ICON 2021

The 18th International Conference on Natural Language Processing

Proceedings of the Workshop on Speech and Music Processing 2021 (SMP2021)

December 16, 2021

Preface

Speech and music are two prominent research areas in the domain of audio signal processing. With recent advancements in speech and music technology, the area has grown manifolds, bringing together the interdisciplinary researchers of computer science, musicology and speech analysis. The languages we speak propagate as sound wave through various media and allow communication or entertainment for us, humans. The music we hear or create can be perceived in different aspects as rhythm, melody, harmony, timbre, or mood. The multifaceted nature of speech or music information requires algorithms, systems using sophisticated signal processing, and machine learning techniques to better extract useful information. This workshop will provide both profound technological knowledge and a comprehensive treatment of essential topics in speech and music processing.

Recent computational advancement has opened up several avenues to explore further the domain of speech and music. A deep understanding of both speech and music in terms of perception, emotion, mood, gesture and cognition is in the forefront, and many researchers are working in these domains. In this digital age, overwhelming data have been generated across the world that requires efficient processing for better maintenance, retrieval, indexing and querying. Machine learning and artificial intelligence are most suited for these computational tasks.

The SMP-2021 workshop was organized with the following objects: (i) to bring researchers and developers together who work on speech and music domain. (ii) to provide a platform for researchers to discuss speech prosody, Indian as well as western music and (iii) to encourage researchers to collaborate and create more annotated resources.

Technical Session:

The SMP-2021 workshop received nine submissions by authors from India, China, Canada, and Ireland. Each paper was reviewed by 2-3 experts. Based on reviewers' comments, six papers were accepted for presentation at the workshop. However, five papers were presented during the workshop session.

The accepted and presented papers include a variety of topics from both speech and music processing domains. Two papers covered speech emotion recognition in multimodal context and speech prosody in Hindi language. While three papers covered music, out of which two papers presented language, artist and melody identification from Indian classical music and one paper discussed about Dorabella Cipher as western music context.

Acknowledgement:

Organizers would like to thank everyone who supported us to organize the Workshop on Speech and Music Processing 2021 (SMP2021). Specifically, the ICON-2021organizers and the technical program committee members need for the workshop facilitation and support in reviewing papers, respectively.

SMP-2021 Organizers:

- Dr. Anupam Biswas, Department of CSE, National Institute of Technology Silchar, India
- Dr. Rabul H. Laskar, Department of ECE, National Institute of Technology Silchar, India
- Dr. Pinki Roy, Department of CSE, National Institute of Technology Silchar, India

Biography of Organizers:

Anupam Biswas received his Ph.D. degree in computer science and engineering from Indian Institute of Technology (BHU), Varanasi, India in 2017. He has received his M. Tech. and B. E. Degree in computer science and engineering from Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India in 2013 and Jorhat Engineering College, Jorhat, Assam in 2011 respectively. He is currently working as an Assistant Professor in the Department of Computer Science & Engineering, National Institute of Technology Silchar, Assam. He has published several research papers in reputed international journals, conference and book chapters. His research interests include Machine learning, Social Networks, Computational music, Information retrieval, and Evolutionary computation. He is Principal Investigator of two on-going DST-SERB sponsored research projects in the domain of machine learning and evolutionary computation. He has served as Program Chair of International Conference on Big Data, Machine Learning and Applications (BigDML 2019) and currently serving as Publicity Chair of BigDML 2021. He has served as General Chair of 25th International Symposium Frontiers of Research in Speech and Music (FRSM 2020) and co-edited the proceedings of FRSM 2020 published as book volume in Springer AISC Series. He edited two books titled "Health Informatics: A Computational Perspective in Healthcare" and "Principles of Social Networking: The New Horizon and Emerging Challenges" published by Springer. Also edited a book "Principles of Big Graph: In-depth Insight" with Advances in Computers book Series of Elsevier, currently in press.

Rabul Hussain Laskar is an Associate Professor in the Department of Electronics and Communication Engineering at National Institute of Technology Silchar. He completed his Bachelor of Engineering (B.E.) in Electronics and Communication Engineering in 1998, Masters in Signal processing (M.Tech) in 2009, and Doctorate in speech signal processing in 2013. He has been involved in teaching and research for the past 21 years. He is leading the signal processing research group in ECE department. His area of specialization includes digital signal processing, machine learning, computer vision, pattern recognition, analog and digital communication. He has been teaching subjects like signals and systems, linear algebra and random process, communication system engineering, digital speech processing, digital image processing, etc., among the student and research community. He has published 80+ research articles in various referred journals, 70+ papers in international/national conferences, 10+ chapters in different book series. He has been an active reviewer in various international journals and conferences. He has supervised 12 Ph.D. scholars, 10 post-graduate, 40 undergraduates in different areas of speech, image, video and biomedical signal processing. There are 08 Ph.D. students working under his supervision to complete their thesis. He has been involved in various R&D projects sponsored by DIT, SERB, MCIT, DST, BARC- BRNS as principal investigator. He is the senior member of IEEE, IETE fellow, IEI fellow.

Pinki Roy is working in NIT-Silchar for past 17 years. Her research interests include machine intelligence, language identification, Speech Processing, Health informatics and Cloud Computing. She has published more than 24 SCI/SCIE/Scopus Journals with highest impact factor of more than 7. She has more than 30 publications in various IEEE conferences/other reputed conferences and in Book Chapters. She had travelled to international destinations like Singapore, Sydney (Australia) and London to attend world top level conferences where her research work was highly appreciated. Already 3 PhD scholars have attained PhD degrees under her guidance in the area of cloud computing and speech processing. At present 4 scholars are working under her guidance. She was the recipient of Distinguished Lady Alumnus Award",

from Dr. Babasaheb Ambedkar technological university, Lonere, Maharashtra, 2014,"Young Scientist Award", Venus International foundation, Chennai 2015. Awarded for major contribution in research during PhD, "Rastriya Gaurav Award", India International Friendship Society, New Delhi 2015. She was also recipient of "Bharat Excellence Award", "Best Golden personalities Award", "Global Award for Education" by the Friendship Forum, New Delhi in the year 2016.She has organized workshops on Machine learning where people from the Academics and Industry were invited to bridge the gap between both the organizations. She has chaired several sessions in many renowned International Conferences and delivered expert talks in various workshops from time to time. She has served as TPC member and had reviewed many papers in International conferences and Journals.

Program Committee:

- Dr. Alicja Wieczorkowska, Polish-Japanese Academy of Information Technology, Poland
- Dr. Archi Banerjee, IIT Kharagpur, India
- Dr. Kaustav Kanti Ganguly, New York University, Abu Dhabi
- Prof. Debasish Samanta, IIT Kharagpur, India
- Prof. Kaushik Roy, West Bengal State University Barasat, India
- Dr. Suyel Namsudra, NIT Patna, India
- Dr. Vijay Bhaskar Semwal, MANIT Bhopal, India
- Dr. Mridula Verma, IDRBT Hyderabad, India
- Dr. Prabu Mohandas, NIT Trichy, India
- Dr. Durgesh Singh, IIITDM, Jabalpur, India
- Dr. Thoudam Doren Singh, NIT Silchar, India
- Dr. Rahul Chandra Kushwaha, RGU, Arunachal Pradesh, India
- Dr. Pravin Kumar, Allahabad University, Prayagraj, India

Table of Contents

Classifying Emotional Utterances by Employing Multi-modal Speech Emotion Recognition
Dipankar Das 1

Prosody Labelled Dataset for Hindi
Esha Banerjee, Atul Kr. Ojha and Girish Jha 14

Multitask Learning based Deep Learning Model for Music Artist and Language Recognition
Yeshwant Singh and Anupam Biswas 20

Comparative Analysis of Melodia and Time-Domain Adaptive Filtering based Model for Melody Extraction from Polyphonic Music
Ranjeet Kumar, Anupam Biswas, Pinki Roy and Yeshwant Singh 24

Dorabella Cipher as Musical Inspiration
Bradley Hauer, Colin Choi, Abram Hindle, Scott Smallwood and Grzegorz Kondrak 33

Conference Program

16 December 2021

Classifying Emotional Utterances by Employing Multi-modal Speech Emotion Recognition
Dipankar Das

Prosody Labelled Dataset for Hindi
Esha Banerjee, Atul Kr. Ojha and Girish Jha

Multitask Learning based Deep Learning Model for Music Artist and Language Recognition
Yeshwant Singh and Anupam Biswas

Comparative Analysis of Melodia and Time-Domain Adaptive Filtering based Model for Melody Extraction from Polyphonic Music
Ranjeet Kumar, Anupam Biswas, Pinki Roy and Yeshwant Singh

Dorabella Cipher as Musical Inspiration
Bradley Hauer, Colin Choi, Abram Hindle, Scott Smallwood and Grzegorz Kondrak

Classifying Emotional Utterances by Employing Multi-modal Speech Emotion Recognition

Dipankar Das

Computer Sc. & Engineering Department,
Jadavpur University, West Bengal, India

dipankar.dipnil2005@gmail.com

Abstract

Deep learning methods are being applied to several speech processing problems in recent years. In the present work, we have explored different deep learning models for speech emotion recognition. We have employed normal deep feed-forward neural network (FFNN) and convolutional neural network (CNN) to classify audio files according to their emotional content. Comparative study indicates that CNN model outperforms FFNN in case of emotions as well as gender classification. It was observed that the sole audio based models can capture the emotions up to a certain limit. Thus, we attempted a multi-modal framework by combining the benefits of the audio and text features and employed them into a recurrent encoder. Finally, the audio and text encoders are merged to provide the desired impact on various datasets. In addition, a database consists of emotional utterances of several words has also been developed as a part of this work. It contains same word in different emotional utterances. Though the size of the database is not that large but this database is ideally supposed to contain all the English words that exist in an English dictionary.

1 Introduction

Human Computer Interaction (HCI) researches the way we humans interact with a computer in order to improve the existing technologies. Thus, Automatic Speech Recognition (ASR) has been an active field of AI research aiming to generate machines that communicate with people via speech [1] [2]. In recent trends, simple text based chatbot systems are adding extra flavor of personalized experiences to their users through speech interactions. However, emotions always play the important roles in our interactions with people and computers. Fundamental publications of Rosalind Picard on affective computing increased the awareness in HCI community regarding important roles of emotion [3] [4] [5] [6]. Since then, researchers have also become increasingly aware of the importance of emotion in the design process [7].

Speech Emotion Recognition (SER) is mostly beneficial for commercial HCI applications, such as speech synthesis, customer service, education, forensics and medical analysis. Emotion recognition is used in call center for classifying calls according to emotions [8] and it serves as the performance parameter for conversational analysis [9], customer satisfaction and so on. SER is also used in automotive industry especially in car board system based on mental state of the driver to initiate his/her safety by preventing accidents to happen [10].

Affective computing and HCI research used to target in reducing user frustration, building tools to support development of socio-emotional skills [11]. Without information about emotions, it is difficult to achieve a harmonic and natural man-machine interface for applications such as patient care, geriatric nursing, call centers, psychological consultation, and human communication [12]. Therefore, health care industry is becoming prominent because it leverages emotion recognition techniques to solve complex patient related problems.

Speech is an information-rich signal that contains paralinguistic information as well as linguistic information. As a result of this, speech conveys more emotional information than text. This reality motivates many researchers to

Proceedings of the Workshop on Speech and Music Processing (SMP-2021), pages 1–13
co-located with ICON-2021, December 16-19, 2021, Silchar, India.

consider speech signal as a quick, effective and natural process to identify interaction mysteries between computer and human. Although, there is a significant improvement in speech recognition but still researchers are away from natural interplay between computer and human, since computer is not capable of understanding human emotional state. The recognition of emotional speech aims to recognize the emotional condition of individual utterances by applying his/her voice automatically. Recognizing of emotional conditions in speech signals are so challenging area for several reasons.

Majority of the speech emotional methods used to select the best features that are powerful enough to distinguish between different emotions.

The presence of various languages, accents, sentences, speaking styles, speakers also adds another difficulty because these characteristics directly change most of the extracted features including pitch and energy [13].

Furthermore, it is possible to have a more than one specific emotion at a time in the same speech signal and each emotion may correlate with a different part of speech signals. Therefore, defining the boundaries between parts of emotion is very challenging task.

In the present task with respect to speech emotion recognition, we have proposed two systems based on deep learning method to classify a speech signal according to its emotional content.

1. The first model is based on simple deep Feed-Forward Neural Network (FFNN). As it is a very basic model, it was unable to recognize enough important features from speech signal to classify it accurately. The overall accuracy that we achieved from this system is only 40%.

2. The second model is based on Convolutional Neural Network (CNN) model. Our main contribution lies in the way we applied the CNN model to our dataset. In several studies, it is observed that CNN have been used to classify speech emotion but the CNN model was applied on the spectrogram image which is a visual representation of the spectrum of frequencies of an audio signal. In contrast, we have applied our CNN model on the array of low-level MFCC features, extracted from the spectrogram image of an audio signal. Due to this fact, we used 1- Dimensional Convolutional layers in our CNN and not 2-Dimensional ones, which are generally used on image data. The overall accuracy we achieved from this model is 65%.

Now apart from these two systems, we have also developed an emotional lexicon that contains utterances of words along with their emotional class. Moreover, the lexicon also contains the utterances of a particular word when belongs to one or more emotion categories (same word can belong to multiple categories of emotion).

Finally, we introduce a deep recurrent encoder model that exploits text data and audio signals both simultaneously to obtain a better understanding of the emotional aspects in speech signals. In real world, a multi-modal dialogue system is composed of sound and spoken content. This actually motivated us to build a system which can encode the information from audio and text sequences and then can combine the information from these sources to predict the emotion class. Our system reported accuracies ranging from 62.7% to 70.8% when it was applied to the IEMOCAP dataset.

The rest of the paper is organized as follows. Section 2 describes the related attempts carried out under speech emotion recognition. The details on two types of emotional speech datasets along with two different models for speech emotion classification are discussed in Section 3. Section 4 describes a deep learning based multi-modal framework that takes into account the roles of speech and text in order to develop an improved system. Experiments and associated results with respect to all the models and framework are explained in Section 5. Finally, Section 6 briefs the process of developing speech emotion lexicon as an outcome whereas the concluding remarks are made in Section 7.

2 Related Work

If we observe a comprehensive review of speech emotion recognition systems targeting pattern recognition researchers who do not necessarily have a deep background in speech analysis, we notice three main aspects of this research field: (1) important design criteria of emotional speech corpora, (2) impact of speech features on the classification performance of SER and (3) classification systems employed in SER.

L. Chen et al. [15] used multi-level SVM classifier and ANN to reduce dimensionality by employing several parmeters (e.g., energy, ZCR, pitch, SC, spectrum cut-off frequency, correlation density (Cd), fractal dimension, MFF etc.) and obtained 86.5%, 68.5% and 50.2% recognition rates at different levels on Beihang University Database of Emotional Speech (BHUDES). Similarly, the authors in [16] used binary classifier and QDC with prosodic and contour features to obtain 75.8% rate of recognition on SEMAINE functional data. In recent trends, H. Cao et al. [14] used SVM with prosodic and spectral features and obtained 44.4% recognition rate on Berlin & LDC & FAU Aibo dataset.

In addition to the above mentioned works, in [17], a novel Modulation Spectral Features (MSFs) for the recognition of human emotions in speech is presented. An auditory-inspired ST representation is acquired by deploying an auditory filter bank as well as a modulation filter bank, to perform spectral decomposition in the conventional acoustic frequency domain and in the modulation frequency domain, respectively

This authors in [18] focused on the data pre-processing techniques which aim to extract the most effective acoustic features to improve the performance of the emotion recognition. The technique can be applied on a small sized data set with a high number of features. The presented algorithm integrates the advantages from a decision tree method and the random forest ensemble. Experiment results on a series of Chinese emotional speech data sets indicate that the presented algorithm can achieve improved results on emotional recognition, and outperform the commonly used Principle Component Analysis (PCA) / Multi-Dimensional Scaling (MDS) methods, and the more recently developed ISO-Map dimensionality reduction method.

In [19], a fusion-based approach to emotion recognition of affective speech using multiple classifiers with acoustic-prosodic information (AP) and semantic labels (SLs) is presented. The acoustic-prosodic information was adopted for emotion recognition using multiple classifiers and the MDT was used to select an appropriate classifier to output the recognition confidence.

It is observed that all the above mentioned approaches are either tried to deal with signals, acoustic features or to use machine learning classifiers and feature reduction techniques to improve the performance of SER. In contrast, our proposed method is based on deep learning and applied on three different datasets to show the effectiveness. In addition, the multi-modal framework deals with both the texts and speech together to capture the insights under the deep learning umbrella. The development of speech emotion lexicon directs us the utilization of the proposed models.

3 Speech Emotion Recognition

A single word can be associated with multiple emotions [20]. Based on this hypothesis, we have built our emotion classifier and chosen datasets carefully. Although there are several other modalities such as facial expression, body language, through which emotions can be expressed but we limited our present study to speech modality only. Speech emotion corpora that were prepared by actors have been used in the current study because the emotions expressed with exaggeration potentially compensate the lack of information provided by other modalities. This also allows us to explore the effectiveness of deep learning models with greater control compared with daily-life utterances. However, we limited our model to classify emotions for 'English' language only.

3.1 Speech Emotion Corpora

SAVEE: British English Database: The Surrey Audio-Visual Expressed Emotion (SAVEE) database was recorded from four native English male speakers (identified as DC, JE, JK and KL), postgraduate students and researchers at the University of Surrey aged from 27 to 31 years. Emotion has been described psychologically in discrete categories: anger, disgust, fear, happiness, sadness and surprise [21]. This is supported by the cross-cultural studies of Ekman [22] and studies of automatic emotion recognition tended to focus on recognizing these [23]. We added the class neutral to provide recordings of 7 emotion categories. The text material consisted of 15 sentences per emotion: 3 common among all emotions, 2 emotion-specific and 10 generic sentences that were different for each emotion and phonetically-balanced. The sampling rate of all recordings was 44.1 kHz. The 3 common and $2 \times 6 = 12$ emotion-specific sentences were recorded as neutral to give 30 neutral sentences. This

resulted in a total of 120 utterances per speaker, for example:

***Common**: She had your dark suit in greasy wash water all year.*

***Anger**: Who authorized the unlimited expense account?*

***Disgust**: Please take this dirty table cloth to the cleaners for me.*

***Fear**: Call an ambulance for medical assistance.*

***Happiness**: Those musicians harmonize marvelously.*

***Sadness**: The prospect of cutting back spending is an unpleasant one for any governor.*

***Surprise**: The carpet cleaners shampooed our oriental rug.*

***Neutral**: The best way to learn is to solve extra problems.*

RAVDESS: Emotional Speech and Song Database: The corpus, Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [24] contains 7356 files (total size: 24.8 GB). The database contains 24 professional actors (12 male, 12 female), vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise and disgust expressions whereas song contains calm, happy, sad, angry and fearful emotions. The statements are "Kids are talking by the door" and "Dogs are sitting by the door". Each expression is produced at two levels of emotional intensity (normal and strong) with an additional neutral expression. All conditions are available in three modality formats: Audio-only (16bit, 48kHz .wav), Audio-Video (720p H.264, AAC 48kHz, .mp4), and Video-only (no sound). We used only the audio modality as our focus was on the recognition of emotion from speech. Speech file (size 215 MB) contains 1440 files: 60 trials per actor x 24 actors = 1440.

3.2 Data Cleaning and Pre-processing

In order to have a consistent sampling rate across all databases, all utterances were resampled and filtered by an antialiasing FIR low pass filter to have frequency rate of 44.1 kHz prior to any processing. All audio utterances were then converted into spectrograms. A spectrogram is an image that displays the variation of energy at different frequencies across time. There are two general types of spectrograms: wide-band and narrow-band spectrograms. Wide-band spectrograms have higher time resolution than narrow-band spectrograms. This property enables the wide-band spectrograms to show individual glottal pulses. In contrast, narrow-band spectrograms have higher frequency resolution than wide-band spectrograms. This feature enables the narrow-band spectrograms to resolve individual harmonics. Considering the importance of vocal fold vibration, along with the fact that glottal pulse is associated with one period of vocal fold vibration, we decided to convert all utterances into wide-band spectrograms.

3.3 Model 1: Feed Forward Neural Network (FFNN)

Deep feed-forward neural network constitutes several layers of hidden neurons, where each neuron is connected to every neuron in its previous layer. The first layer is called input layer. For our study, the input layer consists of 216 MFCC features extracted from the audio data and the batch size has been set to 16. Thus, the dimension of our input data is (16 X 216). We employed three hidden layers in our architecture as depicted in Figure 1. The number of neurons in the first, second and third hidden layer are 256, 512 and 256, respectively. We have used the Rectified Linear Unit (ReLU) activation function in all of the three hidden layers to achieve non-linearity. Only in the output layer, softmax activation function is used as it gives the probability distribution across 10 output classes. As the problem is a classification problem, we have used the cross-entropy loss. Adam optimizer is also employed to minimize the loss function across the training data. We have also employed a dropout rate of 20% after every hidden layer. The dropout layers are employed to reduce the over-fitting problem.

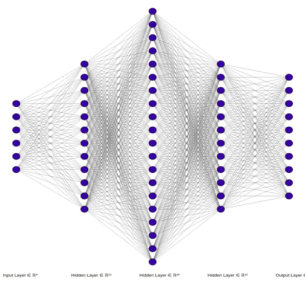

Figure 1: *Baseline architecture of FFNN*

3.4 Model 2: Convolutional Neural Network (CNN)

We have applied Convolutional Neural Network algorithm on audio data. As the data is of one-dimensional, we cannot use the conventional CNN architecture used for image data in general. As a result, we used 1-D convolutional layers instead of the most popular 2-D convolutional layers. All other layers like max-pooling and dense layers are used as it is. The convolutional neural network (CNN) architecture that has been implemented in the current study constitutes two convolutional layers and two fully connected layer, also known as dense layers. Among the two dense layers, the first one has 128 hidden neurons and the second one has 256 hidden neurons. For the current study, we tried to classify each audio file to a particular emotion class among 5 emotion classes and also to classify the gender of the voice. Thus, we have 10 (5 emotions X 2 genders) output classes. As a result we added 10 softmax units in our last (output) layer to estimate the probability distribution of the classes.

In our architecture, every convolutional layer is followed by a max-pooling layer. Each of the first and second convolutional layers is followed by a 1-D max-pooling layer with max-pooling window size of 7 and 4, respectively. The number of kernels (filters) is set to 64 and 128 for the first and second convolutional layers, respectively. The sizes of the kernels that have been applied to the first and second convolutional layers are 5 and 3, respectively. Batch size of 16 is applied throughout the training process. Rectified Linear Units (ReLU) were used in convolutional layers and fully connected layers, except in the last dense layer, as activation functions to introduce non-linearity to the model. Similar to Model 1, as the problem is a classification problem, we have used the cross-entropy loss and adam optimizer. The number of epochs is set to 100. The training procedure for this study was performed entirely on a CPU-based system, no GPU has been used for conducting any part of the training process. We have also used dropout and flatten function. Flatten function is used whenever we needed to reduce the dimension of the data which was output by a layer in the network whereas dropout layer is used to reduce the over-fitting issue during training process. Dropout layers reduce over-fitting by dropping out or ignoring some of the neurons. We have used two dropout layers in our network architecture with each of them residing right after each of the two dense layers. A dropout rate of 20% has been used in both of the two cases. Figure 2 gives a detailed overview of the network with the input and output dimensions of data in each of the layer in the network.

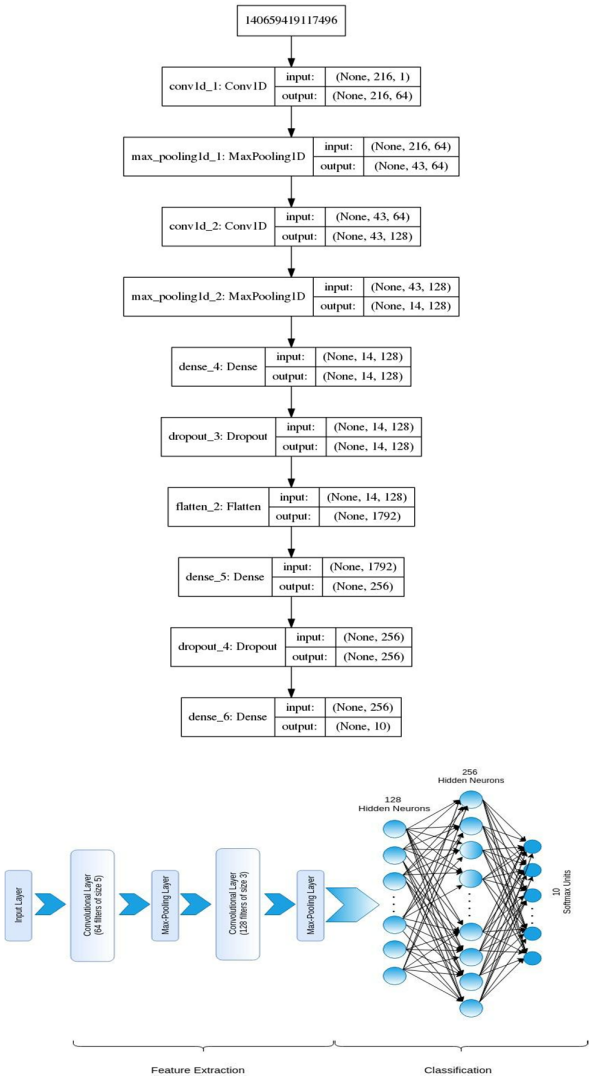

Figure 2: *Baseline architecture of CNN.*

4 Multi-Modal Analysis

Recently, we agree that the deep learning algorithms have successfully addressed problems in various fields, such as image classification, machine translation, speech recognition, text-to-speech generation and other machine learning related areas [30] [31] [32]. Similarly, substantial improvements in performance have been obtained when deep learning algorithms have been applied to statistical speech processing [28]. Even though

various types of deep learning methods have been applied, this problem is still considered to be challenging for several reasons; first, the scarcity of emotion tagged data for training deep neural models and second, the characteristics of emotions must be learned from low-level speech signals. However, feature-based models display limited skills when applied to this problem.

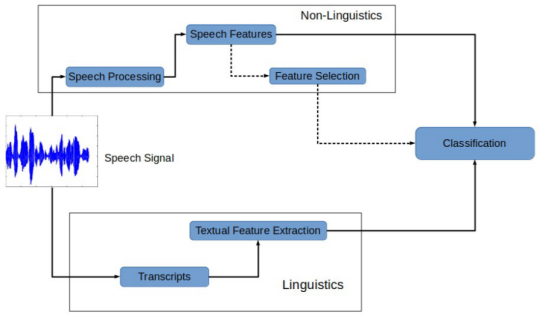

Figure 3: Multi-modal architecture of Audio (non-Linguistic) and Text (linguistic) models for speech emotion classification

In order to overcome these limitations, we have developed a model (as shown in Figure 3) that uses high-level text transcription, as well as low-level audio signals, to utilize the information contained within low-resource datasets to a greater degree. The emotional content of speech is clearly indicated by the emotion words contained in a sentence [29], such as "lovely" and "fantastic," which carry strong emotions compared to generic (non-emotion) words, such as "person" and "day." Thus, we hypothesize that the speech emotion recognition model will be benefit from the incorporation of high-level textual input with the low-level audio features. Moreover, this multimodal approach encodes both audio and textual information simultaneously via a dual recurrent encoder.

4.1 Audio-Only Encoder (AoE)

We have built an *Audio-only Encoder* (AoE) to predict the emotional class of a given audio signal based on only audio features. Once Mel-frequency cepstral coefficients (MFCCs) features have been extracted from an audio signal, a subset of the sequential features is fed into the recurrent neural networks (RNN), which is composed of gated recurrent units (GRUs), which in turn leads to the formation of the network's internal hidden state $\boldsymbol{h}_t$ to model the time series pattern. The updates of the hidden state is performed with the input data $\mathbf{x}_t$ and the hidden state output of the previous time step $\boldsymbol{h}_{t-1}$, which is basically the main working principle of a recurrent neural network. The present time hidden state $\boldsymbol{h}_t$ can be mathematically modeled as following:

$$\boldsymbol{h}_t = \boldsymbol{f}_w\,(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t), \qquad (1)$$

where $\boldsymbol{f}_w$ is a function which imitates the function of an RNN with weight parameter $\boldsymbol{w}$, $\boldsymbol{h}_t$ represents the hidden state at $\boldsymbol{t}^{th}$ time step, and $\mathbf{x}_t$ represents the $\boldsymbol{t}^{th}$ MFCC features in $\mathbf{x} = \{\mathbf{x}_{1:\,ta}\}$. After encoding the audio signal $\mathbf{x}$ with the RNN, the last hidden state of the RNN, $\boldsymbol{h}_{ta}$, is considered to be the representative vector that contains all of the sequential audio data. In this model, we have also incorporated the prosody features of an audio signal. The prosody of an audio signal is characterized by the inherent pattern of stress and intonation in a language. We have incorporated this characteristic in order to better classify the emotional content in an audio signal.

However, in order to implement, we have developed a prosodic feature vector, $\mathbf{p}$, which models the prosody of audio files. Then, we have concatenated the last hidden state vector, $\boldsymbol{h}_{ta}$, with the prosodic feature vector, $\mathbf{p}$, in order to generate more informative vector representation of the audio signal. We denote this more informative vector as $\mathbf{e}$, where $\mathbf{e} = concat\{\boldsymbol{h}_{ta},\ \mathbf{p}\}$. The MFCC and the prosodic features are extracted from the audio signal using the openSMILE toolkit [26] and $\mathbf{x}_t$ has 39 and $\mathbf{p}$ has 35 MFCC features.

Finally, the emotion class is predicted by applying the *softmax* function to the vector $\mathbf{e}$. For a given audio sample $\mathbf{i}$, we assume that $\boldsymbol{y}_i$ is the true label vector, which contains all zeros but contains a one at the correct class, and $\boldsymbol{y}'_i$ is the predicted probability distribution from the *softmax* layer. The training objective then takes the following form:

$$\boldsymbol{y}'_i = \boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{M} + \boldsymbol{b} \qquad (2)$$

$$\partial = -log \prod_{i=1}^{N} . \sum_{c=1}^{C} y_{i,c}\, log\ (\boldsymbol{y}'_{i,c})$$

where, **e** is the calculated representative vector of the audio signal with dimensionality $\mathbf{e} \in \mathbf{R}^d$. The $\mathbf{M} \in \mathbf{R}^{d \times C}$ and the bias **b** are learned model parameters, **C** is the total number of classes, and **N** is the total number of samples used in training.

4.2 Text-Only Encoder (ToE)

Apart from the audio, we tried to use the textual information as another modality in predicting the emotion class of a given signal. To use textual information, the speech transcripts are tokenized and indexed into a sequence of tokens using the Natural Language Toolkit (NLTK) [27]. Each token is then passed through a word embedding layer that converts a word index to a corresponding 300-dimensional vector that contains additional contextual meaning between words. The sequence of embedded tokens is fed into a *Text-only Encoder* (ToE) in such a way that the audio MFCC features are encoded using the AoE represented by equation 1. In this case, $\mathbf{x}_t$ is the t^{th} embedded token from the text input. Finally, the emotion class is predicted from the last hidden state of the text-RNN using the *softmax* function. We use here the same training objective as we adopted for the AoE model, and the predicted probability distribution for the target class is as follows:

$$y'_i = softmax\,(\boldsymbol{h_{last}}^{T}\boldsymbol{M} + \boldsymbol{b}) \qquad (3)$$

where $\boldsymbol{h}_{last}$ is the last hidden state of the text-RNN, $\boldsymbol{h}_{last} \in \mathbf{R}^d$, and $\mathbf{M} \in \mathbf{R}^{d \times C}$ and the bias **b** are learned model parameters. The lower part of Figure 3 and Figure 4 indicates the architecture of the ToE model.

4.3 Merged Recurrent Encoder (MRE)

In order to obtain the benefits from both the audio and text modes, we present an architecture called the merged recurrent encoder (MRE) to overcome the limitations of existing approaches. In this study, we consider multiple modalities, such as MFCC features, prosodic features and transcripts, which contain sequential audio information, statistical audio information and textual information, respectively. These types of data are the same as those used in the AoE and ToE cases.

However, the MRE model employs two RNNs to encode data from the audio signal and textual inputs, independently. The audio-RNN encodes MFCC features from the audio signal using equation 1. The last hidden state of the audio-RNN is concatenated with the prosodic features to form the final vector representation **e**, and this vector is then passed through a fully connected neural network layer to form the audio encoding vector **A**. On the other hand, the text-RNN encodes the word sequence of the transcript using equation 1. The final hidden states of the text-RNN are also passed through another fully connected neural network layer to form a textual encoding vector **T**. Finally, the emotion class is predicted by applying the *softmax* function to the concatenation of the vectors **A** and **T**. We use the same training objective as the AoE model, and the predicted probability distribution for the target class is as follows:

$$\mathbf{A} = \boldsymbol{g_o}(\mathbf{e})\,,\ \mathbf{T} = \boldsymbol{g'_o}(\mathbf{h}_{last})$$

$$y'_i = softmax\,(concat\,(\boldsymbol{A},\boldsymbol{T})^{T}\boldsymbol{M} + \boldsymbol{b}) \qquad (4)$$

where $\boldsymbol{g_o}$, $\boldsymbol{g'_o}$ is the feed-forward neural network with weight parameter **θ**, and **A**, **T** are final encoding vectors from the audio-RNN and text-RNN, respectively. $\mathbf{M} \in \mathbf{R}^{d \times C}$ and the bias **b** are learned model parameters.

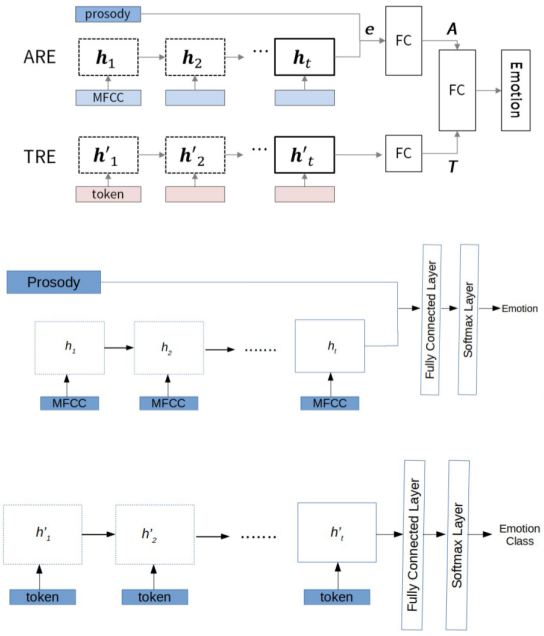

Figure 4: *Merged Recurrent Encoder. (The upper part shows AoE, which encodes audio signals and the lower part shows ToE, which encodes textual information).*

Dataset: We evaluated our multi-modal model on the Interactive Emotional Dyadic Motion Capture (IEMOCAP) [34] dataset. This dataset was collected theatrical theory in order to simulate natural dyadic interactions between actors. We use categorical evaluations with majority agreement. We use only four emotional categories viz. happy, sad, angry, and neutral to compare the performance of our model with other research using the same categories. The IEMOCAP dataset includes five sessions, and each session contains utterances from two speakers (one male and one female). This data collection process resulted in 10 unique speakers. For consistent comparison with previous work, we merge the excitement dataset with the happiness dataset. The final dataset contains a total of 5531 utterances (1636 happy, 1084 sad, 1103 angry and 1708 neutral).

Feature Extraction: In order to extract speech information from audio signals, we use MFCC values, which are widely used in analyzing audio signals. The MFCC feature set contains a total of 39 features, which include 12 MFCC parameters (1-12) from the 26 Melfrequency bands and log-energy parameters, 13 delta and 13 acceleration coefficients. The frame size is set to 25 ms at a rate of 10 ms with the Hamming function. According to the length of each wave file, the sequential step of the MFCC features is varied. To extract additional information from the data, we also use prosodic features, which show effectiveness in affective computing. The prosodic features are composed of 35 features, which include the F0 frequency, the voicing probability, and the loudness contours. All of these MFCC and prosodic features are extracted from the data using the OpenSMILE toolkit [26].

Setup Details: Among the variants of the RNN function, we use GRUs as they yield comparable performance to that of the LSTM and include a smaller number of weight parameters [28]. We use a max encoder step of 750 for the audio input, based on the implementation choices presented in [33] and 128 for the text input because it covers the maximum length of the transcripts. The vocabulary size of the dataset is 3,747, including the "_UNK_" token, which represents unknown words, and the "_PAD_" token, which is used to indicate padding information added while preparing mini-batch data. The number of hidden units and the number of layers in the RNN for each model (AoE, ToE, MRE) are selected based on extensive hyper-parameter tuning.

5 Experiments & Results

This section discusses the experiments performed in this study to classify emotion class and gender of the input audio data using the FFNN and CNN based deep learning models as described in the above sections. In order to have a comparative discussion, we restricted ourselves to 100 epochs for both the models.

The datasets used to train both of these networks already have been discussed in the Section 3.1. We encourage the readers to consult that section to have a detailed idea about the datasets. We have merged the audio files from the two datasets, SAVEE and RAVDESS, to produce the raw data. There are approximately 1900 audio files after merging. However, we were not able to use all the audio files to train our networks as the emotion classes of the two datasets were not identical. The emotion classes reported in SAVEE database are *anger, disgust, fear, happiness, sadness, surprise* and *neutral* whereas the emotion classes in RAVDESS database are *neutral, calm, happy, sad, angry, fearful, disgust* and *surprised*.

As we know *neutral* emotion does not specifically portray any emotion specific feature, we discarded all the sentences belong to *neutral* class from our raw dataset. Furthermore, we considered only 5 main classes of emotions namely, *calm, happiness, sadness, fear* and *anger*. As a result, we have approximately 1200 sentences in our raw database. In case of training and testing our models, we need to split our raw dataset, which is described in the previous section, to form the training and test data. We have taken approximately 80% of the raw dataset as training and the remaining 20% as test data to evaluate our models. The performances of the *neural* network models on this training and test set have been demonstrated in the following sections.

5.1 Results of FNNN Model

The performance of the deep Feed Forward Neural Network is measured in terms of training vs. test accuracy graph, training vs. test loss graph and two confusion matrices, one for emotion classification and another for gender classification.

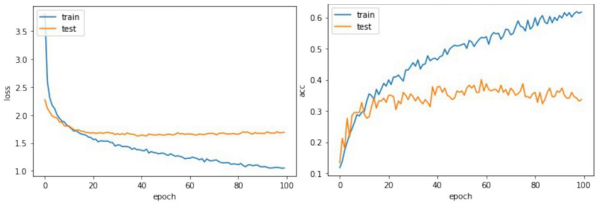

Figure 5: *Training vs. Test Loss and Training vs. Test Accuracy graph for FFNN model.*

It is very much clear from the above graphs in Figure 5 that the model did not perform very well; in fact it is very clear that over-fitting happened in this case. In order to investigate the reasons, we reported the confusion matrices. Figure 6 represents the overall confusion matrix for FFNN model.

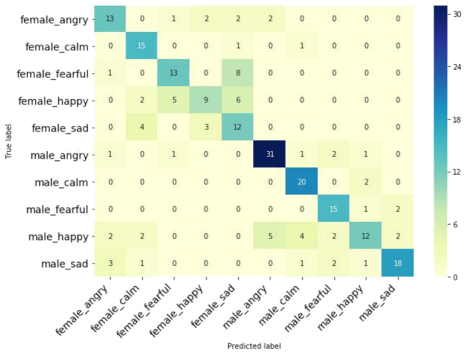

Figure 6: *Confusion matrix for Emotion and Gender classification using FFNN*

If we analyze the confusion matrix, we can conclude that the overall performance of the FFNN model is not good. We can see in the confusion matrix that *male_fearful, male_happy, male_sad* have been misclassified as *male_angry*. In addition, a considerable amount of *female_sad* and *male_sad* labels have been misclassified as *female_happy* and *male_fearful*, respectively. Overall, the accuracy achieved by this model is 40.82%. Figure 7 represents the confusion matrix only for gender labels that is *male* and *female*. The classification performed by this model for gender labels is far better than overall classification.

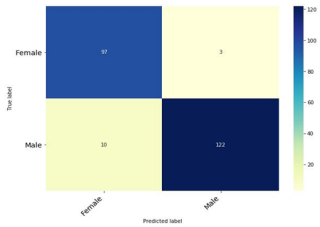

Figure 7: *Confusion matrix for Gender classification (Only) using FFNN model.*

5.2 Results of CNN Model

Unlike the FFNN model, CNN model did not suffer from over-fitting as can be seen in Figure 8. On the other hand, Figure 9 and Figure 10 represent the confusion matrices for overall classification and gender classification, respectively. If we analyze the confusion matrix for the overall classification, it surely outperforms our FFNN model as the misclassification rate is much lower in the case of CNN. Misclassification of *female_fearful* as *female_sad* is the only noticeable misclassification that happened in the whole confusion matrix. The accuracy for the overall classification is approximately 68.38%. This accuracy was achieved by running the training algorithm for 2000 epochs

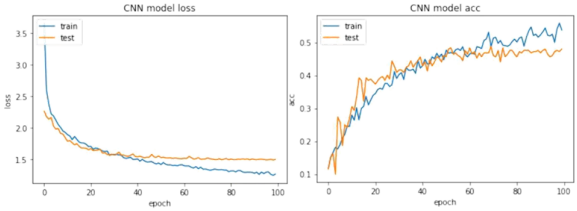

Figure 8: *Training vs. Test Loss and Training vs. Test Accuracy graph for CNN model*

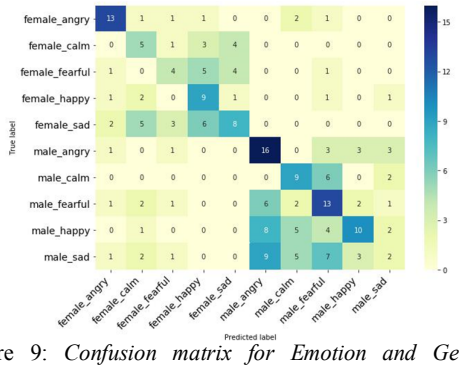

Figure 9: *Confusion matrix for Emotion and Gender classification using CNN.*

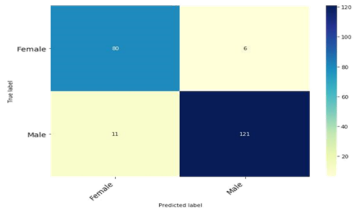

Figure 10: *Confusion matrix for Gender classification using CNN model.*

5.3 Results of Multi-Modal Model

The MRE model combines the benefits of AoE and ToE models and it receives approval if we see the curves of loss as well as accuracies over training and validation set, respectively. Moreover, the performances of individual models justify the multi-modal effects when used in combination. Table 1 shows the accuracies of various models.

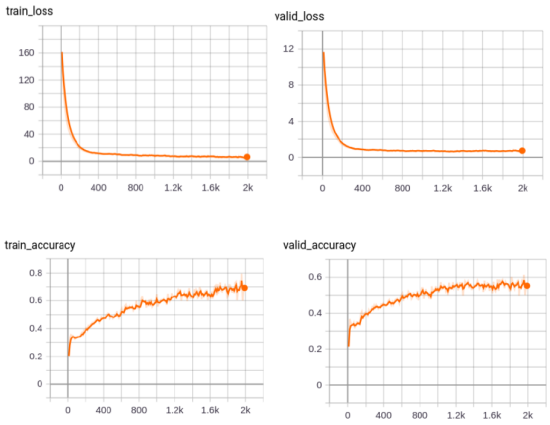

Figure 11: *Loss and accuracy curves for AoE model with (8:0.5:1.5) splitting into training, development and test data*

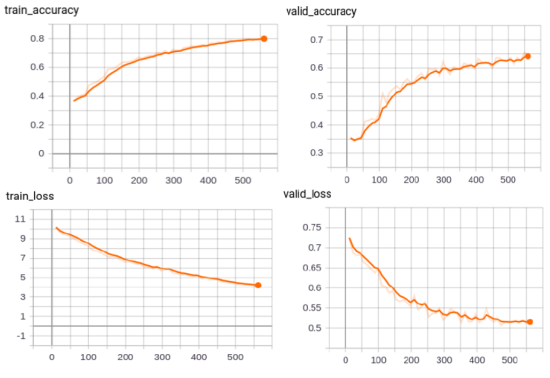

Figure 12: *Loss and accuracy curves for ToE model with (8:0.5:1.5) splitting into training, development and test data*

Model	Accuracy on Validation Set	Accuracy on Test Set
Model 1 (FFNN)	43.02%	40.82%
Model 2 (CNN)	64%-68.38%	52%-54.32%
Model 3.1 (AoE)	59.42%	59.5%
Model 3.2 (ToE)	63.58%	67.27%
Model 3.3 (MRE)	74.12%	74.64%

Table 1: Comparative analysis of accuracies of the various models on validation and test sets of IEMOCAP data.

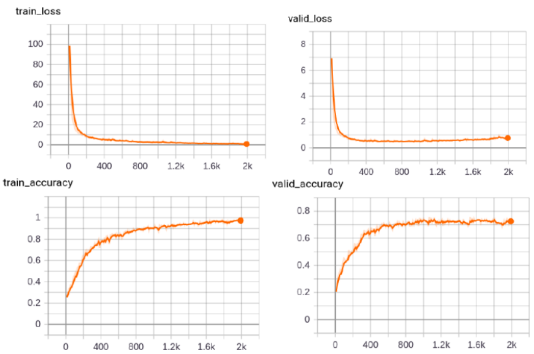

Figure 13: *Loss and accuracy curves for MRE model with (8:0.5:1.5) splitting into training, development and test data*

5.4 Error Analysis

We analyze the predictions of the AoE, ToE, and MRE models. Figure 14 shows the confusion matrix of each model. The ARE model (as shown in Figure 14(a)) incorrectly classifies most instances of *happy* as *neutral* (43.51%); thus, it shows reduced accuracy (35.15%) in predicting the *happy* class. Overall, most of the emotion classes are frequently confused with the *neutral* class. This observation is in line with the findings of [25], who noted that the *neutral* class is located in the center of the activation-valence space, complicating its discrimination from the other classes.

Interestingly, the ToE model (as shown in Figure 14(b)) shows gains in predicting the *happy* class when compared to the AoE model (35.15% to 75.73%). This result seems plausible because the model can benefit from the differences among the distributions of words in *happy* and *neutral* expressions, which gives more emotional information to the model than that of the audio signal data. On the other hand, it is unexpected that the ToE model incorrectly predicts instances of the *sad* class as the *happy* class 16.20% of the time, even though these emotional states are being present at oppose to one another.

The MRE model (as shown in Figure 14(c)) compensates for the weaknesses of the previous two models (AoE and ToE) and benefits from their strengths to a surprising degree. The values

arranged along the diagonal axis show that all of the accuracies of the correctly predicted class have increased. Furthermore, the occurrence of the incorrect "*sad-to-happy*" cases in the ToE model is reduced from 16.20% to 9.15%.

(a) AoE (b) ToE (c) MRE

Figure 14: *Confusion matrices of AoE, ToE and MRE models*

6 Speech Emotion Lexicon

Not only the classification of speeches into different emotion categories, but one of our partial objectives was also to explore the possibility of generating a speech emotional database consisting of emotional utterances also. Therefore, we have developed a speech emotion lexicon containing utterances of different emotional categories. However, it is difficult to simultaneously generate both *male* and *female* voices in a text-to-speech system and thus, we limited ourselves to synthesize only '*male*' voice in this present attempt. For this very reason, we selected our dataset with only *male* speakers

6.1 Pre-Processing

All audio i.e. our WAV files were resampled and filtered by an antialiasing FIR low pass filter to have frequency rate of 44.1 kHz prior to any processing. Silences and non-voiced parts at the start and the end have been removed from the files. The next step of developing the speech lexicon is to classify the emotion of each of the WAV files employing the best classifier.

6.2 Transcript Generation

To make our model robust, we made sure that we can build the lexicon from the WAV files which do not have any transcript associated with it. We made use of IBM Speech to Text API [1] service to obtain the transcript of a given WAV file.

6.3 POS Tagging and Word Segmentation

At this stage, we tag the words based on their part of speech and also segment the words of the audio using the transcript generated by the Text-to-Speech service. At first, Parts of Speech (POS) tagging of all the segmented words has been performed and we discarded the *proper nouns* (names, places etc.) as it conveys very little emotional features than *adjectives* or *adverbs* etc.

After this, we segment the words based on their start and end time in the audio files. We get the start and times of all the words from the results obtained from Text-to-Speech service described in the previous sections. Finally, we use this information to extract the words using Pydub[2], a Python library for audio processing.

6.4 Emotion Word Lexicon

The first column of the lexicon represents the words that have been spoken and second column represents the gender and third column represents the emotion in which the corresponding word has been spoken. The last column represents the location of the WAV file containing the utterance of the corresponding word in the specified emotion. We have also grouped same words spoken in different emotions. Presently, the lexicon contains only 1K words in 5 different emotion categories. Three native speakers have evaluated the emotional utterances and an agreement score of pair wise kappa k=0.92 was found. The minute disagreement was happened due to the segmentation and such words have been discarded from the lexicon.

7 Conclusions

In the present task, two classification models based on deep neural network, one using normal Feed-forward Neural Network (FFNN) and another using Convolutional Neural Network (CNN) architecture has been implemented and also a comparative study between these two models has been reported. Among the two models it has been shown that CNN model outperformed the FFNN model. The models have been developed from a training set which

[1] https://cloud.ibm.com/apidocs/speech-to-text

[2] https://pydub.com/

consists of only English language. It will be an interesting study to apply other languages to train the model and compare the performances for the same.

In addition to that, we have implemented a multi-modal version of the deep neural model using Recurrent Neural Network, in order to improve the classifier system. We have used features of both audio and text and merged them into a single framework to investigate the effectiveness of the system. It is observed that the performance of the multi-modal system is far better that the FFNN or CNN based system in classifying emotions.

As we have mentioned earlier that we have developed a database which contains emotional utterances. However, the size of the database is not very large, we had a limited number of test samples and hence it affected the development of the proposed database.

Acknowledgement

The work is supported by the SERB sponsored IMPRINT-II Project, DST, Government of India.

References

[1] Dong Yu and Li Deng., Automatic Speech Recognition. Springer,2016.

[2] Lawrence R Rabiner and Biing-Hwang Juang. Fundamentals of speech recognition, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.

[3] Picard, R.W. Affective Computing. M.I.T. Press, Cambridge, MA. 1997.

[4] Picard R. W., Healey J., Affective Wearables, Personal Technologies Vol 1, No. 4, pages 231-240. 1997.

[5] Picard R.W., Affective Computing for HCI. In Proc. of the 8th International Conference on Human- Computer Interaction: Ergonomics and User Interfaces-Volume I. Lawrence Erlbaum Associates, Inc. 1999.

[6] Picard R.W., Vyzas E., Healey J. Toward Machine Emotional Intelligence -Analysis of Affective Physiological State. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 23 No. 10, 2001.

[7] Norman, D.A. Emotional Design: Why we love (or hate) everyday things. Basic Books. 2003.

[8] F. Dipl and T. Vogt, "Real-time Automatic Emotion Recognition from Speech", 2010.

[9] S. Lugovic, I. Dunder, and M. Horvat, Techniques and applications of emotion recognition in speech, 2016 39th Int. Conv. Inf. Commun. Technol.797979 Electron. Microelectron. MIPRO 2016 - Proc., November 2017, pages 1278–1283, 2016.

[10] B. Schuller, G. Rigoll, and M. Lang, "Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine - belief network architecture," Acoust. Speech, Signal Process., vol. 1, pages 577–580, 2004.

[11] R. W. Picard, "Affective Computing for HCI," In HCI (1), pages 829– 833, 1999.

[12] F. Ren, "From cloud computing to language engineering, affective computing and advanced intelligence," International Journal of Advanced Intelligence, vol. 2(1), pages 1–14, 2010

[13] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik and Douglas D Edwards. Artificial intelligence: a modern approach, volume 2. Prentice hall Upper Saddle River, 2003.

[14] H. Cao, R. Verma, and A. Nenkova, "Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech," Comput. Speech Lang., vol. 28, no. 1, pages 186–202, Jan. 2015.

[15] L. Chen, X. Mao, Y. Xue, and L. L. Cheng, "Speech emotion recognition: Features and classification models", Digit. Signal Process., vol. 22, no. 6, pages 1154–1160, Dec. 2012.

[16] J. P. Arias, C. Busso, and N. B. Yoma, "Shape-based modeling of the fundamental frequency contour for emotion detection in speech," Comput. Speech Lang., vol. 28, no. 1, pages 278–294, Jan. 2014.

[17] S. Wu, T. H. Falk, and W.-Y. Chan, "Automatic speech emotion recognition using modulation spectral features," Speech Commun., vol. 53, no. 5, pp. 768–785, May 2011.

[18] J. Rong, G. Li, and Y.-P. P. Chen, "Acoustic feature selection for automatic emotion recognition from speech,"Inf. Process. Manag., vol. 45, no. 3, pp. 315–328, May 2009.

[19] C.-H. Wu and W.-B. Liang, "Emotion Recognition of Affective Speech Based on Multiple Classifiers Using Acoustic-Prosodic Information and Semantic Labels,"

IEEE Trans. Affect. Comput., vol. 2, no. 1, pp. 10–21,Jan. 2011.

[20] Changqin Quan, Fuji Ren, "An Exploration of Features for Recognizing Word Emotion", Proceedings of the *23rd International Conference on Computational Linguistics (Coling 2010)*, pages 922–930, Beijing, August 2010

[21] Sanaul Haq, Philip JB Jackson, and J Edge. Speaker-dependent audio-visual emotion recognition. In AVSP, pages 53–58, 2009.

[22] Ekman, P., "Universals and cultural differences in facial expressions of emotion", Nebraska Symposium on Motivation, pages 207-283, 1972.

[23] Zeng, Z., Pantic, M., Roisman, G.I. and Huang, T.S., "Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions", IEEE Trans. PAMI, 31(1), pages 39-58, 2009.

[24] Livingstone SR, Russo FA (2018) 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.

[25] Michael Neumann and Ngoc Thang Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," Proc. Interspeech 2017, pp. 1263– 1267, 2017.

[26] Florian Eyben, Felix Weninger, Florian Gross, and Bjorn Schuller, "Recent developments in opensmile, the ¨ munich open-source multimedia feature extractor," in Proceedings of the 21st ACM international conference on Multimedia. ACM, 2013, pp. 835–838.

[27] Steven Bird and Edward Loper, "Nltk: the natural language toolkit," in Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2004, p. 31.

[28] 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.

[29] Linhong Xu, Hongfei Lin, Yu Pan, Hui Ren, and Jianmei Chen, "Constructing the affective lexicon ontology," Journal of the China Society for Scientific and Technical Information, vol. 27, no. 2, pp. 180–185, 2008.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

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

[32] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in English and mandarin," in International Conference on Machine Learning, 2016, pp. 173–182.

[33] Michael Neumann and Ngoc Thang Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," Proc. Interspeech 2017, pp. 1263– 1267, 2017.

[34] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," Language resources and evaluation, vol. 42, no. 4, pp. 335, 2008.

Prosody Labelled Dataset for Hindi using Semi-Automated Approach

Esha Banerjee[1], Atul Kr. Ojha[2,3], Girish Nath Jha[1]
[1]Jawaharlal Nehru University, New Delhi, [2]DSI, National University of Ireland Galway, Ireland,
[3]Panlingua Language Processing LLP, New Delhi
(esha.jnu,shashwatup9k,girishjjha)@gmail.com

Abstract

This study aims to develop a semi-automatically labelled prosody database for Hindi, for enhancing the intonation component in ASR and TTS systems, which is also helpful for building Speech to Speech Machine Translation systems. Although no single standard for prosody labelling exists in Hindi, researchers in the past have employed perceptual and statistical methods in literature to draw inferences about the behaviour of prosody patterns in Hindi. Based on such existing research and largely agreed upon theories of intonation in Hindi, this study attempts to first develop a manually annotated prosodic corpus of Hindi speech data, which is then used for training prediction models for generating automatic prosodic labels. A total of 5,000 sentences (23,500 words) for declarative and interrogative types have been labelled. The accuracy of the trained models for pitch accent, intermediate phrase boundaries and accentual phrase boundaries is 73.40%, 93.20%, and 43% respectively.

1 Introduction

In order to produce natural sounding speech units, many Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) systems incorporate suprasegmental prosodic features, which generally apply to larger units of representation like phrases or the sentence. Some of the intonational aspects that are covered through prosody include pitch accent, phrasing, duration etc. Spoken in natural rhythm, sentences constitute grammatical breaks and accents which lend specific intonational contours to different sentence types. In general, words may contain lexical stress according to grammatical rules (some words in many languages are not stressed at all), at other times, words may be stressed to convey focus. When strung together, sentences spoken naturally, and impacted by extraneous factors such as speaker motivation, mood, speed etc serve to modify prosodic structure of spoken speech in ways that render it natural sounding to human perception. It is therefore important to be able to input these features, along with the orthography to phonemic conversions, into TTS systems, in order to emulate human-like intelligible voices in building Speech to Speech Machine Translation (SSMT) systems.

Section 2 discusses previous studies in Hindi intonation, with a focused perspective on the development of theories on pitch accent and phrase breaks in Hindi sentences. The theories discussed in this section form the basis for the linguistic analysis and annotation of the declarative and interrogative sentences, discussed in later chapters. Section 3 talks about the speech resource used in the building of this dataset, with a bried overview of the labelling framework discussed in section 4. Sections 5 and 6 discuss the manual and automatic approaches used in the development of this dataset, with the results.

2 Background

Hindi belongs to the Indo-European language family and has over 500 million speakers in India. A number of studies exist in literature on Hindi intonation. One of the most pioneer of works was by (Moore, 1965), who analyzed Hindi intonation in terms of three different segmental levels in hierarchical relation to each other: foot, measure and sentence. According to his theory, foot consists of one or more syllables in which pitch rises from beginning to end continuously. Measure is the second level of phrasing in which a focused element is separated from the rest of the sentence. Sentence is the topmost level, which encompasses the entire sentence intonation. (Harnsberger, 1994) makes an observation along similar lines in which he states

Proceedings of the Workshop on Speech and Music Processing (SMP-2021), pages 14–19
co-located with ICON-2021, December 16-19, 2021, Silchar, India.

that there is a rising pitch contour on content words, in which the low part of the rising contour is a low pitch accent and the high part is either a high trailing tone or boundary tone. The other level of phrasing is the sentence. (Féry and Kügler, 2008) talk about the rising pitch contour on each constituent of the data that they have considered and call it the prosodic phrase and relate it to the syntax of the sentence. (Nair et al., 2001), (Dyrud, 2001) suggest, in their work, that Hindi has lexical stress, such that every word has a particular syllable on which prominence is realized. (Sengar et al., 2012), from their investigative studies, put forward the theory that Hindi is an accentual phrase language and that the Accentual Phrase (AP) was the smallest tonally marked prosodic unit, characterised by a rising contour, the observation being similar to that proposed for Bangladeshi Standard Bengali (Khan, 2008), a closely related Indo European language. Their research hypothesized that the intonation pattern of Hindi sentences contained a series of APs, characterised by rising contours (which correlate to pitch patterns within the AP) and that the domain of each AP is marked by prosodic boundaries, which may or may not be equal to a single word. The final tone can be overridden by a falling tone in case of declaratives. The entire sentence constituted an IP (intonational phrase) comprised of many ip (intermediate phrase), characterized by silence junctures, and each ip contains one or more APs.

(Jyothi et al., 2014), through exploratory investigation using non-expert and expert transcribers, concluded that prosodic phrasing was more consistently agreed on between non-expert transcribers amongst themselves and with the expert transcribers (measured by Cohen's kappa coefficient). It was also observed that the degree of agreement in prominence (pitch accent) marking was lower, in both cases.

3 Speech data resource

The speech corpus obtained and used for this work was developed through the Indian Language Technology Proliferation and Deployment Centre[1] under the Technology Development in Indian Languages (TDIL) program, Ministry of Electronics & Information Technology (MeitY), MC&IT, Govt of India. The corpus contains 50 hours of synthetic speech data for both male and female speakers of Hindi. The corpus contains varied sentence kinds (simple, complex) and types (declarative, interrogative, negative, exclamatory etc.) that have been used to choose a varied representation. Sentence units have been selected and extracted from this data for this work.

4 Labelling framework

4.1 Autosegmental-Metrical (AM) model

The Autosegmental-Metrical framework is a mode of intonational structure that is one of the foremost frameworks used for prosody analysis that was built on the tenets of fundamental work by (Pierrehumbert, 1980) with further refinements by (Beckman, 1986), (Pierrehumbert and Beckman, 1988), (Gussenhoven et al., 2004) and others. The term 'autosegmental-metrical', coined by (Ladd, 2008) was based on the Autosegmental and Metrical frameworks of phonology, with the autosegmental tier representing intonation structure and metrical tier the phrasing and prominence. Drawn from Autosegmental Phonology, the proposal by (Pierrehumbert, 1980) was that pitch levels are seen as autosegments for intonational analysis while tones are represented by the pitch accent, phrase tone and boundary tone. The tones High (H) and Low (L) were formalized as being associated with stressed syllables as well as prosodic boundaries. The tones associated with stressed syllables were pitch accents and represented with an asterisk (*) while the boundary tones were marked with a percent (%) sign. In addition, phrasal tones were observed on the intermediate phrase boundaries, which were notated with a hyphen (-). Intermediate phrases were seen to be prosodic units that were larger than the syllable and smaller than the intonational phrase, whose prosodic domain included the whole sentence. Subsequently, this model has been applied to various languages (Japanese (Venditti, 1997), Korean (Jun, 2000), Dutch, German, Italian, French, etc.) with minor modifications.

4.2 Tones and Break Indices (ToBI)

ToBI (Tones and Break Indices) is a system for transcribing the intonation patterns and other aspects of the prosody of originally, English utterances (Beckman and Ayers, 1997). The labelling scheme consists of:

[1] http://tdil-dc.in/index.php?option=com_download&task=showresourceDetails&toolid=268&lang=en

- 6 discrete intonation accents types: H*, !H, L*, L*+H and L+H*.
- 2 phrase accent type: H- and L-
- 4 boundary tones: L-L%, L-H%, H-L% and H-H%
- 4 break levels: 1, 2, 3, and 4
- A HiF0 marker for each intonational phrase

An utterance marked using ToBI labeling conventions contains a number of tiers of information: a tone tier, carrying accent information, a break tier for marking prosodic boundaries and a comment tier for miscellaneous information.

ToBI is a standard transcription system for modeling prosodic events of spoken utterances in different languages. It has become a framework to analyze the intonation system and relationship between prosodic and intonation structures of different languages.

5 Manual Prosodic Labelling

500 simple sentences of the types declarative and interrogative were selected and labelled within the frameworks of the intonational framework observed in previously mentioned studies. This annotation follows the proposed framework that the domain of intonation phrase (IP) is the whole utterance, ending with the boundary tone and which may contain one or more intermediate phrases (ip), demarcated by the phrase tone. The smallest prosodic domain is the Accentual Phrase (AP) containing the pitch accent and this may cover one or more words in length. The default pitch accent is observed to be the rising pitch accent (L*Hp) falling on each content word which starts at the left edge of the AP, rising towards the rightmost edge and declines towards the start of the next AP. The only exception is in the final AP, where the boundary tone may override the final AP decline.

Praat[2], a freely available speech analysis software, was used to identify and mark the prosodic boundaries and tones associated with pitch movements. 3 native Hindi speakers with training in phonetics and phonology, transcribed the data.

5.1 Declarative sentences

In Fig 1, the declarative sentence is divided into 2 prosodic phrases and shows the pitch pattern L*Hp, overridden by the L% boundary tone for declaratives.

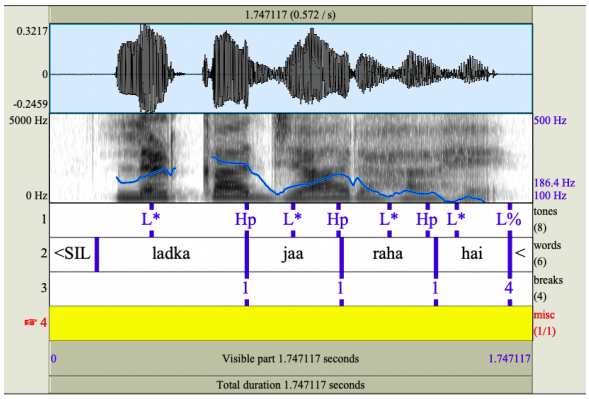

Figure 1: 'ladka ja raha hai'

Figure 1 example: ladka ja raha hai
boy go is-PROG-MASC
The boy is going

This relatively straightforward pattern may be affected by other phenomena that carry information structure, like scrambling and focus. Hindi being a head final, relatively free word order language conveys information by the scrambling of focused constituents to the head of the structure and/or placing a higher pitch accent on the focused element. Focus has also been shown to insert a prosodic break in the post focus word (Moore, 1965) as well as create a compression in pitch range post focus (Harnsberger and Judge, 1996).

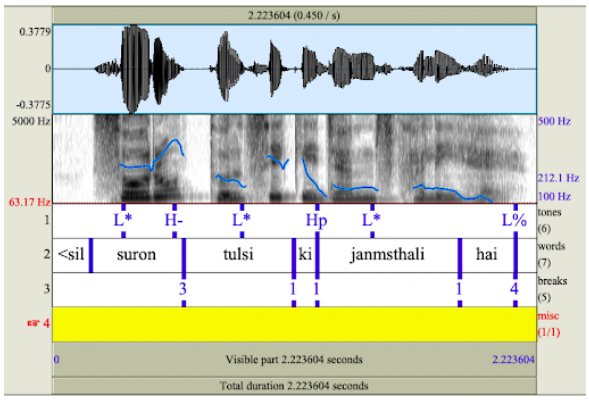

Figure 2: 'Suron Tulsi ki janmasthali hai'

In Fig 2 the object 'suron' contains the focus and is marked by a relative raised pitch accent compared to the utterance level and the postfocal word is lowered. Figure 2 example:
Suron Tulsi ki janmasthali hai
Suron Tulsi of birthplace is
Suron is Tulsi's birthplace

[2] https://www.fon.hum.uva.nl/praat/

5.2 Interrogative sentences

In the interrogative sentence in Fig 3, the intonation pattern follows the L*Hp rising pattern, with a rising H% boundary tone.

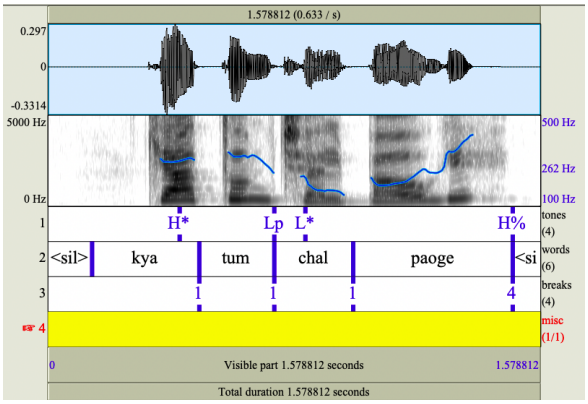

Figure 3: 'kya tum chal paoge'

Figure 3 example: kya tum chal paoge
are you walk able-FUT
will you be able to walk

This was found to be the case in most simple interrogative sentences, except in case of relative higher pitch on seemingly focused elements, as in on the focused 'kahan' (where) in Fig 4.

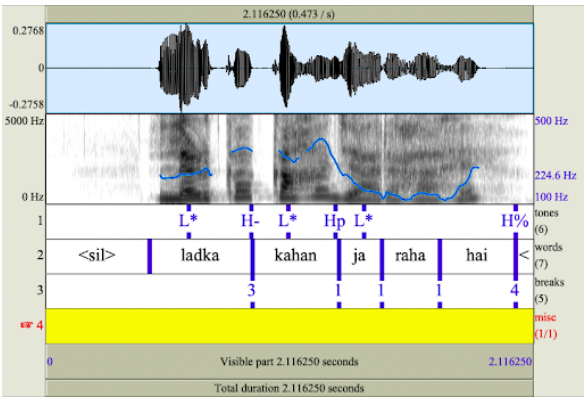

Figure 4: 'ladka kahan ja raha hai'

Figure 4 example: ladka kahan ja raha hai
boy where go is-PROG
where is the boy going

The H tone accompanying this rise and fall was observed to have the downtrend component, associated with another closely related Indo-Aryan language, Bengali (Jun et al., 2014).

Figure 5 example: Kamala chai piya karegi
Kamala tea drink do-HABI
Kamala will drink tea

The downtrend observed in the consecutive H tones in Fig 5 are consistent with the observation

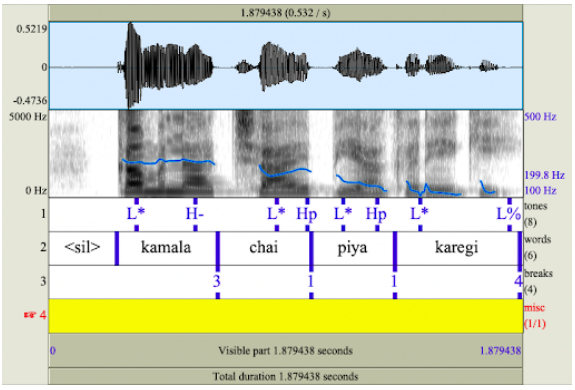

Figure 5: 'Kamala chai piya karegi'

that H tones in successive APs are of a lower pitch than the preceding. Apart from minor effects of microprosody, there were not many deviations observed in this intonation pattern

5.3 Inter-Annotator Agreement

Three linguists (native speakers of Standard Hindi) belonging to Delhi, with familiarity in ToBI annotation conventions, were asked to label the dataset. Some initial training was provided for the analysis as well as the annotation labels presented to them for this research. Initial training consisted of individual instructions as well as calculating inter annotator consistency, and this was carried out iteratively to achieve the desired accuracy. Overall transcriber agreement (calculated using Cohen's kappa) for prosodic breaks was 0.87, and for pitch accents was 0.69.

6 Automatic Prosodic Labelling

The manually labelled sentences developed in the previous section has been used as training data to fine-tune Au-ToBI, an existing automatic prosody labelling toolkit widely available (Rosenberg, 2010), by building newly trained models within their standard specifications. The study is a comparative analysis on the performance of accuracy between pre-existing and newly trained Au-ToBI models for this research. Au-ToBI was particularly selected for its adaptability to ToBI, which had been used as the labelling conventions for the manually annotated data as well.

6.1 Automatic ToBI

Au-ToBI (Rosenberg, 2010) is a publicly available tool that runs on Java, which contains models trained on English sentences to automatically detect and extract prosodic breaks and pitch accents

from spoken utterances . Based on pre-trained models of English, initial detection of pitch accents and phrase boundaries is carried out, based on cues like pitch excursions and silence duration. This is followed by the classification of the phrase boundary tones and type of pitch accent prediction. The classification of prosodic breaks and pitch accents is done as per the ToBI annotation conventions.

6.2 Experiments in Automatic Labelling

This experiment was conducted in two parts. The Hindi manual prosodic dataset developed in section 5 was divided into training and test data in 90:10 ratio. The first experiment on the English model was evaluated with the test data, while the second experiment was conducted with the training and test sets. The experiments are divided into two steps. First, use pre-trained English model detection and classification algorithms in Au-ToBI to generate automatic labels for Hindi utterances and measure accuracy, and second, use manual annotated data to build Hindi prosody models for Au-ToBI.

The pre-processing of this dataset consisted of manually segmentation of sentences into words in the TextGrid files. The transcription was carried out in Devanagari. Since the Hp boundary tone for AP was a distinct feature from the standard ToBI guidelines, the tones in the training sentences were mapped to their corresponding ToBI labels. The "breaks" section in the uploaded files was converted to Au-ToBI format, under the alignment process. This included conversion of "number" to "time" etc. TextGrid and WAV files were named similarly and located in the same folder for use in the training.

Parameters and values for all three tiers "words", "breaks" and "tones" were implemented, along with the Hindi model classifiers and detectors. Multiple command lines were provided for training pitch accent, intonational phrase boundary, intermediate phrase boundary, phrase accent and boundary tone detection and classification models. The default features were selected for the building of these models, using feature extractor and feature classifier. Since the test files used for prediction of Hindi labels came from one speaker, normalization parameters were not used in this set up. The built Hindi Au-Tobi models were evaluated on the test data. The 50 hours TDIL speech corpus was used to extract a further 4,500 declarative and interrogative sentences, split 50:50 for declarative and interrogative sentences.

6.3 Results

Results are output as TextGrid files and in addition to the "words" tier that was present, contains two additional generated tiers named "tones" (for generated pitch accents) and "breaks" (for generated prosodic breaks). The accuracy of the models are demonstrated in the below figure.

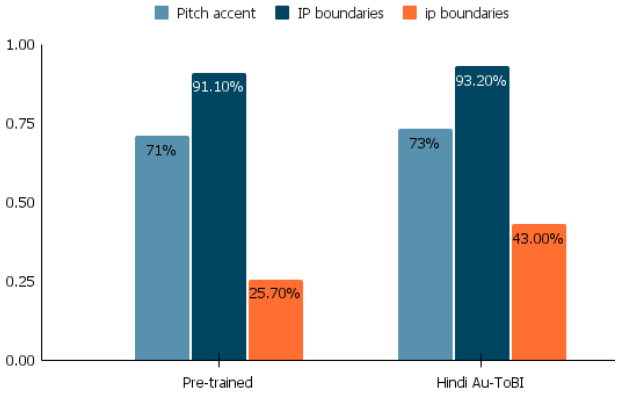

Figure 6: 'Results of pre-trained and newly trained Au-ToBI models'

7 Conclusion and Future Work

The sentences labelled with the prosodic labels are a valuable source of training data for ASR and TTS systems to introduce the naturalness component that is often derived from prosodic elements. This study aims to employ the studies on intonational behavior of simple declarative and interrogative sentences in Hindi done in recent years and develop a semi-automatically annotated labelled dataset that can be used to enhance the prosodic output in SSMT systems for a natural sounding voice. The approach is modeled on the principles of Tones and Break Indices (ToBI) annotation guidelines, and recent research on prosodic boundaries and prominence marking in Hindi and related languages. 2,550 words (500 sentences) are manually annotated and these sentences are used to extend the corpus size up to 5,000 sentences, using Automatic ToBI (Rosenberg, 2010), (Jyothi et al., 2014). The research aims to develop a prosody labelled database for Hindi for training speech models for natural sounding voices. The prosodic labelled dataset and developed Hindi-AuToBi model will be available on GitHub at https://github.com/esha-banerjee/Hindi_Au-ToBI.

Acknowledgments

Atul Kr. Ojha would like to acknowledge the EU's Horizon 2020 Research and Innovation programme through the ELEXIS project under grant agreement No. 731015.

References

Mary E Beckman. 1986. Intonational structure in english and japanese. *Phonology yearbook*, 3:255–309.

Mary E Beckman and Gayle Ayers. 1997. Guidelines for tobi labelling. *The OSU Research Foundation*, 3:30.

Lars O Dyrud. 2001. *Hindi-Urdu: Stress accent or non-stress accent?* Ph.D. thesis, University of North Dakota Grand Forks.

Caroline Féry and Frank Kügler. 2008. Pitch accent scaling on given, new and focused constituents in german. *Journal of Phonetics*, 36(4):680–703.

Carlos Gussenhoven et al. 2004. The phonology of tone and intonation.

James D Harnsberger. 1994. Towards an intonational phonology of hindi. *Ms., University of Florida*.

James D Harnsberger and Jasmeet Judge. 1996. Pitch range and focus in hindi. *The Journal of the Acoustical Society of America*, 99(4):2493–2500.

Sun-Ah Jun. 2000. K-tobi (korean tobi) labeling conventions. *Speech Sciences*, 7(1):143–170.

Sun-ah Jun et al. 2014. The intonational phonology of bangladeshi standard bengali.

Preethi Jyothi, Jennifer Cole, Mark Hasegawa-Johnson, and Vandana Puri. 2014. An investigation of prosody in hindi narrative speech. In *Proceedings of Speech Prosody*, volume 7, pages 623–627.

Sameer Ud Dowla Khan. 2008. Intonational phonology and focus prosody of bengali (phd thesis). *University of California, Los Angeles: University of California*.

D Robert Ladd. 2008. *Intonational phonology*. Cambridge University Press.

Robert Ripley Moore. 1965. *A study of Hindi intonation*. University of Michigan.

Rami Nair et al. 2001. Acoustic correlates of lexical stress in hindi. In *Linguistic Structure and Language Dynamics in South Asia—papers from the proceedings of SALA XVIII roundtable*.

Janet Pierrehumbert and Mary Beckman. 1988. Japanese tone structure. *Linguistic inquiry monographs*, (15):1–282.

Janet Breckenridge Pierrehumbert. 1980. *The phonology and phonetics of English intonation*. Ph.D. thesis, Massachusetts Institute of Technology.

Andrew Rosenberg. 2010. Autobi-a tool for automatic tobi annotation. In *Eleventh Annual Conference of the International Speech Communication Association*.

Anuradha Sengar, Robert Mannell, et al. 2012. A preliminary study of hindi intonation. In *Proceedings of SST*.

Jennifer J Venditti. 1997. Japanese tobi labelling guidelines.

Multitask Learning based Deep Learning Model for Music Artist and Language Recognition

Yeshwant Singh **Anupam Biswas**
Department of Computer Science and Engineering,
National Institute of Technology Silchar, Assam, India, 788010
{yeshwant_rs, anupam}@cse.nits.ac.in

Abstract

Artist and music language recognitions of music recordings are crucial tasks in the music information retrieval domain. These tasks have many industrial applications and become much important with the advent of music streaming platforms. This work proposed a multitask learning-based deep learning model that leverages the shared latent representation between these two related tasks. Experimentally, we observe that applying multitask learning over a simple few blocks of a convolutional neural network-based model pays off with improvement in the performance. We conduct experiments on a regional music dataset curated for this task and released for others. Results show improvement up to 8.7 percent in AUC-PR, similar improvements observed in AUC-ROC.

1 Introduction

Music is a universal language that we innately understand. It can influence or induce new emotions in the listeners. Artists project their emotions and feeling onto their music that is felt and observed in the music. Artist recognition of a music recording is an active area of research in music information retrieval (MIR) (Mesaros et al., 2007; Sharma et al., 2019; Hu et al., 2021) and has various applications.

Artist recognition is crucial in the areas of music index, retrieval, and recommendation. The digitization of the music industry and music streaming platforms have created large volumes of digital music that need to be processed and stored on a large scale. This has reignited the research in the music domain. Recognition of the artist of a song is crucial for these music streaming platforms. We also have our favorite artists, whom we search for on these streaming platforms as music listeners, and it shows how vital artist recognition is.

Machine learning-based approaches treat this problem as a multi-label classification problem. There have been many recent deep learning-based techniques that perform very well for this task. The approach in these techniques is to use variants of spectrogram and train the deep neural network model on that visual representation (Yu and Slotine, 2009; Kalantarian et al., 2014; Wu et al., 2018). Some techniques have used raw waveforms to train sequence-based models. These techniques have revealed that related tasks specific noise filtering can boost the overall generalization of deep learning-based models for music-related tasks.

Surprisingly, given techniques by researchers do not leverage the shared representation learned by multitask learning of related tasks. In this paper, we propose a multitask learning-based model for artist recognition that leverages the shared representation learned from the related task of music language recognition. The results show improvement over single-task learning. We have used multitask learning with convolutional neural networks (CNN) for artist recognition, and we observed improved performance.

Multitask learning is a machine learning paradigm in which related tasks are trained together using the same model which shares bottom layers (in neural networks) among the related tasks. The training signals (gradients) from different related tasks force the model to learn more generalized data representation by filtering out noise for each related task (Böck et al., 2019; Zeng et al., 2019). Alternatively, we can say the knowledge learned for a task helps in the performance over another related task. In our case, we use two related tasks of artist recognition and music language recognition, where artist recognition is the primary task for leveraging shared representation from multitask learning. Music recordings spectrograms are used as an input for the model, and corresponding artists and language are predicted as an output by the model.

This paper is organized as follows: Section 2

Proceedings of the Workshop on Speech and Music Processing (SMP-2021), pages 20–23
co-located with ICON-2021, December 16-19, 2021, Silchar, India.

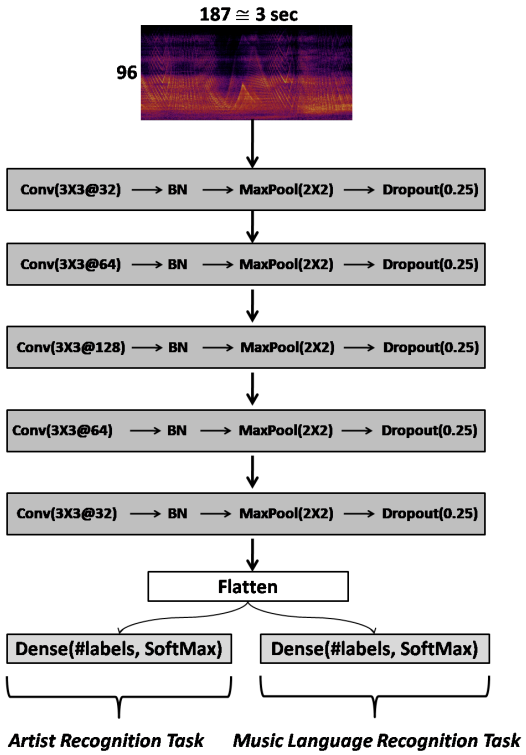

Figure 1: Proposed Multitask learning based model architecture.

presents the proposed approach. The dataset, experimental setup is summarized in Section 3. In Section 4, we present the results and discussions of our findings. Finally, we conclude in Section 5.

2 Proposed Method

Our proposed approach is a model based on convolutional neural networks (CNN) similar to VGG architecture. CNNs are very popular in the computer vision domain. Here we have chosen CNN because the spectrogram representation of music samples can be treated as an image, and CNNs can extract relevant knowledge from them. This spectrogram approach is not new and has been used in the past with traditional machine learning. However, advances in computer vision have given new avenues in the spectrogram representation for music tasks.

The architecture of our model is shown in Figure 1. A mel-spectrogram taken as a 2D-tensor with (187x96) is taken as an input to the network. The mel-spectrogram consists of 187 frames which correspond to 3 seconds of music data and 96 mel-bands. The extraction of the spectrogram from the music recordings is discussed in Section 3.2.1.

A batch of spectrograms with size 64 is passed to five blocks of CNN layers. Each block consists of a CNN layer along with batch normalization, max-pooling, and dropout. The kernel size of CNN layers is fixed to (3x3), and the numbers of channels are 32, 64, 128, 64, and 32, respectively, with stride 1 of CNN layers in five blocks. The non-linearity in the CNN is set ReLU. After each CNN layer, batch normalization (BN) is applied to normalize the weights during training. Following BN, max-pooling is applied with a pool size of (2x) and stride of (2x2). Lastly, a dropout layer with a 0.25 drop rate is applied.

After these CNN layer blocks, a flatten layer is used to compress the output shape from CNN blocks to the 1D tensor. Dense layers then use this tensor corresponding to each related task (two in our case, namely artist recognition, and music language recognition). Each dense layer has the number of units equal to the number of labels in the corresponding task (64, 17 in our tasks). The softmax activation function is used in these dense layers' outputs to get the probability distribution of labels relating to the tasks.

These are 193k trainable parameters in the top five blocks of our proposed model. The parameters in the output layers are dependent on the number of parameters in the previous layer and the number of labels in the corresponding task. The model shares the five blocks of the model between two related tasks. The representation learning during the training of these two tasks forces the model to generalize better than it would have been training on a single task. The loss used for training the model is a weighted sum of losses from individual tasks. Categorical cross-entropy is used as a loss for both tasks. The weighted sum for overall loss function:

$$L = L_{artist} + \alpha L_{lang}$$

Here, L_{artist} and L_{lang} denote the corresponding loss for artist and music language recognition tasks, respectively. α represents the hyper-parameter for the weightage given to the loss of music language recognition task in the overall loss. Setting it to zero disables the multitask learning and is done when we are done with training the network. The choice of α is discussed in Section 3.2.4.

3 Experiments

3.1 Dataset Description

The model is trained on a dataset prepared for regional music by us (Singh, 2021). The dataset consists of 17 languages: Hindi, Gujarati, Marathi, Konkani, Bengali, Oriya, Kashmiri, Assamese, Nepali, Konyak, Manipuri, Khasi & Jaintia, Tamil, Malayalam, Punjabi, Telugu, Kannada.

For each language, four artists are chosen (two male and two female), and for each artist, five songs are collected. The artists are chosen considering the veteran and contemporary artists. So, two out of four artists are veteran performers, and the remaining two are modern artists. Overall, there are 68 artists and 340 music songs with 23.2 hours of duration.

3.2 Experimental Setup

3.2.1 Preprocessing

A preprocessing step is performed over the music recordings before feeding them to the model. Music recordings are converted to mel-spectrograms of 3-sec segments. These 3-sec segments are created from two 1 minute long pieces taken out from each music recording. The spectrograms are generated from resampled waveforms with 96 mel-bands. The shape of the spectrogram is $(t \times 96)$, where t is the number of frames (proportional to time). Librosa library (McFee et al., 2015) is used for the extraction of mel-spectrogram from 3-sec music segments.

3.2.2 Evaluation Metrics

We have used average precision (AP) as an evaluation metric. It is commonly used in multi-label classification tasks. It is the weighted average of precision values across different recall values, or it can be said as the area under the precision-recall curve (AUC-PR). We also report another evaluation metric called the area of the receiver operating characteristic curve (AUC-ROC).

3.2.3 Training

The training of the model is performed with a batch size of 32. As we take 3-sec long music data for generating spectrogram, they must be batched together to speed up the training process. Tensorflow is used as a deep learning framework for building and training our model. This framework handles the batching of spectrograms.

We split the data into three splits of 80, 10, and 10 percent of samples for training, validation, and testing purposes. The splits are ensured to happen at song level instead of segment level to ensure that segments from a few songs are not just present in validation and testing datasets, preventing data leakage. Adam optimizer is used for training the model with a 0.001 learning rate.

3.2.4 Multitask Learning Weighted Loss Function

The overall loss function is defined as a weighted sum of the losses of individual tasks. That is $L = L_{artist} + \alpha L_{lang}$. The hyperparameter α influences the contribution of L_{lang} in the overall loss. We tried different values for α for training the model on the mentioned dataset to get the optimal value. We found following formula works well in our scenario:

$$\alpha = \frac{N_{artist}}{N_{lang}}$$

Here, N_{artist} and N_{lang} are the number of labels in the artist and music language recognition tasks. It can be said that α balances the numbers of labels in the given tasks. The α for our experiments is computed using the above formula.

4 Results

We performed multiple experiments over the described dataset in Section 3.1. Baseline and Multitask are two models selected and trained for two tasks. Both models are the same architecture except for the final dense layer. The Baseline model has a single dense layer having units equal to the labels in the given task. While in the multitask model, there are two dense layers side by side for each task, having units equal to two task labels. Comparing the result of baseline and multitask models allow us to observe the impact of multitask learning on the given task with the help of additional related task. The comparative analysis is presented in Table 1, which shows models performance across different configurations.

We report that the multitask model shows improvement over the baseline model for both tasks and can be observed in both evaluation metrics. On the curated dataset, we observe an increase of 4.43 percent for artist recognition in the AUC-ROC metric, while the AUC-PR metric recorded a 7.48 improvement. For the music language recognition task, improvements are even further. 5.55 improvement is observed in the AUC-ROC and 8.69 for

Model type	N_{artist}	N_{lang}	# songs	# 1-min segments	# 3-sec segments	AUC-ROC (%)	AUC-PR (%)
Baseline	68	17	340	680	1360	70.45	39.78
Multi-task I	68	17	340	680	1360	74.88 (**+4.43**)	47.26 (**+7.48**)
Multi-task II	68	17	340	680	1360	76.00 (**+5.55**)	48.47 (**+8.69**)

Table 1: Baseline and Multitask models performance reported in AUC-ROC and AUC-PR metrics. N_{artist} and N_{lang} represents the number of labels in artist recognition and music language recognition tasks, repectively. **(+x.xx)** represents the increase in the given evaluation metric compared to the performance of Baseline model.

the AUC-PR metrics. We observe from these experiments that the performance can be improved by adding more related tasks and increasing the number of data samples.

5 Conclusions

In this paper, we present the importance of artist recognition from the perspective of music streaming platforms for storage, indexing and music information retrieval tasks. It is crucial to building a more generalized system by these platforms.

Towards building a more generalized system, we observe that multitask learning can help achieve a more generalized system by leveraging the model's representation across different related tasks. We propose a multitask learning model for artist and music language recognition tasks. Experiments depict that the multitask learning approach improves the performance of the single-task baseline model.

Acknowledgement

This research was funded under the grant number: ECR/2018/000204 by the Science & Engineering Research Board (SERB).

References

Sebastian Böck, Matthew EP Davies, and Peter Knees. 2019. Multi-task learning of tempo and beat: Learning one to improve the other. In *ISMIR*, pages 486–493.

Shichao Hu, Beici Liang, Zhouxuan Chen, Xiao Lu, Ethan Zhao, and Simon Lui. 2021. Large-scale singer recognition using deep metric learning: an experimental study. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

Haik Kalantarian, Nabil Alshurafa, Mohammad Pourhomayoun, Shruti Sarin, Tuan Le, and Majid Sarrafzadeh. 2014. Spectrogram-based audio classification of nutrition intake. In *2014 IEEE Healthcare Innovation Conference (HIC)*, pages 161–164. IEEE.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer.

Annamaria Mesaros, Tuomas Virtanen, and Anssi Klapuri. 2007. Singer identification in polyphonic music using vocal separation and pattern recognition methods. In *ISMIR*, pages 375–378.

Bidisha Sharma, Rohan Kumar Das, and Haizhou Li. 2019. On the importance of audio-source separation for singer identification in polyphonic music. In *INTERSPEECH*, pages 2020–2024.

Yeshwant Singh. 2021. Regional music dataset. https://github.com/yeshwantsingh/regional_dataset. (Accessed on 11/14/2021).

Yu Wu, Hua Mao, and Zhang Yi. 2018. Audio classification using attention-augmented convolutional neural network. *Knowledge-Based Systems*, 161:90–100.

Guoshen Yu and Jean-Jacques Slotine. 2009. Audio classification from time-frequency texture. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1677–1680. IEEE.

Yuni Zeng, Hua Mao, Dezhong Peng, and Zhang Yi. 2019. Spectrogram based multi-task audio classification. *Multimedia Tools and Applications*, 78(3):3705–3722.

Comparative Analysis of Melodia and Time-Domain Adaptive Filtering based Model for Melody Extraction from Polyphonic Music

Ranjeet Kumar Anupam Biswas Pinki Roy Yeshwant Singh
Department of Computer Science and Engineering
National Institute of Technology Silchar, Assam, India
{ranjeet_rs, anupam, pinki, yeshwant_rs}@cse.nits.ac.in

Abstract

Among the many applications of Music Information Retrieval (MIR), melody extraction is one of the most essential. It has risen to the top of the list of current research challenges in the field of MIR applications. We now need new means of defining, indexing, finding, and interacting with musical information, given the tremendous amount of music available at our fingertips. This article looked at some of the approaches that open the door to a broad variety of applications, such as automatically predicting the pitch sequence of a melody straight from the audio signal of a polyphonic music recording, commonly known as melody extraction. It is pretty easy for humans to identify the pitch of a melody, but doing so on an automated basis is very difficult and time-consuming. In this article, a comparison is made between the performance of the currently available melody extraction approach that is state-of-the-art Melodia and the technique based on time-domain adaptive filtering for melody extraction in terms of evaluation metrics introduced in MIREX 2005. Motivating by the same, this paper focuses on the discussion of datasets and state-of-the-art approaches for the extraction of the main melody from music signals. Additionally, a summary of the evaluation matrices based on which methodologies have been examined on various datasets is also present in this paper.

1 Introduction

In recent times, the music business and music suppliers such as Google, Spotify, and others have seen significant growth. By that time, the music business had also been reorganized from the cylinder age to the digital era, resulting in the current scenario where consumers may acquire millions of songs on personal phones or via cloud-based services, as well as the future. It is necessary to cope with the enormous quantity of music to search for and recover the required record effectively. At the moment, the primary issue of music suppliers is to categorize the vast number of songs available on the market based on their many components, such as rhythm, pitch, melody, and so forth. When we need to identify a particular soundtrack, we often reproduce the melody. There is a great deal of continuous progress in audio processing, which may assist customers in interacting with the songs via their sound component. Music transcription is the act of translating an aural input into a detailed description of all the notes being performed (Gómez et al., 2012). It is a task that a competent music student should be able to do very efficiently. It has, on the other hand, long been the topic of computer research. Despite this, owing to musical harmony's intricate and intentionally overlapping spectral structure, it has proved to be very difficult to achieve (Dressler, 2011).

> "It is melody that enables us to distinguish one work from another. It is melody that human beings are innately able to reproduce by singing, humming, and whistling. It is melody that makes music memorable: we are likely to recall a tune long after we have forgotten its text.". (Hofmann-Engl, 1999)

The definition given by Poliner et al. (2007) is one of the most frequently cited in the literature and is one of the most widely used:

> "roughly speaking, the melody is the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the 'essence' of that music when heard in comparison".

The melody is restricted to a single sound source throughout the work being examined, which is deemed the most prominent instrument or voice in the mix (Yeh et al., 2012; Klapuri, 2004). When

Proceedings of the Workshop on Speech and Music Processing (SMP-2021), pages 24–32
co-located with ICON-2021, December 16-19, 2021, Silchar, India.

polyphonic music is played, the melody is the single or monophonic pitch grouping that an audience may replicate during any moment in time to whistle or hum a piece of the music, so a large number of listeners would perceive as the 'essence' of the music when the music is played in contrast (Reddy and Rao, 2018). This concept is now susceptible to a great deal of subjective interpretation since different members of an audience may hum other portions in the aftermath of listening to a comparable piece of music.

Because of these vast number of various interpretations of melody and polyphony available, it becomes easier to categorize melody retrieval as a signal processing task than it was before.: We wish to correctly predict the series of $f0$ values that correlate to the voices or devices that are prominently featured in a clip of polyphonic music Aside from that, we must approximate the periods during which this voice is absent from the mixture (a challenge also termed as the "voicing detection" issue) (Salamon et al., 2013). While this job may seem virtually insignificant to a human listener, many of us are capable of singing along to the melodies of our favorite songs even if we have no formal musical training.

It is necessary to automatically acquire a series of frequency values of the dominant melodic line for polyphonic audio signals in order to complete the melody extraction job successfully Fig. 1. As defined by the American Institute of Music, polyphonic music is music in which at least two notes may be played at the same time on a variety of instruments (for example, bass, voice, and guitar) or on a single instrument that can play numerous notes in a single period (for example, the piano). A listener may imitate the tunes even if he or she does not have any musical training. However, when we try to automate this process, things become a little more complicated primarily due to two reasons: First, a polyphonic music signal is generated up of all the sound waves from all the devices in the track superimposed on each other. In the spectral content of the signal, various sources' frequency components overlap, making it difficult to assign particular energy levels in specific frequency bands to separate instruments' notes. Second, even after obtaining a pitches-based representation of the audio stream, we must still determine the pitch values that correspond to the dominant melody in the audio stream.

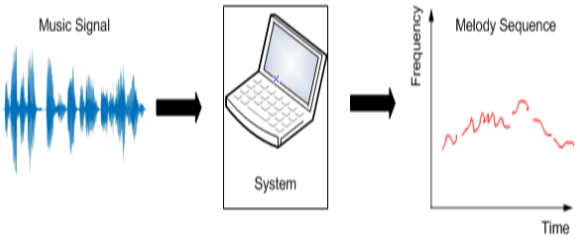

Figure 1: Melody extraction from audio signal of polyphonic music.

The task of automated melody extraction is common in the area of Music Information Retrieval (MIR). There have been a plethora of methods developed for the extraction of melodies from polyphonic music. Based on the methods used to develop them, these algorithms can be classified namely Source separation-based approach and salience-based approach (Salamon and Gómez, 2012). On the other hand, some methods do not fall under any of these categories. Algorithmic technique which is categorised as data-driven approaches, the power spectrum is directly send to deep neural network based machine learning system, which attempts to determine the melody frequency from each frame.

1.1 Salience-based approach:

Following the principles established by Scheirer Scheirer (2000), melody extraction approaches based on salience function are founded on the concept of "understanding without separation." Primarily, the following steps are required in melody extraction: The majority of the time, in preprocessing phase, to increase the melodic content of a composite signal, filtering is applied to it (**?**). Aspects of the music signal's time-domain samples are divided into frames of similar length and translated to the spectrum domain during the spectral representation and processing step. To follow the f_0 transitions in the dominant instrument, the selected window widths give sufficient frequency resolution to differentiate sinusoidal partials (Goto, 2004; Hsu and Jang, 2010). Most techniques handle the modified signal's raw spectral peaks. To put it simply, a salience function is just an evaluation of the salience of pitch values over time that is dependent on the recently identified partial peaks. Candidate melodies for the melody f_0 are considered to

be the peaks in the salience function (Klapuri, 2004). It is necessary to discover the salience peaks that correlate to actual melody peaks like the last stage in this process. The majority of algorithms directly monitor the melody peaks from the salience function.

1.2 Source separation-based approach:

It is feasible to distinguish the source responsible for the fundamental frequency from the remainder of the composite signal by using several source separation techniques (Ryynänen and Klapuri, 2008). By considering the polyphonic signal's power spectrum as the sum of lead and harmony voices, it was suggested to use source separation-based melody extraction to extract melodies (Durrieu et al., 2010). It is suggested to characterise lead vocals using a source-filter-based paradigm, and to describe accompaniment as a sum of arbitrary sources with different spectral shapes, respectively. For the source-filter model, two new models are proposed: the "Smooth-Instantaneous Mixture Model (SIMM)" and the "Smooth Gaussian-Scaled Mixture Model (SGSMM)". The SIMM is used to represent the dominating voices, while the SGSMM is used to represent the accompaniment. The expectation maximization approach is used to estimate the system model parameters. In order to determine the singer's f_0 contour from the tape, Tachibana et al. (2010), employed the temporal variability of the song.

1.3 Data-driven approach:

In contrast to data-driven strategies, which have only been examined seldom, most algorithms, as we have previously stated, are based on the salience function and source separation from music mixing. However, in recent years, this sort of method has emerged as a promising new field of investigation (Park and Yoo, 2017; Su, 2018). In order to visualise the distribution of energy in a music signal across time and frequency, spectrograms are used in preprocessing step. To minimise the leakage that happens during spectral transforms hanning window is used. The majority of researchers chose STFT because it gives time-based frequency information regarding signals whose frequency components fluctuate over time. When it comes to music recordings, a time-frequency representation is provided by the Constant—Q Transform (CQT). In compared to STFT, CQT is virtually the best fit, and the resultant representation is very low in dimensionality as a consequence. (Kum et al., 2016; Rao and Rao, 2010) devise the concept of multi-column deep neural networks for the extraction of musical notes As a classification-based technique, Using the aforementioned methodology, scientists trained each neural network how to correctly anticipate a pitch label. Author combined the output of networks and post-processed it using a hidden Markov model to deduce the melodic contour, which they labelled as a result of their efforts.

Some of the state-of-the-art approaches for extracting the melody from music signals are described in detail in this paper, which also demonstrates how these techniques are instantly applicable to MIR research. Further results of these models upon well-known datasets are also analyzed. The following is the outline for the rest of the paper. Section II describes the experimental setup in which melody extraction approach has been discussed and including dataset and performance measures are also being discussed here. Results of the assessment are reported in Section III, followed by a result analysis. finally conclusions in section IV.

2 Experimental setup

2.1 Models:

This section provides a quick overview of some of the state-of-art ways for extracting melody from a piece of music data.

2.1.1 Melodia

Salamon and Gómez (2012) proposed a model which is very popular in the filed of MIR in which he uses the Pitch Contour Characteristics to extract the melody from polyphonic music. In this model, Contour characterization and its use for melodic filtering are the most significant contributions. As seen in Fig. 2, this technique is composed of four major components that work together.

Sinusoid Extraction: Three states are present in this stage: filtering, spectral transform, and sinusoid frequency correction. In this case, an loudness filter (equal) has been applied to increase frequencies that the ear of human is more sensitive to. Then the ShortTime Fourier Transform (STFT) applied and taken small hop size to improve F0 tracking while creating pitch contours. The FFT's bin frequencies constrain the position of spectral peaks, resulting in high peak frequency estimate errors for low frequencies. For overcome this

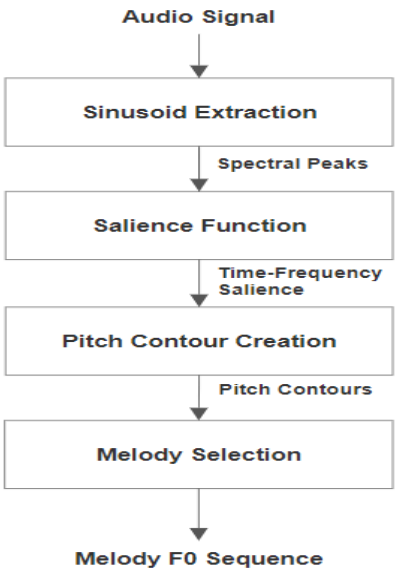

Figure 2: Block diagram of Melodia.

issue, they have calculated peak's instantaneous frequency (f_i) and amplitude by using phase spectrum.

$$\hat{f}_i = (k_i + \kappa(k_i))\frac{f_s}{N} \quad (1)$$

Salience Function: To illustrate the change in pitch salience over time, a salience function is constructed from the spectra that have been extracted and plotted against time. When this function is used, the peaks create the F0 candidates for the main melody. In this model, harmonic summation is used to calculate salience. An integer multiple (harmonic) of a frequency's salience is calculated as the sum of the weighted energies present there. The summing solely uses the spectral peaks, excluding spectral values with masking or noise. Salience function $S(b)$ at each frame can be evaluated using following definition:

$$S(b) = \sum_{h=1}^{h}\sum_{i=1}^{I} e(\hat{a}_i).g(b,h,\hat{f}_i).(\hat{a}_i)^{\beta} \quad (2)$$

where, β represents the parameter of magnitude compression and $g(b,h,\hat{f}_i)$ defines the weighting function.

Pitch Contours: It is then determined which peaks at each frame are probable melody F0 possibilities based on the salience function that was produced. Firstly, non-salient peaks are filtered out to minimize the noise contours creation. In order to determine the most appropriate parameters for contour formation, they compared contours created from various excerpts to the melodic ground truth of the excerpts and assessed them in terms of pitch accuracy and voicing accuracy. After contours creation, the main challenge is to finding the specific contours which belongs to pitch. It is necessary to establish a set of contour attributes that will be utilised to assist the system in picking melodic contours in order to do this.

Melody selection and extraction: As an alternative to picking melody contours, they formulate this issue as a contour filtering problem, with objective being to filter out any contours that are not melodic. The job of detecting when the melody is there and when it is not is referred to as voicing detection. In the last stage, choose the peaks that are associated with the primary melody from among the remaining shapes.When a frame has more than one contour, the melody is selected as the peak of the most salient contour. A frame without a contour is considered unvoiced.

2.1.2 Model based on Time-Domain Adaptive Filtering

Model developed by Reddy and Rao (2018) is basically based on time-domain adaptive filtering. The suggested approach extracts the voice melody in phases from polyphonic music. The difference in excitation intensity between the vocal and non-vocal regions of the music signal distinguishes the vocal from the non-vocal regions. The vocal regions are then split into a sequence of notes by detecting their onsets in the composite signal's frequency representation, which is then used to segment the sequence of notes further. Individual voice note melodic contours may be obtained by using adaptive zero-frequency filtering in the time domain.

In order to distinguish vocal and non-vocal areas, the music signal is first passed through a zero-frequency filter (ZFF), after which the vocal regions are segmented into a series of notes is created. According to the original ZFF approach, the monaural speech signal with a single excitation source is utilised to extract the F0 from the signal before further processing. The mean subtraction filter is designed using the time domain autocorrelation function to produce the average pitch period, which is obtained using the time domain autocorrelation function Music, on the other hand, is a composite signal that is made up of a number of

different pitched sources. The autocorrelation function cannot be used to determine the singer's resonance frequency or average pitch period because it is too complex. The next step is to detect the voiced and unvoiced segments.

Voiced and Unvoiced segment detection: Because the ZFR attenuates the vocal tract resonances to a large extent, passing the signal through it twice has considerably highlighted the source signal. When comparing the vocal source to the other sources in a polyphonic music signal with a lead voice, it is the vocal source that is most prominent. It is thus possible to identify the vocalic areas by analysing the strength of excitation (SOE) (Salamon et al., 2014). A consequence of the vocals' dominance feature is that the ZFF signal contains a significant amount of energy in the voiced areas and a very low amount of energy in the regions which is unvoiced. In order to determine the intensity of the excitation contour, the ZFF signal's slope at the instants of zero crossings of the ZFF signal is calculated.

Voiced Note Onset detection: The melody source's fundamental frequency fluctuates greatly between notes. In order to produce an accurate F0 for the lead voice, a simple mean subtraction filter is insufficient. By recognising note onsets, the voiced segments discovered before may be further split into voiced note-like regions. Signal parameters such as short-time energy, spectral magnitude, phase spectrum, etc. exhibit considerable changes at an onset. Using a low-pass filtering technique, the difference between the current frame and prior frames of a detection function, which are exponentially weighted, is calculated by

$$y(n) = F(n) - \sum_{a=1}^{A} \frac{F(n-a)}{a} \qquad (3)$$

Where the onset detection functions are represented by $F(n)$ and a represents the weighting factor.

Melody detection: A polyphonic music signal's lead voice melody may be found by removing the trend in the ZFR output of each note segment adaptively using a mean subtraction window length that corresponds to average pitch period of the lead voice in the segment. As a final step, each segmented note is subjected to Zero Frequency Filtering with a trend elimination window based on its average pitch period. In order to get the melody of the lead voice, the inverse of the difference between consecutive GCI's is calculated using the note segments that represent the GCI's.

Table 1: Dataset description.

Name	*Sample Rate (in KHZ)*	*Number of clips*
MIREX 2005	*44.1*	*13*
ADC 2004	*44.1*	*20*
IITKGP HPMD	*44.1*	*28*

2.2 Dataset:

The state-of-the-art to evaluating the melody of an audio clip have been described in previous section. In this part of article, we will cover datasets that are used to analyze the aforementioned approaches. In the form of time–frequency pairings, the datasets include music snippets as well as the accompanying melodic ground truth. Specifically, the ADC2004, Mirex05TrainFiles, and IITKGP HPMD which each included 20, 13 and 28 excerpts, were employed, respectively.

ADC 2004: This dataset contains four clips from each of the following genres: pop, jazz, daisy, opera, and MIDI (Musical Instrument Digital Interface). This dataset comprises of twenty audio clips were captured at a sample rate of 44,100 Hz for about 20 seconds each using pulse code modulation (16-bit) and a length of around 20s.

MIREX05: MIDI datasets with genres such as rock, pop, jazz, and classical piano are the most often utilised in melody extraction. This database contains 20–30 s segments of single channel 16-bit 44,100 Hz sampling .

IITKGP HPMD: (Reddy and Rao, 2018) Hindustani Classical Polyphonic Music recorded by professional musicians, which are known as IITKGP HPMD. The dataset contains 28 music clips, each of which has an average length of 30 seconds and is performed by both male and female musicians.

Table 1 lists all the datasets that were utilised in the assessment process.

2.3 Performance measures:

In order to extract the melody, techniques must perform two objectives: first to estimate which part of audio has melody and which part does not contain melody (voicing detection) and secondly, to

predict the proper predominant fundamental frequency as melody (pitch estimation). A melody extraction method usually outputs two columns, the first with fixed interval timestamps generally of 10ms and with f_0 values indicating the algorithm's pitch estimate for the melody at each timestamp in the second column. Additionally, for each frame, the algorithm specifies whether or not it believes the melody is present or missing in that particular frame. For frames when the melody is judged to be missing, this is usually expressed in a third output column or by returning an f_0 value with a negative sign. It is possible for algorithms to report a pitch label even in frames where algorithm assume the pitch is missing i.e., unvoiced frames, which is helpful for evaluating the performance of the algorithm. The accuracy of a pitch estimation algorithm may be evaluated independently of the quality of its voice detection method in this way. In another word, voicing detection mistakes do not affect pitch estimation accuracy.
The output of an algorithm is compared with the ground truth of an audio excerpt in order to assess its performance for a particular audio clip. Ground truth files are identical to output files, but they include the proper sequence of f_0 values indicating the melody of the audio clip. A monophonic pitch tracker is used to create the ground truth on the excerpt's solo melody track. In other word, every song we evaluate requires a multi track recording. In order to evaluate an algorithm, it is necessary to compare its output on a frame-by-frame basis to the ground truth file supplied by the ground truth file. The algorithm should report that it has identified the lack of melody in unvoiced frames in the ground truth. It is anticipated that the method will provide a frequency value that is identical to the one found in the ground truth for voiced frames. Some of the performance metrics frequently employed for melody extraction methods have been addressed in this section.

We calculate five global metrics based on this frame-by-frame comparison that evaluate various elements of the algorithm's performance for the audio sample in the issue. These metrics were introduced in MIREX 2005 and are now often used to assess melody extraction methods.

The uni-dimensional estimated melodic pitch frequency sequence and ground truth frequency sequence, represented by the vectors f and F, respectively (Kumar et al., 2020, 2019). The voicing indication vector is denoted by the v, whose i^{th} element $v_i = 1$ when the i^{th} frame is judged to be voiced (i.e., when a melody is present in the frame), with matching ground truth values V for the other elements in the vector. Unvoicing indications are expressed by the notation $\bar{v_i} = 1 - v_i$.

Voice Recall (VR): The algorithm's estimated voiced frame ratio to the ground truth melodic frame ratio. i.e., Frames that are really labeled as melodic/melodic frame based on ground truth.

$$VR = \frac{\sum_i v_i V_i}{\sum_i v_i} \tag{4}$$

Voicing False Alarm (VFA): The ratio of frames that were incorrectly assessed as melodic frames by the algorithm to frames that were labeled as non-melodic frames in ground truth.

$$VFA = \frac{\sum_i v_i \bar{v}_i}{\sum_i \bar{v}_i} \tag{5}$$

Raw Pitch Accuracy (RPA): The proportion of properly pitched frames compared to frames that are judged to be unpitched.

$$RPA = \frac{\sum_i v_i \tau[\zeta(f_i) - \zeta(F_i)]}{\sum_i v_i} \tag{6}$$

where, threshold feature is describe by τ and can be defined as:

$$\tau[a] = \{\, 1\ if \mid a \mid < 500 if \mid a \mid > 50 \tag{7}$$

Function ζ maps a frequency (Hz) to a perceptually motivated axis in which each semitone is split into a hundredth of a cent. A significant value number of cents may be used to indicate frequency over a reference frequency f_{ref}.

$$\zeta(f) = 1200 \log_2(\frac{f}{f_{ref}}) \tag{8}$$

Raw Chroma Accuracy (RCA): RCA works in the same way as the RPA, except it doesn't take into account the octave mistake (a common error made during melody extraction). i.e., The ground truth and approximated f_0 sequences are both assigned to a single octave.

$$RCA = \frac{\sum_i v_i \tau[\langle \zeta(f_i) - \zeta(F_i) \rangle_{12}]}{\sum_i v_i} \tag{9}$$

Table 2: Evaluation result achieved by Melodia for various testset.

Testset	VR	VFA	RPA	RCA	OA
ADC 2004	0.83	0.18	0.64	0.80	0.74
MIREX05	0.76	0.24	0.57	0.70	0.61
IITKGP HPMD	0.77	0.27	0.75	0.86	0.76

Table 3: Evaluation result achieved by Time-Domain AdaptiveFiltering-Based Method for various testset.

Testset	VR	VFA	RPA	RCA	OA
ADC 2004	0.87	0.11	0.65	0.83	0.79
MIREX05	0.80	0.20	0.62	0.73	0.65
IITKGP HPMD	0.83	0.30	0.71	0.86	0.73

Where,

$$\langle a \rangle_{12} = a - 12\lfloor \frac{a}{12} + 0.5 \rfloor \quad (10)$$

Overall Accuracy (OA): Overall Accuracy is the percentage of frames properly identified with both pitch and voicing based on the combination of voicing detection and pitch estimation. In terms of L, OA may be characterised as:

$$OA = \frac{1}{L} \sum_i V_i \tau[\zeta(f_i) - \zeta(F_i)] + \bar{V}_i \bar{v}_i \quad (11)$$

3 Result analysis

In this section we are comparing the result evaluation for Melodia and time domain adaptive filtering based model. In table 2 we can see the evaluation metrices performed on the Melodia for melody extraction and table 3 represents the result achieved by the time domain adaptive filtering based model. With the exception case of (VFA), which runs from 0 for best case to 1 for worst case scenarios, and all other measures range from worst (0) to best (1). The algorithm's efficiency is calculated by averaging the evaluation score of all music excerpts for the measure in consideration across the entire music dataset.

For analysis of these models lets check for its best possible outcome. Assuming that we have a flawless contour filtering strategy, we run tests to evaluate the best possible outcome our state-of-the-art algorithm could obtain. Taking a look at the findings that our system produced, we can make some observations. The total accuracy of the ideal contour filtering simulation, for starters, is less than

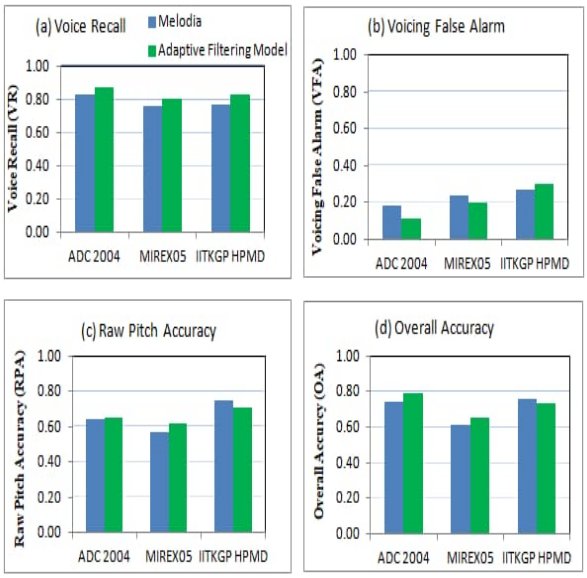

Figure 3: Performance comparison of Melodia and time domain adaptive filtering model over various test set. (a) Voicing Recall(VR) for Melodia and time domain adaptive filtering model. (b) Voicing False Alarm (VFA) for Melodia and time domain adaptive filtering model. (c) Raw Pitch Accuracy (RPA) for Melodia and time domain adaptive filtering model. (d) Overall Accuracy (OA) for Melodia and time domain adaptive filtering model.

one hundred percent, as shown in table. When comparing the datasets ADC2004 and Mirex05, we can see in Fig. 3, that the adaptive filtering based technique performs much better than Melodia in terms of RPA and OA. TWM is able to provide a resonance frequency that falls inside the ZFF's invariance range because of the predominance of the voices. On the IITKGP HPMD dataset, the time domain adaptive filtering technique achieves RP and OA results that are equivalent to those obtained with the Melodia method. It follows from this that the adaptive filtering based technique works better when dealing with music signals that have a high concentration of voices. Furthermore, owing to the impulsive nature of the percussion instrument's source, ZFF was unable to extract the proper GCI placements of the voices. In the datasets ADC2004, Mirex05, and IITKGP HPMD, an overall increases for adaptive model in VR is found, which may be ascribed to the broad dynamic range of the SoE contour used for threshold. SoE and misclassification of non-vocals into vocals have grown in IITKGP HPMD owing to the frequent stimulation of the Tabla, as well as the Drum, which causes an

increase in VFA performance.

4 Conclusion

For the purpose of automatically extracting the primary melody from a polyphonic piece of music, we investigated the performance of Melodia and a time domain adaptive filtering based model in this study. In Melodia, pitch contours were formed by combining the melodic pitch candidates that were obtained via various signal processing procedures. It is possible to identify melodic and non-melody contours by analysing these pitch contours and their distributions. In time domain adaptive filtering model, The ZFF's bandpass filtering properties are taken advantage of to create a hybrid time- and frequency-domain melody extraction approach. In polyphonic music, the SoE contour is thresholded to discern vocal and non-vocal parts. The note segment sequence is produced by sensing their frequency onsets. TWM method obtains the mean subtraction filter resonance frequency. Finally, the melody contour is retrieved by time-domain adaptive zero-frequency filtering each note segment. When using this approach, the lowered results are mostly due to the mean subtraction window length being identified often outside of the invariance range.

Acknowledgments

"This research was funded under grant number: ECR/2018/000204 by the Science & Engineering Research Board (SERB)."

References

Karin Dressler. 2011. An auditory streaming approach for melody extraction from polyphonic music. In *ISMIR*, pages 19–24.

Jean-Louis Durrieu, Gaël Richard, Bertrand David, and Cédric Févotte. 2010. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE transactions on audio, speech, and language processing*, 18(3):564–575.

Emilia Gómez, Francisco J Cañadas-Quesada, Justin Salamon, Jordi Bonada, Pedro Vera-Candeas, and Pablo Cabañas Molero. 2012. Predominant fundamental frequency estimation vs singing voice separation for the automatic transcription of accompanied flamenco singing. In *ISMIR*, pages 601–606.

Masataka Goto. 2004. A real-time music-scene-description system: Predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311–329.

Ludger Hofmann-Engl. 1999. Review of wb hewlett & e. selfridge-field, eds., melodic similarity: Concepts, procedures, and applications (cambridge, massachusetts: Mit press, 1999). *Music Theory Online*, 5(4).

Chao-Ling Hsu and Jyh-Shing Roger Jang. 2010. Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion. In *ISMIR*, pages 525–530.

Anssi Klapuri. 2004. *Signal processing methods for the automatic transcription of music*. Tampere University of Technology Finland.

Sangeun Kum, Changheun Oh, and Juhan Nam. 2016. Melody extraction on vocal segments using multi-column deep neural networks. In *ISMIR*, pages 819–825.

Ranjeet Kumar, Anupam Biswas, and Pinki Roy. 2019. Melody extraction from polyphonic music using deep neural network: A literature survey. *Journal of Software Engineering Tools & Technology Trends*, 6(3):16–21.

Ranjeet Kumar, Anupam Biswas, and Pinki Roy. 2020. Melody extraction from music: A comprehensive study. In *Applications of Machine Learning*, pages 141–155. Springer, Singapore.

Hyunsin Park and Chang D Yoo. 2017. Melody extraction and detection through lstm-rnn with harmonic sum loss. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2766–2770. IEEE.

Graham E Poliner, Daniel PW Ellis, Andreas F Ehmann, Emilia Gómez, Sebastian Streich, and Beesuan Ong. 2007. Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1247–1256.

Vishweshwara Rao and Preeti Rao. 2010. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE transactions on audio, speech, and language processing*, 18(8):2145–2154.

M Gurunath Reddy and K Sreenivasa Rao. 2018. Predominant melody extraction from vocal polyphonic music signal by time-domain adaptive filtering-based method. *Circuits, Systems, and Signal Processing*, 37(7):2911–2933.

Matti P Ryynänen and Anssi P Klapuri. 2008. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86.

Justin Salamon and Emilia Gómez. 2012. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770.

Justin Salamon, Emilia Gómez, Daniel PW Ellis, and Gaël Richard. 2014. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134.

Justin Salamon, Joan Serra, and Emilia Gómez. 2013. Tonal representations for music retrieval: from version identification to query-by-humming. *International Journal of Multimedia Information Retrieval*, 2(1):45–58.

Eric D Scheirer. 2000. Machine-listening systems. *Unpublished Ph. D. Thesis, Massachusetts Institute of Technology*.

Li Su. 2018. Vocal melody extraction using patch-based cnn. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 371–375. IEEE.

Hideyuki Tachibana, Takuma Ono, Nobutaka Ono, and Shigeki Sagayama. 2010. Melody line estimation in homophonic music audio signals based on temporal-variability of melodic source. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 425–428. IEEE.

Tzu-Chun Yeh, Ming-Ju Wu, Jyh-Shing Roger Jang, Wei-Lun Chang, and I-Bin Liao. 2012. A hybrid approach to singing pitch extraction based on trend estimation and hidden markov models. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 457–460. IEEE.

Dorabella Cipher as Musical Inspiration

Bradley Hauer, Colin Choi, Abram Hindle, Scott Smallwood, Grzegorz Kondrak
University of Alberta, Edmonton, Canada
{bmhauer,cechoi,hindle1,ssmallwo,gkondrak}@ualberta.ca

Abstract

The Dorabella cipher is an encrypted note written by English composer Edward Elgar, which has defied decipherment attempts for more than a century. While most proposed solutions are English texts, we investigate the hypothesis that Dorabella represents enciphered music. We weigh the evidence for and against the hypothesis, devise a simplified music notation, and attempt to reconstruct a melody from the cipher. Our tools are n-gram models of music which we validate on existing music corpora enciphered using monoalphabetic substitution. By applying our methods to Dorabella, we produce a decipherment with musical qualities, which is then transformed via artful composition into a listenable melody. Far from arguing that the end result represents the only true solution, we instead frame the process of decipherment as part of the composition process.

1 Introduction

The Dorabella cipher (henceforth, simply *Dorabella*) is an encrypted note sent by Edward Elgar, the composer of the "Enigma Variations", to his friend Dora Penny in 1897 (Santa and Santa, 2010). While many cryptography researchers have assumed that the underlying message is an English text, it has also been hypothesized that it may encode music, since Elgar was a composer and a music teacher. This raises several interesting questions. Is it possible to find evidence for or against the music hypothesis? What kind of music notation could be devised with only two dozen possible distinct symbols? How would a musical decipherment compare to the proposed textual decipherments?

In this paper, we attempt to answer these questions in a principled manner, by using n-gram language models derived from collections of transcribed music. However, we also approach musical

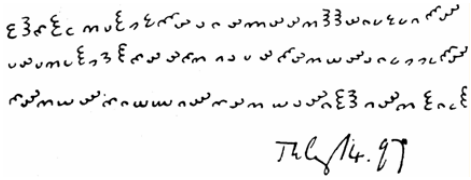

Figure 1: The Dorabella Cipher.

decipherment as a creative process. We demonstrate this technique on Dorabella, producing a decipherment that has musical qualities, transformed via artful composition into a listenable melody. While prior work typically pursues a single correct decipherment, we instead adopt a creative approach of converting ciphers into music, which might lead to composition of new works.

This paper has the following structure: In Section 2 we provide background on n-grams, perplexity and monoalphabetic substitution ciphers. In Section 3, we discuss prior work. In Section 4, we describe our methodology, including datasets for training language models. In Section 5, we describe our results on encrypted melody samples. In Section 6, our highest-scoring decipherment of Dorabella as a melody is used as inspiration to compose a new work.

2 Background

Substitution ciphers, their properties, and cryptanalysis techniques have been studied for centuries (Singh, 2011). A *monoalphabetic substitution cipher* enciphers a *plaintext* by applying a 1-to-1 mapping of symbols to each character token, producing a *ciphertext* which has a length equal to the length of the plaintext. The symbol mapping function is called the *key*. Given the key, reversing the encipherment process and recovering the plaintext is trivial: simply apply the inverse of the key to each ciphertext symbol. The process of recovering the plaintext when the key is *not* given

Proceedings of the Workshop on Speech and Music Processing (SMP-2021), pages 33–38
co-located with ICON-2021, December 16-19, 2021, Silchar, India.

```
A2 E3 B2 A3 A1 C2 G1 A3 D1 H2 B3 F2 F1 B1 F2 C3 F2 F2 C2 E3 E3 F2 B1 H1 H2 H1 C1 B3 F3
G1 F2 G1 C2 H1 A3 D1 D2 A3 B2 F2 F2 B2 C2 C1 F1 G1 F2 B3 F2 C2 G2 F3 F1 B1 H1 D1 D1 H1 B3 F3
B2 F3 C2 G2 F3 B2 B1 G2 G3 C1 F3 B2 F2 C2 G2 F1 F3 C1 A3 E3 C1 F3 C2 A3 B1 H1 A3
```

Figure 2: Our transcription of Dorabella which encodes the orientation and semicircle count of each symbol.

is called *decipherment*. Common measures of the decipherment success are: (1) *decipherment accuracy*, which is the percentage of correctly recovered symbols in the ciphertext; and (2) *key accuracy*, which is the percentage of correctly mapped symbols in the cipher alphabet. Decipherment accuracy is typically higher than key accuracy, because more frequent symbols are more likely to be deciphered correctly.

Computational decipherment methods are based on heuristic search algorithms guided by statistical n-gram language models (Nuhn et al., 2013; Hauer et al., 2014). An *n-gram language model* estimates the probability of a token in a sequence based on the previous n$-$1 tokens. If the token is near the beginning of the string, a special start token is used in place of the missing prior tokens. Through repeated applications of this model, the probability of the entire sequence can be estimated. *Perplexity* is a function of probability, which measures the ability of a statistical model to predict a particular sequence. Lower perplexity indicates that the model is "less surprised" by the data, and so is said to be a better fit. Tokens may be characters or words in natural language, or symbols used in music notation. For compatibility with prior work which models sequences of characters in natural language, we refer to a language model over music notation as a *character language model*.

3 Prior Work

This section describes prior works that use n-gram models for composition, and prior attempts to solve the Dorabella cipher.

3.1 N-Gram models for composition

N-gram models have been applied to study the structure of music (Manzara et al., 1992) and to compose music. N-gram models in music research and composition range from serial notes, to chords, to pitch and duration pairs (Lo and Lucas, 2006; Wołkowicz et al., 2008), and more complicated structures (McCormack, 1996).

Manzara et al. (1992) investigate the entropy of music from an n-gram perspective. They test how well people can guess the next note, and compare that to n-gram models of Bach's four part *Chorale*. They report that people outperform n-gram models, and that both people and n-gram models have relatively consistent performance.

McCormack (1996) employs n-grams and similar Bayesian structures to compose music. His focus was on Markov chains, which are related to n-gram language models and perplexity estimations.

Lo and Lucas (2006) combine genetic algorithms and n-gram language models to evolve musical sequences. The n-gram models act as fitness functions to guide the creation of musical sequences that have lower perplexity given an n-gram language model. In addition, they use their models to identify composers.

Wołkowicz et al. (2008) also use n-grams to identify composers. They process MIDI files and produce n-grams of pitch and duration tuples. They achieve up to 84% accuracy at identifying composers using a large corpus of 10000 notes of each composer's work, and about 54% accuracy when using only 100 notes, which is at a similar level of accuracy as Lo and Lucas (2006).

3.2 Dorabella Cipher

The earliest computational attempt at solving the Dorabella Cipher that we are aware of is that of Sams (1970). He applies statistical analysis based on character frequencies and brute force cryptanalysis. The work considers the assumptions that the cipher encrypts English text which may be partly phoneticized, is not strictly monoalphabetic, and may involve multiple layers of encryption. The author ultimately proposes the following solution to the cipher: *"Larks! It's chaotic, but a cloak obscures my new letters, a, b. I own the dark makes E. E. sigh when you are too long gone."*

Santa and Santa (2010) provide an overview of Elgar's work on cryptography, focusing on the "enigma" that he implied to be hidden within his musical piece *Variations on a Theme*. They note the connections Elgar made between that piece and Dorabella, neither of which has been conclusively solved. despite this and other "hints" from Elgar. In particular, they raise the possibility of mathematical concepts being used in Dorabella, specifically

the constant π, as well as the encoding of scale-degrees with numerical values.

As well-known techniques, such as frequency analysis, have not proven effective on Dorabella, Schmeh (2018) proposes to consider less common techniques. These include vowel detection and a frequency-based consonant identification method. The author applies these techniques both to Dorabella, and on a control plaintext. He does not propose a solution to Dorabella, but demonstrates that these methods can distinguish between vowels and consonants in the control cipher. With the same techniques, he attempts to identify some Dorabella symbols as vowels or consonants. He also notes that certain statistical properties of Dorabella are consistent with English text.

Packwood (2020) proposes a natural language solution to Dorabella. The method is complex, and involves breaking the cipher into discrete blocks, among which patterns can be observed, and an elaborate system of transposition. The author further speculates that the cipher also conceals a musical composition, but makes no attempt at a musical decipherment.

Hauer et al. (2021) experiment with several monoalphabetic substitution cipher solvers to decipher music. They rely on a corpus of Bach and Elgar MIDI files, and try to decipher synthetic music ciphers using a pitch/duration language model, but the results are quite low compared to textual ciphers. They conclude that it is unlikely that Dorabella represents music encoded using an alphabet of pitch and duration.

4 Methodology

In this section, we describe our methodology, including datasets for training language models.

4.1 Transcribing Dorabella

The first step is to render Dorabella into a machine-readable form. In order to establish such a transcription, we compared five different manual transcriptions attempts, including Schmeh (2018), Hartmeier (2017), Pelling (2012), as well as transcriptions by two of the authors of this paper. A majority consensus transcription is shown in Figure 2. It consists of 87 tokens made of an uppercase letter encoding the symbol orientation, followed by the number of semicircles. There are 8 possible orientations (A-H), while the number of semicircles ranges from 1 to 3.

4.2 Pitch-Duration Dataset and Encoding

We use the music dataset created by Hauer et al. (2021). The dataset was created from MIDI files, a form of digitally representing musical composition which encodes pitch, pitch amplitude, and duration over a timeline, usually including metric and tempo information. The files represent music from both Elgar and Bach. The Elgar data consists of 29 files containing a total of 1.2M notes, while the Bach data consists of 295 files containing 3.7M notes. We include the Bach data due to the relatively small size of the Elgar corpus; this increases the total size of our data by a factor of four. Each dataset is divided into training and testing splits. This is done to ensure that experimental results are generalizable to data not used to provide statistical information for the models used by the decipherment algorithms. The test set is further divided into 87-note sequences, the same length as Dorabella.

Hauer et al. (2021) assume that enciphered music must, before encipherment, be represented in some serial, symbolic notation. To this end, they transpose all music into the key of C major, and use only one octave. All symbols except notes (e.g. rests) are removed. All notes are normalized to one of three durations: quarter note, shorter than a quarter note, or longer than a quarter note. Further, all notes were normalized to one of the eight most frequent notes: A, B, C, D, E, F, F$\sharp$, and G. Thus, just as each Dorabella cipher symbol has one of three semicircle counts and one of eight orientations, giving a theoretical vocabulary of 24 symbols, the encoding assigns to each symbol one of three durations and one of eight notes, yielding 24 distinct symbols. While there is much more information encoded in musical notation, we are constrained by the 24-symbol alphabet of the cipher. For example, if we assumed that some cipher symbols represent rests, we would need to further reduce the already limited range of notes that the cipher can represent. While this encoding was designed to match the form of the Dorabella cipher, we present a more principled approach in Section 4.4.

4.3 English Dataset

To assess the ability of our statistical models to fit music, we induce models of both music and English, and compare the fitness of our modelling method on different types of data. We use the English language dataset of Hauer et al. (2021), which is a subset of the letters of Jane Austen. This

corpus was deemed appropriate since it consists of written epistolary correspondences, which is the hypothesized domain of Dorabella. The text was first processed to remove all non-alphabetic characters, including white space. 300 excerpts from the corpus were selected at random, each consisting of a sequence of 87 characters. We use this set of 300 texts for the perplexity measurement experiment described in Section 5.

4.4 Melody Dataset

We experiment with the CANTUS corpus of folk music (Lacoste, 2012). We conjecture that Elgar would be more likely to create something that reflected contemporary styles of folk music rather than the more complex, and often more chromatically adventurous music of his own. This leads us to consider musical databases that are limited to a single line of music, as well as to simplify the issues around key signature, meter, rhythm, and other factors. Our melody corpus reduces all examples to the common key of C. We also chose to not attempt to model rhythms, dynamics, articulations, and other components, looking mainly at pitch, and assuming 4/4 meter. There are other composition decisions which could have been made. Given the extremely small set of symbols, and the short length of the cipher, we necessarily had to make some simplifying assumptions, and we did so based on our intuitions regarding what setting would produce the most natural-sounding composition.

Our melody dataset is from the CANTUS Database of chants and melodies (Lacoste, 2012; Helsen and Lacoste, 2011), an online searchable database that encodes melodies as sequences of pitches without including their durations.[1] Table 1 shows the sources of melodies in the CANTUS dataset and their average length. The melodies in CANTUS are monophonic, and most include notes in the range of F3 to D6. Since there are only 17 distinct notes in our subset of CANTUS, we interpolate the range from A3 to E6 to yield 24 symbols used to decipher Dorabella: A3, B♭3, B3, F3, G3, A4, B♭4, B4, C4, D4, E4, F4, G4, A5, B♭5, B5, C5, D5, E5, F5, G5, C6, D6, E6. Because of the smaller alphabet and vocabulary of the melody dataset, we expect it to have lower perplexity, which should lead to better results than with the dataset described in Section 4.2.

[1] An example melody http://cantusindex.org/melody/msch001

Name	Dataset	Melodies	Length
Gloria	mbos	102	8.9
Kyrie	mmel	226	8.7
Agnus Dei	mscb	267	8.9
Alleluia	msch	409	34.9
Hymn	msta	344	49.4
Sanctus	mtha	228	9.0
	All	1576	

Table 1: Melody datasets extracted from CANTUS (Lacoste, 2012; Helsen and Lacoste, 2011)

Our training corpus is created by randomly sampling 467 melodies without replacement. Our datasets, code, and compositions are released at: https://zenodo.org/record/4764819

4.5 Decipherment

As our decipherment method for enciphered music, we use the solver of Norvig (2009), which we refer to as HILLCLIMBC.[2] We selected it for its effectiveness on deciphering monoalphabetic substitution ciphers, even when word boundaries are not preserved in the cipher. This is important, as our encoding of music has no analogy to word boundaries, and no such boundaries are indicated in Dorabella. The solver maximizes the probability of the decipherment as estimated by a trigram character language model. Starting from a random initial key, HILLCLIMBC applies a hill climbing algorithm as a heuristic search strategy. At each step, many successor keys are generated by applying permutations to the current key; whichever successor gives the greatest increase in probability (equivalently, the greatest decrease in perplexity) becomes the key in the next iteration. We run the algorithm for 4000 iterations, with 90 random restarts. The decipherment with the lowest perplexity across all iterations is returned.

5 Decipherment Results

Table 2 shows the decipherment results on a test set of 300 distinct melody samples, sampled without replacement from the corresponding training set. Clearly, the results on the melody dataset are much better than those on the pitch/duration datasets, which in turn are better for Bach than for Elgar. The mean key accuracy across all examples in the melody dataset is 50%, that is, half of the key symbols are correct. Approximately half of ciphers

[2] http://norvig.com/ngrams

Source	Key Acc	Dec Acc
pitch/duration (Elgar)	7.0%	12.0%
pitch/duration (Bach)	26.5%	32.0%
melody (CANTUS)	50.0%	54.5%

Table 2: HILLCLIMBC results on music ciphers of length 20,000.

were deciphered with 70% decipherment accuracy or higher, and nearly one third of ciphers were deciphered entirely correctly. This suggests that our approach is effective for melody decipherment.

One reason for the lower accuracy on the pitch/duration datasets may be their quality. The original MIDI files were created by multiple authors, leading to a low consistency in musical transcription. In addition, the files are polyphonic; even for piano music, they often have separate channels for each hand. On the other hand, our melody dataset is monophonic and consistently transcribed.

Dataset	Average Perplexity
English (Austen)	16.2
pitch/duration (Elgar)	24.4
pitch/duration (Bach)	24.5
melody (CANTUS)	5.6

Table 3: Average perplexity using a trigram character language model.

Another possible explanation for the divergent performance could be the encoding. Table 3 shows the perplexity of different datasets. We used trigram language character models with modified Kneser-Ney smoothing and discounts. The relatively high perplexity values of the pitch/duration datasets suggests that the pitch-only encoding may be better suited to modelling music than the pitch/duration encoding. Indeed, based on these results, predicting the next note of a melody is easier than than predicting the next English character in a sentence.

6 Composition from Decipherment

In this section we apply our algorithm to the Dorabella cipher, and take the resulting melody as a basis for a composition. In particular, we manually analyze the output for musical content, and modify it according to subjective musical tastes. This creative process is guided by the familiarity with the composer's style, and is not itself replicable.

E6 B5 A4 B4 B3 F4 D6 B4 E5 F5 G5 G4 C4 C5 G4 A5 G4 G4 F4 B5 B5 G4 C5 D5 F5 D5 B♭4 G5 D4 D6 G4 D6 F4 D5 B4 E5 C6 B4 A4 G4 G4 A4 F4 B♭4 C4 D6 G4 G5 G4 F4 E4 D4 C4 C5 D5 E5 E5 D5 G5 D4 A4 D4 F4 E4 D4 A4 C5 E4 A3 B♭4 D4 A4 G4 F4 E4 C4 D4 B♭4 B4 B5 B♭4 D4 F4 B4 C5 D5 B4

Figure 3: A decipherment of Dorabella as melody.

Figure 3 shows our highest-scoring decipherment of Dorabella assuming 4/4 time. Figure 4 depicts its musical transcription, which was obtained by applying HILLCLIMBC with a language model derived from the melody dataset (467 samples). This decipherment attempt has some interesting musical features. The notes in Figure 4 seem at times to imply logical harmonic progressions. In the second half, there are even moments of motivic repetition, albeit not exact, which evoke a musical composition.

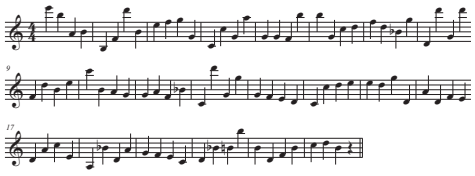

Figure 4: Note output from Dorabella

After manual analysis we decided to realign the notes from the 4/4 meter to the 3/4 meter, as this appears to fit better the contours and implied harmonic progression. Surprisingly, the melody seems to be match two 16-bar 3/4 phrases, except for a premature end in the second phrase. Considering that we only use quarter-note rhythms, this could be an illusion, but the resulting musical piece is intriguing. In the spirit of the creative process, we also decided to relax the strict matching of the decipherment symbols into notes.

Figure 5 shows the final version of the output in which some notes (shown in red) have been altered or added in order to create a cadential con-

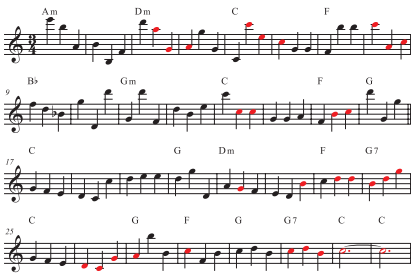

Figure 5: Adjusted output, with chords. Notes which have been modified from the output in Figure 4 are color-coded red.

clusion. Interestingly, our altered 32-bar segment features a highly disjunct first 16-bar phrase, with an altered pitch (B-flat) implying a potential transposition, and a second phrase that is much more lyrical and based on smaller step-wise intervals, complete with the "repeated motive." Upon adding the implied harmonic accompaniment, we can see that in some cases there even seem to be an implied V-I cadences, such as between mm. 16-17, mm. 24-25, and the final added measures. Adding some phrasing and interesting timbres, as well as chords based on the implied harmony, gives us the audio rendering[3] shown in Figure 5.

It is important to point out several caveats to this seemingly encouraging result. First and foremost, any analysis of musical composition necessarily has subjective elements. Second, we assume that rhythmic values are not encoded in the cipher, and limit the decipherment to quarter-notes. It is also possible that the score may not be connected to common practice notation or even diatonic pitches at all. For example, these could be referring to a very specific set of church bells, or perhaps some other kind of instrument or sonic contraption, or even just rhythm.

7 Conclusions

Although we do not claim to have solved the mystery of Dorabella, our process produced a listenable melody, which opens up interesting avenues of investigation. In the future, we plan to experiment with different corpora and musical attributes, such as rhythm only. Our approach represents a creative way to generate new forms of musical melodies. What seems certain is that Elgar's intention to confound left us with a tantalizing riddle that invites further speculation in the future.

References

Daniel Hartmeier. 2017. Clues (or red herrings) to the Dorabella cipher. *benzedrine.ch*. https://www.benzedrine.ch/dorabella.html.

Bradley Hauer, Colin Choi, Anirudh Sundar, Abram Hindle, Scott Smallwood, and Grzegorz Kondrak. 2021. Experimental analysis of the Dorabella cipher with statistical language models. In *Proceedings of the International Conference on Historical Cryptology (HistoCrypt)*.

Bradley Hauer, Ryan Hayward, and Grzegorz Kondrak. 2014. Solving substitution ciphers with combined language models. In *Proceedings of COLING, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2314–2325, Dublin, Ireland.

Kate Helsen and Debra Lacoste. 2011. A report on the encoding of melodic incipits in the cantus database with the music font'volpiano'. *Plainsong & Medieval Music*, 20(1):51.

Debra Lacoste. 2012. The cantus database: Mining for medieval chant traditions. *Digital Medievalist*, 7.

ManYat Lo and Simon M Lucas. 2006. Evolving musical sequences with n-gram based trainable fitness functions. In *2006 ieee international conference on evolutionary computation*, pages 601–608. IEEE.

Leonard C Manzara, Ian H Witten, and Mark James. 1992. On the entropy of music: An experiment with Bach chorale melodies. *Leonardo Music Journal*, 2(1):81–88.

Jon McCormack. 1996. Grammar based music composition. In *In R. Stocker et al. (Eds.), From Local Interactions to Global Phenomena, Complex Systems 96*. ISO Press.

Peter Norvig. 2009. Natural language corpus data. In Toby Segaran and Jeff Hammerbacher, editors, *Beautiful Data: The Stories Behind Elegant Data Solution*, pages 219–242. O'Reilly Media.

Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. Beam search for solving substitution ciphers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1568–1576.

Wayne Packwood. 2020. Elgar as cryptographer – tuning and turing. *Musical Opinion Quarterly*, pages 34–37.

Nick Pelling. 2012. Dorabella cipher. *Cipher Mysteries*.

Eric Sams. 1970. Elgar's cipher letter to Dorabella. *The Musical Times*, 111(1524):151–154.

Charles Richard Santa and Matthew Santa. 2010. Solving Elgar's Enigma. *Current Musicology*.

Klaus Schmeh. 2018. Examining the Dorabella Cipher with three lesser-known cryptanalysis methods. In *Proceedings of the International Conference on Historical Cryptology (HistoCrypt)*, pages 145–152.

Simon Singh. 2011. *The code book: the science of secrecy from ancient Egypt to quantum cryptography*. Anchor.

Jacek Wołkowicz, Zbigniew Kulka, and Vlado Kešelj. 2008. N-gram-based approach to composer recognition. *Archives of Acoustics*, 33(1):43–55.

[3] https://zenodo.org/record/4764819/files/fig3.wav?download=1

Author Index

Banerjee, Esha, 14
Biswas, Anupam, 20, 24

Choi, Colin, 33

Das, Dipankar, 1

Hauer, Bradley, 33
Hindle, Abram, 33

Jha, Girish, 14

Kondrak, Grzegorz, 33
Kumar, Ranjeet, 24

Ojha, Atul Kr., 14

Roy, Pinki, 24

Singh, Yeshwant, 20, 24
Smallwood, Scott, 33