised our findings of tuning a previously proposed CNN model which operates on low-level acoustic contours. Despite the incorporation of a GRU for modelling dependencies across an utterance, finer positional encoding and separate CNN fitlerbanks for distinct feature groups, the model is unable to outperform the hand-crafted acoustic features proposed in the baseline. Inspired by this failure to extract robust features from acoustic contours, we then propose an end-to-end deep learning model operating directly on waveform segments for the task of prominence detection (although the architecture, after some tuning, can be also applied for the task of phrase boundary). Our results indicate that it is challenging to outperform the performance of hand-crafted features computed from across the prior extracted suprasegmental contours of speech essential to prosody realization, at least with moderate sized training datasets. However, it was found that some human audition motivated constraints such as Sinc-based convolution at the feature extraction stage can improve the performance of deep learning models. The optimal hyperparameters for the Sinc filters turned out to correspond to low strides and low widths (high time resolution and low frequency resolution), confirming previous findings in [32].

Backed by research emphasising the linguistic association between prominence and phrase boundary, we also explored the effectiveness of various multi-task learning frameworks. It was found that conditional MTL, in which the prominence prediction is conditioned on the concurrent boundary prediction, along with a shared Sinc layer for feature extraction, gives the best performance and finally outperforms the hand-crafted acoustic features proposed in the baseline. Moreover, on visualising the cumulative frequency response of the learned filters, we can justify the performance benefit of Sinc filters over standard unconstrained convolution filters. Also, the benefit of sharing the Sinc layer in an MTL framework is also evident on observing the cumulative frequency response plots of shared Sinc filters vs separate task-specific Sinc filters.

Finally, we explored the incorporation of lexical features using two NLP embeddings, namely BERT and GloVe, and confirmed the findings that lexical information is complementary and further boosts performance. This indicates that the lexical identity of words guides the top-down expectations of raters, even in the case of not-so-proficient beginning readers. An analysis of the model error as a function of the ground truth votes shows a clear trend: as the degree of prominence increases, the model error increases.

9.1 Future work

Over the course of this work, the following ideas were explored but could not be thoroughly tested due to paucity of time:

- Transfer Learning: Emotion recognition and conflict detection are two related tasks which also operate on suprasegmental features discussed in this work. We trained a CRNN architecture [3] on two datasets: RAVDESS [66] and SSPNet Conflict Corpus [67]. The CNN layers at the input, which ideally extract relevant acoustic features, were then transferred to our task i.e. the weights of prominence prediction CNN were initiliased using these pre-trained CNN models. A number of considerations affect the performance: how many layers should be shared, should the weights of the prominence CNN be frozen or fine-tuned with a lower learning rate, etc. Since datasets with adult speech are abundant, another direction which can be pursued is transfer learning from adult's speech to children's speech [68].

- Data Augmentation: The size and quality of dataset is crucial for any deep learning model. In [34], it was found that changes in the suprasegmental attributes (such as pitch, rhythm and intensity) across the word are more important in perceiving prominence than their absolute values. As a result, pitch shifting in a certain range (e.g. $\pm$50 cents) should not affect the target labels. Similarly, we can also perturb the rate of speech (e.g. 0.9 - 1.1 times the original rate). Such augmentation can be either applied offline or on-the-fly.

In addition to the above ideas, we plan to explore the following in more depth:

- In LEAF [56], every stage in the pre-processing pipeline is parameterised. Apart from Gabor filters (in place of Sinc or unconstrained 1D convolution), the architecture also includes Gaussian low-pass filtering (instead of max-pooling) and per-channel energy normalisation (instead of logarithmic compression). They found an improvement in performance for a wide variety of tasks, ranging from audio classification to emotion recognition.

- More sophisticated CNN architectures can be experimented with for more efficient feature extraction. In particular, deeper networks can be trained by using skip connections, which alleviate the problem of the vanishing gradient [69, 70].

- Transformers [65], which have replaced sequential models in almost every NLP task, are a promising replacement for GRU. Transformers have access to the entire context, as opposed to a GRU which, in practice, has a finite capacity for retaining memory. This can lead to better exploitation of dependencies at the utterance level.

- In this work, acoustic and lexical features were simply concatenated. This might not be optimal since they capture information from different domains. Instead, we can explore attention-based mechanisms for fusing such information which comes from two distinct modalities [71].

- Self-Supervised Learning is a popular paradigm which artificially generates labels for a supervised task using large unlabelled datasets [46]. With access to a vast amount of unlabelled children's speech dataset, we can train autoencoder-style models for extracting prosody embeddings [72, 73].

- By disentangling speaker characteristics (such as timbre) through a separate model [73], the prosodic event detection network can more efficiently extract features relevant to the task of prominence and phrase boundary detection.

- Analyse the performance of various architectures as a function of the dataset size. Since manual rating of prosodic events is time-consuming and expensive, models which can operate on minimal labelled data can be more easily deployed. In particular, it would be interesting to note the effect of transfer learning and data augmentation on performance for low-resource settings.

References

[1] Kamini Sabu and Preeti Rao. Prosodic event detection in children's read speech. *Computer Speech and Language*, 68:1–19, 2021.

[2] Kamini Sabu, Mithilesh Vaidya, and Preeti Rao. Deep learning for prominence detection in children's read speech, 2021.

[3] Vandana Rajan, Alessio Brutti, and Andrea Cavallaro. Conflictnet: End-to-end learning for speech-based conflict intensity estimation. *IEEE Signal Processing Letters*, 26(11):1668–1672, 2019.

[4] Danqing Luo, Yuexian Zou, and Dongyan Huang. Investigation on joint representation learning for robust feature extraction in speech emotion recognition. In *Proceedings of INTERSPEECH*, pages 152–156, Hyderabad, India, 2018.

[5] Zexu Pan, Zhaojie Luo, Jichen Yang, and Haizhou Li. Multi-modal attention for speech emotion recognition, 2020.

[6] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. Emotion recognition from speech using wav2vec 2.0 embeddings. *arXiv preprint arXiv:2104.03502*, 2021.

[7] Sri Harsha Dumpala, Sheri Rempel, Katerina Dikaios, Mehri Sajjadian, Rudolf Uher, and Sageev Oore. Estimating severity of depression from acoustic features and embeddings of natural speech. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7278–7282. IEEE, 2021.

[8] Mireia Farrús and Joan Codina-Filbà. Combining prosodic, voice quality and lexical features to automatically detect alzheimer's disease. *arXiv preprint arXiv:2011.09272*, 2020.

[9] Abhijit Mohanta, Prerana Mukherjee, and Vinay Kumar Mirtal. Acoustic features characterization of autism speech for automated detection and classification. In *2020 National Conference on Communications (NCC)*, pages 1–6. IEEE, 2020.

[10] J. Kathryn Bock and Joanne R. Mazzella. Intonational marking of given and new information: Some consequences for comprehension. *Memory & Cognition*, 11(1):64–76, 1983.

[11] Lieke van Maastricht, Tim Zee, Emiel Krahmer, and Marc Swerts. L1 perceptions of L2 prosody: The interplay between intonation, rhythm, and speech rate and their contribution to accentedness and comprehensibility. In *Proceedings of INTERSPEECH*, pages 364–368, Stockholm, Sweden, 2017.

[12] John M. Levis and Alif O. Silpachai. Prominence and information structure in pronunciation teaching materials. In *Proceedings of the Pronunciation in Second Language Learning and Teaching conference*, pages 216–229, Ames, IA, USA, 2017.

[13] Mara Breen, Evelina Fedorenko, Michael Wagner, and Edward Gibson. Acoustic correlates of information structure. *Language and Cognitive Processes*, 25(7/8/9):1044–1098, 2010.

[14] Kamini Sabu, Prakhar Swarup, Hitesh Tulsiani, and Preeti Rao. Automatic assessment of children's l2 reading for accuracy and fluency. In *SLaTE*, pages 121–126, 2017.

[15] Heini Kallio, Antti Suni, and Juraj Šimko. Fluency-related temporal features and syllable prominence as prosodic proficiency predictors for learners of english with different language backgrounds. *Language and Speech*, 0(0):00238309211040175, 0. PMID: 34479458.

[16] Antti Suni, Sofoklis Kakouros, Martti Vainio, and Juraj Šimko. Prosodic prominence and boundaries in sequence-to-sequence speech synthesis. *arXiv preprint arXiv:2006.15967*, 2020.

[17] Andrew Rosenberg. *Automatic Detection and Classification of Prosodic Events*. PhD thesis, Columbia University, 2009.

[18] Taniya Mishra, Vivek Rangarajan Sridhar, and Alistair Conkie. Word prominence detection using robust yet simple prosodic features. In *Proceedings of INTERSPEECH*, pages 1864–1867, Portland, OR, USA, 2012.

[19] George Christodoulides and Mathieu Avanzi. An evaluation of machine learning methods for prominence detection in French. In *Proceedings of INTERSPEECH*, pages 116–119, Singapore, 2014.

[20] Matthew P. Black, Daniel Bone, Zisis Iason Skordilis, Rahul Gupta, Wei Xia, Pavlos Papadopoulos, Sandeep Nallan Chakravarthula, Bo Xiao, Maarten Van Segbroeck, Jangwon Kim, Panayiotis G. Georgiou, and Shrikanth S. Narayanan. Automated evaluation of non-native english pronunciation quality: Combining knowledge- and data-driven features at multiple time scales. In *Proceedings of INTERSPEECH*, pages 493–497, Dresden, Germany, 2015.

[21] Rose Sloan, Syed Sarfaraz Akhtar, Bryan Li, Ritvik Shrivastava, Agustin Gravano, and Julia Hirschberg. Prosody prediction from syntactic, lexical, and word embedding features. In *10th ISCA Speech Synthesis Workshop*, 2019.

[22] Andrew Rosenberg, Raul Fernandez, and Bhuvana Ramabhadran. Modeling phrasing and prominence using deep recurrent learning. In *Proceedings of INTERSPEECH*, pages 3066–3070, Dresden, Germany, 2015.

[23] Andrew Rosenberg. AuToBI - A tool for automatic ToBI annotation. In *Proceedings of INTERSPEECH*, pages 146–149, Makuhari, Japan, 2010.

[24] Yizhi Wu, Hongyan Li, and Sha Li. Automatic pitch accent detection using long short-term memory neural networks. In *Proceedings of International Symposium on Signal Processing Systems*, Beijing, China, 2019.

[25] Binghuai Lin, Liyuan Wang, Xiaoli Feng, and Jinsong Zhang. Joint detection of sentence stress and phrase boundary for prosody. In *Proceedings of INTERSPEECH*, pages 4392–4396, Shanghai, China, 2020.

[26] Elizabeth Nielsen, Mark Steedman, and Sharon Goldwater. The role of context in neural pitch accent detection in English. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2020.

[27] Sabrina Stehwien and Ngoc Thang Vu. Prosodic event recognition using convolutional neural networks with context information. In *Proceedings of INTERSPEECH*, pages 2326–2330, Stockholm, Sweden, 2017.

[28] Sabrina Stehwien, Antje Schweitzer, and Ngoc Thang Vu. Acoustic and temporal representations in convolutional neural network models of prosodic events. *Speech Communication*, 125:128–141, 2020.

[29] Long Zhang, Fanbo Meng Jia Jia and, Suping Zhou, Wei Chen, Cunjun Zhang, and Runnan Li. Emphasis detection for voice dialogue applications using multi-channel convolutional bidirectional long short-term memory network. In *Proceedings of INTERSPEECH*, Hyderabad, India, 2018.

[30] Yibin Zheng, Jianhua Tao, Zhengqi Wen, and Ya Li. Blstm-crf based end-to-end prosodic boundary prediction with context sensitive embeddings in a text-to-speech front-end. In *Interspeech*, pages 47–51, 2018.

[31] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with SincNet. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018.

[32] Dan Oneață, Lucian Georgescu, Horia Cucu, Dragoş Burileanu, and Corneliu Burileanu. Revisiting SincNet: An evaluation of feature and network hyperparameters for speaker recognition. In *Proceedings of European Signal Processing Conference*, pages 1–5, 2021.

[33] Joseph Roy, Jennifer Cole, and Timothy Mahrt. Individual differences and patterns of convergence in prosody perception. *Laboratory Phonology*, 8(1):22, 2017.

[34] Jason Bishop, GraceKuo, and Boram Kim. Phonology, phonetics, and signal-extrinsic factors in the perception of prosodic prominence: Evidence from rapid prosody transcription. *Laboratory Phonology*, 82:22, 2020.

[35] Antti Suni, Sofoklis Kakouros, Martti Vainio, and Juraj Šimko. Prosodic prominence and boundaries in sequence-to-sequence speech synthesis. In *Proceedings of Speech Prosody*, pages 940–944, Tokyo, Japan, 2020.

[36] Aarne Talman, Antti Suni, Hande Celikkanat, Sofoklis Kakouros, Jörg Tiedemann, and Martti Vainio. Predicting prosodic prominence from text with pre-trained contextualized word representations. In *Proceedings of Nordic Conference on Computational Linguistics*, pages 281–290, Turku, Finland, 2019.

[37] Sabrina Stehwien, Ngoc Thang Vu, and Antje Schweitzer. Effects of word embeddings on neural network-based pitch accent detection. In *Proceedings of Speech Prosody*, pages 314–318, Poznan, Poland, 2018.

[38] Jennifer Cole, Timothy Mahrt, and Joseph Roy. Crowd-sourcing prosodic annotation. *Computer Speech and Language*, 45:300–325, 2017.

[39] Joseph Roy, Jennifer Cole, and Timothy Mahrt. Individual differences and patterns of convergence in prosody perception. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 8(1), 2017.

[40] Stefan Baumann and Bodo Winter. What makes a word prominent? Predicting untrained German listeners' perceptual judgments. *Journal of Phonetics*, 70:20–38, 2018.

[41] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[42] Vikas Yadav and Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*, 2019.

[43] Ashima Yadav and Dinesh Kumar Vishwakarma. Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*, 53(6):4335–4385, 2020.

[44] Shengli Song, Haitao Huang, and Tongxiao Ruan. Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, 78(1):857–875, 2019.

[45] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[46] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota, 2019.

[47] Trang Tran, Shubham Toshniwal, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Mari Ostendorf. Parsing speech: A neural approach to integrating lexical and acoustic-prosodic information. In *Proceedings of NAACL-HLT*, New Orleans, Louisiana, 2018.

[48] Dimitri Palaz, Ronan Collobert, and Mathew Magimai-Doss. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 04 2013.

[49] Tara Sainath, Ron J Weiss, Kevin Wilson, Andrew W Senior, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. 2015.

[50] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5200–5204. IEEE, 2016.

[51] Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, et al. The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism. In *Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France*, 2013.

[52] Florian Eyben, Klaus R Scherer, Björn W Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y Devillers, Julien Epps, Petri Laukka, Shrikanth S Narayanan, et al. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE transactions on affective computing*, 7(2):190–202, 2015.

[53] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[54] Jee-Weon Jung, Hee-Soo Heo, Il-Ho Yang, Hye-Jin Shim, and Ha-Jin Yu. A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5349–5353. IEEE, 2018.

[55] Paul-Gauthier Noé, Titouan Parcollet, and Mohamed Morchid. CGCNN: COMPLEX GABOR CONVOLUTIONAL NEURAL NETWORK ON RAW SPEECH. In *ICASSP 2020*, Barcelona, Spain, May 2020.

[56] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi. LEAF: A learnable frontend for audio classification. In *International Conference on Learning Representations*, 2021.

[57] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[58] Sofoklis Kakouros and Okko Räsänen. 3PRO - An unsupervised method for the automatic detection of sentence prominence in speech. *Speech Communication*, 82:67–84, 2016.

[59] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[60] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[61] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

[62] Rainer Kelz, Sebastian Böck, and Cierhard Widnaer. Multitask learning for polyphonic piano transcription, a case study. In *Proceedings of International Workshop on Multilayer Music Representation and Processing*, pages 85–91, 2019.

[63] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, San Diego, CA, 2015.

[64] Hannah Muckenhirn, Mathew Magimai Doss, and Sébastien Marcell. Towards directly modeling raw speech signal for speaker verification using cnns. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4884–4888. IEEE, 2018.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[66] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[67] Samuel Kim, Fabio Valente, Maurizio Filippone, and Alessandro Vinciarelli. Predicting continuous conflict perceptionwith bayesian gaussian processes. *IEEE Transactions on Affective Computing*, 5(2):187–200, 2014.

[68] Prashanth Gurunath Shivakumar and Panayiotis Georgiou. Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations. *Computer speech & language*, 63:101077, 2020.

[69] Taejun Kim, Jongpil Lee, and Juhan Nam. Comparison and analysis of samplecnn architectures for audio classification. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):285–297, 2019.

[70] Jongpil Lee, Taejun Kim, Jiyoung Park, and Juhan Nam. Raw waveform-based audio classification using sample-level cnn architectures. *arXiv preprint arXiv:1712.00866*, 2017.

[71] Manraj Singh Grover, Yaman Kumar, Sumit Sarin, Payman Vafaee, Mika Hama, and Rajiv Ratn Shah. Multi-modal automated speech scoring using attention fusion, 2020.

[72] Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR, 2020.

[73] Jack Weston, Raphael Lenain, Udeepa Meepegama, and Emil Fristed. Learning de-identified representations of prosody from raw audio. In *International Conference on Machine Learning*, pages 11134–11145. PMLR, 2021.