**CARNEGIE MELLON UNIVERSITY**

# De-Entanglement: A Framework towards building Ubiquitous speech technologies

by

Sai Krishna Rallabandi

CMU-LTI-22-001

Thesis submitted in partial fulfillment
for the degree of Doctor of Philosophy

**Thesis committee:**
Alan W Black (Chair)
Eric Nyberg
Louie Phillippe Morency
Kalika Bali, Microsoft Research India

April 2022

# Abstract

Speech driven devices and interfaces like Apple Home pod, Google Home, Amazon Echo are increasingly becoming ubiquitous and have tremendous potential to affect our daily lives. However, deep learning models underlying these applications have yet unaddressed challenges like scalability, explainability and concerns like privacy and security.

In my dissertation I propose a framework called **De-Entanglement** that has linguistic concepts as first class objects. De-Entanglement attempts to build speech technology using two core concepts referred to as content and style. Specifically, content encompasses *acoustic phonetic information* while style encompasses *paralinguistic information* from the raw audio. In my dissertation I provide experiments that show how De-Entanglement can address three challenges in a holistic fashion:

- (1) **Scalability**: How to build speech technologies for new languages / language phenomena such as code switching? I demonstrate how De-Entanglement helps build more natural Text to Speech (TTS) voices. This part of the work has been deployed in the form of Android application in 13 Indian languages and has been assisting people since 2016.
- (2) **Flexibility**: How to build models that can be manipulated to accomplish a variety of functionality such as finetuning, meta learning, augmentation and self training? I present experiments in two types of models. In the context of generative models, I present an approach to show that De-Entanglement allows explicit global and local control of synthetic voices. In the context of discriminative models, I present approaches that leverage style information to detect para-linguistic events from a speech utterance.
- (3) **Explainability**: How to build technology that is reasonable to the stakeholders? I posit that explainable speech technologies should be characterized by two properties: (a) Reasonable Understanding of internal mechanisms in the model and (b) Demonstrable Utility of the model for downstream applications. Using language identification and intent recognition from acoustics as the target applications, I demonstrate (a) how suitable priors can be incorporated into a model and (b) how such an approach leads to strong performance in low and under resourced scenarios.

Since these linguistic constructs(concepts) are shared across different tasks within and beyond speech processing, the solutions designed using De-Entanglement hold promise to be applicable across different tasks. I present experiments to this end in both speech processing as well as broader Natural Language Processing.

# Contents

# List of Figures

# List of Tables

# Part I

# Overview

# 1

# Introduction

Language is a hallmark characteristic of human species[1]. Understandably, most depictions of a sufficiently advanced civilization involve language driven seamless interactions. The progress being made in language technologies today is making applications once assumed only in the realm of science fiction realizable. Systems today can identify objects in images and video, recognize and convey information via speech and translate across multiple languages. These systems are providing immense benefit to people across industry, government as well as society. Within language technologies, speech is regarded as the natural form of human communication. I believe that speech driven interactions are a cornerstone object in bringing technology closer to humans. I have therefore structured my dissertation around speech processing. I am interested in building ubiquitous speech technologies - technologies that can be deployed anywhere on the planet and used by anyone.

Over the years, speech technologies have been making a transition from working for 'some people in some scenarios' to 'most people in most scenarios'[2]. For instance, speech driven devices and interfaces like Apple Home pod, Google Home, Amazon Echo are increasingly becoming commonplace (Reid, 2018). These interfaces already have a tremendous impact on our daily lives: A few years ago one would have manually searched the internet for answer to a question like "*What is the capital of Peru?*", but today it is near natural to use a voice assistant. These interfaces support a variety of consumer applications like voice based device control, placing a call or an order, making a dinner reservation, etc.

[1] "Language is a soft historical artifact … a way of abstracting the experience", *Lera Boroditsky*, World Science Festival 2019

[2] This phrasing is borrowed from one of the talks my advisor gave to prospective students at Language Technologies Institute

The models underlying these real life applications are typically powered by machine learning and are characterized by yet unaddressed technical challenges like scalability, flexibility and explainability. Scalability in the context of this dissertation refers to the challenge of building speech technologies for all the languages on the planet, not just a handful. In addition to being a technical challenge, scalability also has implications that echo alongside free will and equal opportunity(Bali et al.). Flexibility refers to the challenge of building systems that can be adapted to accomplish different functionality. The ability to adapt to a scenario is a defining trait of human species and it is a reasonable expectation for technologies interacting with humans to depict this. Explainability refers to the challenge of building systems which can be interpreted by the stakeholders.

This dissertation is inspired to answer the question *"How should we build ubiquitous speech technologies?"*. I propose a concept based framework called **De-Entanglement** that has linguistic concepts as first class objects that are shared across different tasks. The linguistic concepts I employ are *'content'* and *'style'*. Content refers to the acoustic phonetic information present in the signal while style refers to the para-linguistic information. I envision any real life speech processing application such as Automatic Speech Recognition(ASR) or Text to Speech(TTS) as a process that involves meticulous manipulation of these linguistic concepts. Since all the processes share content and style, I posit that using these concepts as building blocks provides way to design holistic solutions to the technical challenges.

## 1.1 Technical Challenges

In (Rosenberg, 2018), nine technical challenges have been proposed in the context of speech prosody. In my dissertation I propose a more generic set of challenges that encompass all the nine challenges as well as extend them to include ubiquity of the deployed technology. The differences between the sets of challenges can be seen in table 1.1. The challenges I propose are:

- Scalability
- Flexibility
- Explainability
- Privacy and Security Concerns
- Ethical Considerations
- Deployment Considerations

I posit that a concept based framework like De-Entanglement offers elegant solutions to these challenges. Within the scope of this dissertation, I am addressing the first three challenges:

Scalability, Flexibility and Explainability. Application of frameworks such as De-Entanglement to address the other challenges is beyond the scope of the current dissertation and I leave it as future work.

**Scalability**

A major bottleneck in the progress of many data-intensive language processing tasks such as speech recognition and synthesis is scalability to new languages and domains [3]. Building such technologies for unwritten or under-resourced languages is often not feasible due to lack of annotated data or other expensive resources. I posit that there are two distinct categorizations that pose challenges in terms of scalability: (1) Unwritten languages and low resource scenarios and (2) Code switching and other non native speech phenomena.

*Scalability for Unwritten Languages and Low Resource Scenarios*

Let us consider building speech technology for unwritten or under-resourced languages. A fundamental resource required to build speech technology stack in such languages is phonetic lexicon: something that translates acoustic input to textual representation. Having such a lexicon - even if noisy and incomplete - can help bootstrap speech recognition and synthesis models which in turn enable other applications such as key word spotting. I hypothesize that a concept based framework can be employed to obtain a phonetic lexicon.

*Scalability for Code switching and other non native Speech phenomena*

Code switching(or mixing) is a pseudo native phenomenon where speakers alternate between the languages while speaking. It occurs in multilingual societies such as India, Singapore, etc. and is used both to express opinions as well as for personal and group communications. The technology today - from speech processing systems through conversational agents - assume monolingual mode of operation and do not process code-switched content. However, the mixed content is intuitively the most important part in the content. Since the systems are now handling conversations, it becomes important that they handle code-switching.

In this dissertation I present the observation that speech has both continuous as well as discrete priors: The generative process of speech has been shown to be modeled by a Gaussian prior distribution which is continuous in nature. However, the language which is also present in the utterance can be approximated to be sampled from a discrete prior distribution. Exact manifestation of this sampling can be at different levels: phonemes, words, syllables, sub word units, etc. Based on this insight, we show(Rallabandi and Black, 2019) that incorporating priors help encode language independent information thereby facilitating synthesis of code mixed content. In addition to basing priors on knowledge about observations, I hypothesize that it is also possible to base them on discovered patterns. In (Rallabandi et al., 2018b), we have discovered several code switching styles based on (Guzmán et al., 2017) and show that modeling code mixed language using these priors improves perplexity.

[3] "By excluding these languages from reaping the benefits of the advancements in language technology, we marginalize the already vulnerable groups even further.", (Joshi et al., 2019)

### Flexibility

Consider a generative model such as speech synthesis from given text: Long form text is characterized by rich natural variations in terms of content, persona, speaking style, etc. Typical statistical approaches to speech synthesis have known to normalize out the rich variations resulting in bland speech. This effect, often referred to as averaging effect, has also been shown to contribute towards listening effort. I posit that a concept based framework like De-Entanglement which explicitly handles style information as a core concept can address this issue.

I divide the challenge of flexibility into two sub challenges: global and local flexibility. Global flexibility refers to the ability of a model to adapt to a consistent speaking style throughout the utterance. On the other hand, local flexibility refers to the ability of a model to consistently generate appropriate prosody based on the context. For instance, consider synthesizing children's stories in the form of audio books. Since stories contain multiple characters, the model should be able to (a) generate content in different styles based on the character, demonstrating global prosody and (b) simultaneously generate word level or phrase level local prosody depending upon the context. In this dissertation, I present experiments that demonstrate how a concept based framework can be employed to identify the global style of a speech utterance (Rallabandi et al., 2018a; Wu et al., 2019). I then present an approach to automatically discover and incorporate word level emphasis in TTS systems based on quantized Fundamental frequency(F0).

### Explainability[4]

Real life applications involving language are inherently composite in nature. In other words, they often require modeling the abstract relationships so as to capture the unseen compositions of seen concepts at test time. Attempts at accomplishing compositionality without a handle on explainability is a deceptively non trivial task and might lead to models learning just surface level associations(Agrawal et al., 2017b; Chen et al., 2017a; Goyal et al., 2017). I present a case that concept based frameworks like De-Entanglement provide additional information that can be exploited to make progress towards addressing the challenge of explainability. Specifically, I posit that models trained with concept based frameworks can act as justifying modules for the primary task. To validate this I present an example application where the model is aimed at identifying the language of a speech utterance as the primary task while hypothesizing discrete latent linguistic units as the auxiliary task. Using the observation that the latent units are different for monolingual and code mixed utterances, I show how De-Entanglement can present ideas towards building justifications.

In my dissertation I propose to build speech technologies employing information[5] as the frame of reference. Specifically, I propose content information and style information as the two core

[4] "Interpretability is not a monolithic concept, but in fact reflects several distinct ideas", (Lipton, 2018)

[5] "...if we were to name the most powerful assumption of all, which leads one on and on in an attempt to understand life, it is that all things are made of atoms, and that everything that living things do can be understood in terms of the jigglings and wigglings of atoms",(Feynman et al., 2011).

| Challenge from (Rosenberg, 2018) | Challenge from this Dissertation |
|---|---|
| Make More Available Labeled Data | Scalability |
| Bridge the Gap Between Data and Technology | Scalability |
| Develop, expand theory to describe conversational phenomena | Scalability, Flexibility |
| Develop, expand theory to describe dimensions of Variation | Flexibility |
| Increase Understanding of the Prosody of Emerging Bilinguals | Scalability, Flexibility |
| Lower the Barriers to Including Prosody in Applications | Flexibility |
| Improve the Objective Evaluation of Prosodic Assignment | Flexibility |
| Understand Prosody in End-to-End Speech Synthesis | Scalability, Flexibility |
| Demonstrate the Value of Prosody to Speech Recognition | Scalability, Flexibility, Explainability |

TABLE 1.1: Categorization of challenges from (Rosenberg, 2018)

concepts that abstract out and move the frame of reference from individual speech processing tasks to linguistic concepts.

## 1.2 Motivations for the proposed framework

In this section I provide the theoretical motivations that inspired the proposed framework. These motivations have been derived from Physics. I acknowledge that my training is not directly in sciences, rather than in engineering. Therefore, my understandings and interpretations may not be completely accurate. However, I use my current comprehension of some of the concepts proposed in sciences and apply them while deriving my framework while taking creative liberties as and when I see fit.

### 1.2.1 Physics

This set of motivations are derived by the observation that current process based approach used for building speech technologies does not explicitly capture the notion of information. Typical approaches to training a task involve maximizing the likelihood(entropy). At the outset, it appears that we are considering the notion of information since entropy is related to information. In fact, there is nothing wrong with this approach per se. After all, we have systems like ASR which have already achieved *human level parity* using this very approach. However, for generative models such as TTS, the formulation of maximum likelihood is typically via L1 loss. It has been well documented that generative models spend a lot of time learning useless correlations ignoring the conditional information (Oord et al., 2018). This can be observed in practice as well. It has been a common consensus that while the model reaches its optimal scalar loss value within 5 percent of training, we need to keep training the model 20-25 times longer for the convergence to translate into good speech quality. In addition, there is no way looking at the scalar loss value one can determine if the model has indeed converged or not. This is a troublesome thing from the perspective of an engineer. I propose to explicitly consider

information and argue that it provides a more clear framework for training models. To reach this conclusion, I draw inspiration from physicists.

**Information** Physicists have always considered information and entropy while designing their frameworks. Consider this line from the public lectures on Astrophysics by Arthur Eddington (Eddington, 2012):

*"The law that Entropy always increases(second law of thermodynamics) holds the supreme position among the laws of nature. If someone points out to you that your pet theory of the universe is in disagreement with Maxwell's equations, so much the worse for Maxwell's equations. If it is found to be contradicted by observations, well, these experimentalists do bungle things some times. But if your theory is found to be against the second law of thermodynamics, I can give you no hope. There is nothing for it but to collapse in deepest humiliation."*

To show how profound this impact is, let me highlight an interesting work by Beckenstein (Bekenstein, 1973). Beckenstein observed that second law is violated when principles of quantum mechanics are applied at the horizon of Black Holes. To ensure that the law is not violated, he proposed that Black Holes themselves must have entropy and it should be proportional to its area, leading to a lot of further work, for example in Holography(Emparan et al., 2002).

**Content and Style** While it is tempting to simply apply any framework of information to our present day models, it is a non trivial task sometimes leading to questionable interpretations. For instance, consider the series of works on Information Bottleneck theory by Tishby(Tishby et al., 2000). This theory explicitly applies the principles of entropy and information to training deep learning models. The central idea in IB theory is that DNNs are first order Markov chains and that the Mutual Information(MI) between the inputs(X) and outputs(Y) cannot increase as we proceed along the layers. By explicitly minimizing MI, IB theory promises to address challenges like Generalizability and Interpretibility(Tishby and Zaslavsky, 2015). This theory has caught a lot of attention. Tishby also point out interesting observations using this theory that characterize the dynamics of DNN training into two phases: fast compression phase and slow fitting phase. This in part might also explain the observation from previous subsection that current generative models take long time to converge even after reaching near optimal scalar loss value. However, there have been several criticisms of IB theory. Perhaps the strongest one comes from (Saxe et al., 2019) where they point out that the resolution of Mutual Information (between X,Y) is infinite and hence the magnitude does not necessarily indicate minimization of entropy.

I acknowledge that this is an ongoing debate and seemingly far from completion. Instead of adding to the debate, I take a different approach by inheriting the core ideas but taking creative liberties in how I exploit them. Specifically, I hypothesize that a better approach is to use the principles of entropy/information but ground them in linguistic concepts, thereby leading to formulations that do not have infinite resolution. Linguistic concepts are discrete in nature. Consider this line from the book Acoustic Phonetics by Kenneth Stevens(Stevens, 2000) (chapter 4):

*" Examination of time varying sources and the filtering leads to the observation that some aspects of the transformation from articulation to sound are categorical. These classes are closely related to the discrete linguistic categories or features that describe how words appear to be stored in the memory of a listener. "*

Language is a wonderful abstraction of our experiences and since we can exploit the discrete nature of linguistic concepts(it is A or B), I hypothesize that formulation information using linguistic constructs avoids the infinite resolution problem. I provide experimental evidence(part2 of thesis) that explicitly modeling content and style information not only leads to building better TTS voices, but also makes training faster.

Finally, I would like to point out two seemingly different papers from 1935, both by Albert Einstein: one on General Relativity(Einstein and Rosen, 1935) and the other on Quantum Mechanics(Einstein et al., 1935). It has been shown that these disparate concepts are indeed related, leading to conclusions that can be summarized(Relation and Duality) using the statement,

*" The fabric of space time is stitched together by Entanglement "*

Inspired by this observation, I hypothesize the following analogue with respect to speech processing extending the above arguments about content and style:

*" The fabric of a speech utterance is stitched together by entanglement of content information and style information"*

## 1.3 What is the proposed framework

In my dissertation I propose a concept based framework called De-Entanglement that has linguistic concepts as first class objects. De-Entanglement attempts to build speech technology using two core concepts referred to as content and style.

Content refers to acoustic phonetic information in the speech signal. Most of the existing works characterizing content information attempt to perform analysis of speech. There is a rich body of literature examining content information in speech at the subsegmental(Murty and Yegnanarayana, 2008) and segmental levels(Stevens, 2000). (Murty and Yegnanarayana, 2008) present an extensive analysis of how sub segmental level representations such as epoch encode information helpful in estimation of various speech parameters. (Stevens, 2000) investigate various properties of segment level representations such as phonemes and show they encode information that helps humans distinguish different sounds even in presence of external noise. Some authors extend the representations to include both phonemes and allophones. Since I am interested in building ubiquitous speech technologies[6] - technologies that can be

[6]Content information in the case of applications such as Alexa and Google Home exists at the level of words in the context of intent recognition and phonemes in the context of phonetic decoding.

deployed anywhere on the planet and used by anyone - I extend the characterization of content to simultaneously include information beyond the level of phonemes: sub words, words and phrases in human speech.

While content seems to be defined at multiple levels sometimes leading to confusion, the definitions of style are even more tricky. Paralinguistic style of speech is often defined ex negativo[7]: it comprises everything not considered content. In the literature, various forms of style are investigated such as emotion, personality and affect. (Pike, 1945) define intonation as the style of an utterance and characterize utterances based on it. Some authors identify speaker and language as the defining feature corresponding to style. Some identified that style has additional functionality revealing information such as age and gender. In my dissertation I build on this definition of style and extend it to include the intent of the speaker. An example of intent is "request", manifested in the sentence "Alexa, play songs". In the world of De-Entanglement, style information simultaneously includes the information about speaker, language, intent and other paralinguistic information in the utterance.

Few researchers have considered the notion of combining content and style leading to an intersection of both in an utterance as opposed to a disjoint definition. Note that while there have been works combining one form of content information(such as phonemes) and one form of style information(such as speaker) in the past, works that account for the fact that multiple forms of content and style information simultaneously co-exist are very limited. Within De-Entanglement, all forms of content - such as characters, phonemes, sub words, syllables, words, sentences simulatneously co-exist. I believe this provides a richer and more interesting way of modeling speech in the context of a real life application. For example, in the sentence "I need to withdraw 500 dollars", the words '500 dollars' corresponds to the content information when intent recognition is being performed while the phonemic representation is the content when the sentence is being decoded by an ASR system.

## 1.4 Limitations of the framework

While De-Entanglement attempts to build speech technologies, there are certain limitations that are not addressed by this framework. Here are some of the limitations I have currently identified:

- **De-Entanglement does not yet consider multimodal information** Real life application typically involve more than one modality. De-Entanglement natively supports only speech based applications and ignores data from other modalities.
- **Perhaps missing an additional construct 'structure'** - Since De-Entanglement only processes speech data, it is currently defined using two constructs: content and style.

[7] term borrowed from (Schuller and Batliner, 1988) chapter 1

However, it might be necessary to expand the concepts to include additional construcs such as structure to effectively handle multiple modalities.

- **Does not address the challenge of 'real time deployment'** De-Entanglement presupposes that the real time deployment challenge has already been addressed. While this is an important challenge, I consider this challenge beyond the scope of my dissertation.

- **Does not specify evaluation** To comprehensively address the challenges, it is important to consider the evaluation criterion. This is especially true in the context of generative models. Imagine an expressive speech synthesis system. It is currently not feasible to objectively evaluate the effectiveness of such a system. I consider this beyond the scope of my current dissertation.

## 1.5 Organization of this Dissertation

This dissertation is organized into 4 parts.

- **Part 1**: I introduce the framework and motivations behind it. I then provide a detailed explanation of De-Entanglement. In chapter 3, I describe 'FALCON', a toolkit I am developing as part of my PhD and used to perform experiments in this dissertation.

- **Part 2**: I address the challenge of 'Scalability'. To demonstrate the effectiveness of De-Entanglement, I present experiments first on De-Entanglement of content using Blind source separation as task(chapter 5). I then present experiments from De-Entanglement of style using code switching(chapter 4).

  I highlight two scenarios that the challenge of scalability poses: conversational speech and multilingual conditions. In my dissertation I am specifically limiting my self to code switching which is related to both these scenarios. I present experiments that show joint De-Entanglement of content and style helps build systems capable of effectively handling code switched inputs(chapters 6 and 7).

- **Part 3**: I address the challenge of Flexibility. I first present De-Entanglement of style by using detection of paralinguistic information from speech. Specifically, I present experiments that use utterance level representations in chapter 9 and approach that learns representation based on divergence in chapter 10. In chapter 11 I will present De-Entanglement of content using priors with Acoustic Unit Discovery as the target application. I employ the extracted units to accomplish Voice Conversion to prove that the learnt units address Flexibility.

- **Part 4**: In this part I address the challenge of Explainability. I posit that explainable speech technologies should be characterized by two properties: (a) Reasonable Understanding of internal mechanisms in the model and (b) Demonstrable Utility of the model

for downstream applications. Using acoustic unit discovery (chapter 15), I present experiments to show that we can inject reasonable priors into the model architecture. Using acoustic intent recognition, I show that such a model that we have reasonable understanding can be reliably employed for a downstream application under a variety of scenarios.

## 1.6 Technical Contributions from the Dissertation

- I present a concept first framework called De-Entanglement to build speech processing applications. Specifically, I posit that a speech utterance can be viewed as an entangelement of two types of information - content and style. Each of the information types have multiple forms of realizations: for example content information is simultaneously realized at character, phoneme, sub word, syllable, word, phrase and sentence levels.
- I propose several challenges that models underlying real life applications need to address - Scalability, Flexibility, Explainability, Privacy, Security and ethical considerations beyond deployment of technology.
- I present approaches employing De-Entanglement that address the challenges of Scalability, Flexibility and Explainability.
- I present approaches to build Text to Speech systems that can process code switched text and Speech Recognition systems that can decode code switching.
- I demonstrate that De-Entanglement can be employed to build TTS voices that exhibit explicit local as well as global control of prosodic characteristics.
- I show that De-Entanglement can be employed to build models that can isolate reasonable priors from the acoustic models. I also present experiments to show that such models can then be employed to build speech technologies in the form of intent recognition in low resource scenarios.

## 1.7 Other Contributions from the Dissertation

- **Applicability to other modalities**: While the work in this dissertation is primarily focused towards speech processing, the concept of De-Entanglement can be applied to other modalities as well. I present experiments from image captioning and Visual Question Answering to show that these ideas can be extended to multimodal scenarios as well.
- **Real world Impact**: The work done addressing 'Scalability' challenge has been employed to support TTS system in Indian languages. In collaboration with a startup 'Hear2Read', this work is now available in the form of Android applications from Google

Play Store in 10 Indian languages - Assamese, Gujarati, Hindi, Kannada, Marathi, Malayalam, Punjabi, Sanskrit, Tamil and Telugu along with Indian accented English.

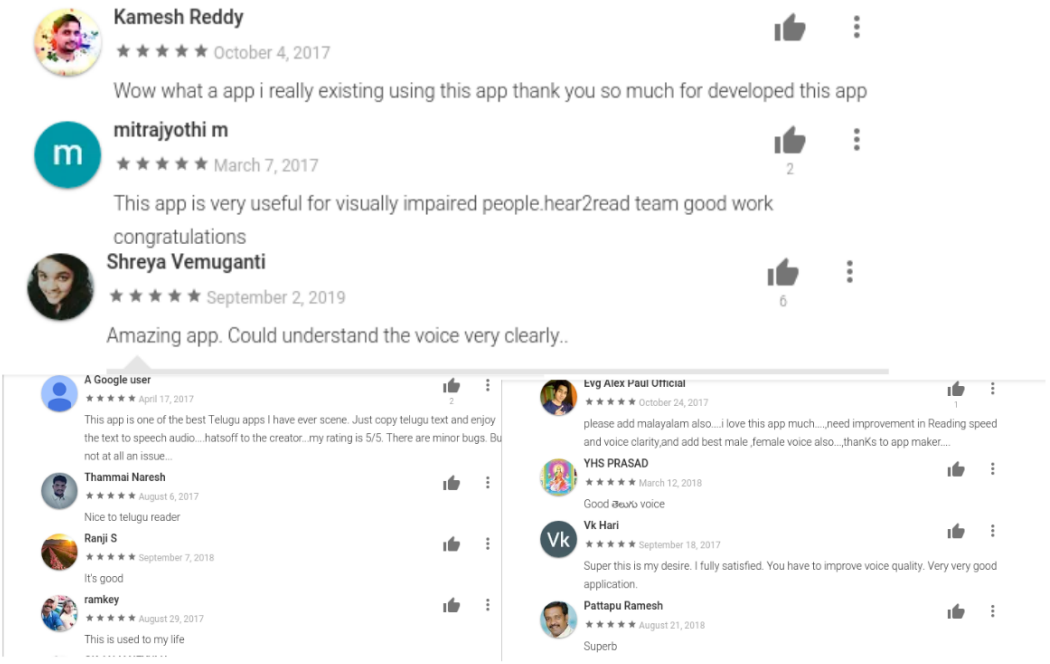



FIGURE 1.1: Reviews from TTS voice built and deployed using De-Entanglement. Note: All these reviews are from 'Telugu' app which was built using my own voice.

- **Research Toolkit**: To facilitate research and its applications within De-Entanglement, I have built the toolkit FALCON which extends FestVox(Anumanchipalli et al., 2011), the voice building suite of LTI.[8].
- **Assistant**: The dissertation work also includes development of a digital personal assistant - Judith which ships with FALCON and has capabilities for literature recommendation and maintaining GPU clusters. De-Entanglement is employed to add additional multimodal capabilities such as scene description[9] and empathy.

[8]The toolkit is opensource and is available at https://github.com/festvox/festvox/tree/master/src/falcon
[9]More information about assistants can be found here - http://www.cs.cmu.edu/ srallaba/ProjectAssistCore/

# 2

# De-Entanglement

*Idea of the framework is to provide a foundation on which either a full statistical model or a rule based model can be built.*

- Robert A.J. Clark, *PhD Dissertation*

In this chapter I present **De-Entanglement**, a theoretical framework aimed at building ubiquitous speech technologies. I first present the relevance of information in current deep learning systems, followed by an argument for De-Entanglement with information as the core building block. I then contrast De-Entanglement with a closely related term Disentanglement. Finally, I posit four approaches to accomplish De-Entanglement and review related works in this direction.

## 2.1 Information Theory and Relevance to Deep Learning

Given the rise of deep learning, there has been a lot of interest in trying to understand the reason behind success of these models. This has paved way to interpretations and explanations of the performance from seemingly disparate perspectives: theoretical physics and chemistry, statistical mechanics, differential geometry, communication theory, etc. In this writeup, I will be employing the lens of communication, specifically information theoretic concept referred to as Information Bottleneck Theory(Tishby et al., 2000), to guide the intuitions and provide inspiration for De-Entanglement. I encourage the interested readers to refer to (Tishby et al.,

2000; Tishby and Zaslavsky, 2015) for a comprehensive understanding of information bottleneck theory. Instead, I will use analogies to explain the reasoning behind the core idea. I acknowledge that there are several works that critique the role of information bottleneck theory. Instead of contributing to the ongoing debate, I would like to take inspiration from the hypothesis and observations from both the sides and apply them as necessary.

### 2.1.1 Mutual Information

One of the fundamental tools Information Bottleneck theory employs is the mutual information between two random variables. Mutual Information(MI) quantifies the amount of information that can be obtained about one random variable by observing another random variable. Consider the following two random variables **X** and **Y**:

| Random Variable | Event denoted by the random variable |
|---|---|
| **X** | You coming across this writeup. |
| **Y** | Marvel producing a new movie. |

It is safe to infer that **X** and **Y** are independent of each other. In other words, **X** does not provide any information about **Y**. It is commonplace to employ MI to articulate our inference in the context of Information Theory. Mathematically,

$$I(X;Y) = D_{KL}(P_{XY}||P_X P_Y) \tag{2.1}$$

where $I(X;Y)$ is the mutual information between **X** and **Y**, $P_{XY}$ is the joint distribution while $P_X$ and $P_Y$ denote the marginal distributions of of **X** and **Y** respectively. $D_{KL}$ denotes the Kullback-Leibler Divergence(KLD) between the joint and marginal distributions. Following from our inference about **X** and **Y** and the properties of KLD, $I(X;Y) = 0$ and $P_{XY} = P_X.P_Y$. Now let us move on to a more interesting scenario: where **Y** is a deterministic function of **X** subject to some assumptions. It has to be noted that this scenario acts as a proxy for the tasks we target using our favourite models of the day: Labels(**Y**) are conditionally dependent on the input data(**X**). For this, let us update the events being tracked by our random variables:

| Random Variable | Event denoted by the random variable |
|---|---|
| **X** | You coming across this writeup. |
| **Y** | You forming an opinion about this writeup. |

Consider that you do form opinions about writeups you come across. Subject to this assumption, it is evident that **Y** is a deterministic function of **X**. Also note that we have defined **X** loosely as coming across the writeup, not strictly as reading this writeup. In such scenarios, MI collapses to the entropy of **X** - the uncertainty in **X**. Before closing this subtopic, let me

present some caveats since ignoring them would be unfair. (1) **Y** being deterministic with respect to **X** is not always true. Since we are not programmed robots, our mental state almost always affects how we interpret a writeup. Hence the mapping is highly stochastic(stochastic as opposed to subjective). (2) Dependencies can be marginalized. Consider a typical scenario where you do come across this writeup, determine to get back to it at a later point in time but never could since you are involved in a thousand other interesting things. The joint probability is still collapsed but independent of **X**.

### 2.1.2 Downstream Task and Information Propagation

Lets move on. Now consider you want to communicate the opinion you formed above with your friend about this writeup. For the sake of simplicity, consider that there are only two possible messages and are the following:

| Message |
|---|
| This writeup sucks! |
| This is a reasonable writeup |

One of the most efficient ways to communicate this message would be to use a signal that has two states: Head and Tail of an unbiased coin.

| Signal | Message |
|---|---|
| Head | This writeup sucks! |
| Tail | This is a reasonable writeup |

The probability of your friend decoding the right message depends on the ability to associate the signal to its intended message. Let us extend this to a more imaginative scenario. You are still using an unbiased coin but your friend is using consonant phonemes in Dothraki[1] to decode the message. Now the problem starts becoming interesting. For your friend to decode the right message, two things need to happen:

- Identify which two of the twenty three consonants in Dothraki correspond to the signal states. For now let us assume that there is a one to one correspondence between the states.
- Attribute the identified states to the right signals.

Due to the large number of possibilities Dothraki consonant phonemes can encode(high model capacity), the error probability of the communication increases(Schumacher, 1995). To compensate, your friend needs to learn a projection function($\phi$) that projects the 23 phonemes into

[1]Dothraki is a fictional constructed language in the novel : *A Song of Ice and Fire*

2 plausible signal states that you perhaps used to encode the message. I argue that our current deep learning approaches fall under this scenario: The information capacity of these models is very high where as the tasks, due to the way they are formulated, do not require such high capacity. To make this argument more concrete, let us reformulate our random variables to denote a typical task in language technologies - speaker recognition or object identification:

| Random Variable | Event denoted by the random variable |
|---|---|
| **X** | Feature Representation of speech / image |
| **Y** | Speaker Label / Object Label |

This still is in line with our previous understanding: **Y** is conditionally dependent on **X**. However, the dimensionality of **X** is extremely large compared to **Y**, the model needs to discard the irrelevant information present in X. Since the information capacity of typical models is sufficiently high as well, the model capacity is utilized simply to capture the surface level associations(Goyal et al., 2017) and biases(Agrawal et al., 2017b) in the data distribution. In other words, models with high capacity can give us an illusion about task accomplishment just by memorizing the observed data distribution. The problem becomes even more pronounced in the context of tasks that lack strict objective measures like speech synthesis, image captioning and machine translation.

## 2.2 Case for Controlled De-Entanglement

I believe that complete isolation of input data into its independent causal factors of variation is not fully useful. A more attractive option is to control what and how much gets de-entangled in a a task dependent manner. It has to be noted that given a particular downstream task, some causal factors of variation might not be relevant, in which case modeling them would be unnecessary. Let us consider a data distribution X which consists of class examples $\{x_1, x_2..., x_n\}$, where each $x_i$ is described by attribute-set $(a, b, c)$. The prior distribution of X can be represented by a parameteric function g such that g maximizes the likelihood of X over the set of its attributes:

$$P_\omega(X) = g_\omega(a, b, c) \tag{2.2}$$

Note that the attribute-set can either contain individual entities or the relationships between them or both. To illustrate this, let us consider a toy-example where we build a binary classifier to predict if a given integer triplet is a Pythagorean triplet. Pythagorean triplets are a triplet of numbers that follow Pythagoras Theorem such as $\{3, 4, 5\}$ and $\{5, 12, 13\}$. In this task, the attribute-set consists of the relationship between the first two-elements of the triplet. If the model is able to discover this attribute, it can generalize for any given numbers. However, if we have a more complicated task like building a classifier for MNIST digits, then the attribute-set

has multiple first and second order relations like brush strokes, shape of the digits etc. The success of modelling $P_\omega(X)$, and ultimately the success on the downstream task, relies on how well can the model isolate these individual attributes from the observed data $X_t$. This isolation ability becomes even more important in case we want to regenerate the digits using a generative model. Mathematically, let us consider the posterior probability of a training instance $x_1$ expressed as

$$P_\theta(x_1) = f_\theta(x_1) \tag{2.3}$$

where $f$ denotes arbitrary function and $\theta$ denotes the parametric family used to model the distribution $X$. It can be seen that compositionality over an unseen training instant $x_{new}$ would be possible if $f$ is related to $g$. In other words, $f$ needs to intuitively have some information about the latent causal factors of variation that generated $X$ in the first place. In such scenarios, the test instance can be appropriately expressed as

$$P_\theta(x_{new}) = h(a, k(b, c)) \tag{2.4}$$

where h and k can be a novel combination of functions that embed these attributes in the manifold of original distribution of $X$. Not tracking the relevant factors of variation typically leads to model memorizing only the surface level associations leading to mode collapse and lack of diversity in the generated outputs. On the other hand, explicitly caring about the factors of variation can be seen as a way of incorporating inductive bias into the model and has the potential to avoid such pitfalls.

## 2.3 Implicit De-Entanglement in current models

Let us consider a typical deep learning architecture such as AlexNet(Krizhevsky et al., 2012). It is characterized by a series of convolutional layers (feature extraction module) followed by a pooling layer and a SoftMax layer(classification module). Note that while I mention AlexNet as an example, this abstraction can be extended to most sequence to sequence architectures with encoder as feature extraction module and decoder as the classification module(Rousseau and Tsaftaris, 2019) across modalities and tasks. It can be shown that the pooling layer acts as information bottleneck(Tishby et al., 2000) module in such architectures. I point out2.3.1 that in case of conventional Sequence to Sequence architectures deployed today, attention plays the role of information bottleneck module regulating the amount of information being utilized by the decoder. Such bottlenecks therefore control optimization in encoder decoder models leading to (1) Disentanglement of Causal Factors of variation in the data distribution(Higgins et al., 2016) (2) Marginalization of nuisance factors of variation from the input distribution(Oord et al., 2018). In case of models that employ stochasticity, two more effects can be observed :

(a) Posterior collapse or Degeneration due to powerful decoders and (b) Loss of output fidelity due to finite capacity decoders.

### 2.3.1 Implicit De-Entanglement: Deterministic Attention vs Stochastic Attention

I will illustrate this sub section with a typical generative model of speech: Text to Speech. Consider that we are interested in building a code mixed version: a model that can accomodate two languages in a single utterance. Let us also consider a speech corpus $X$ consisting of languages $\{l_1, ..., l_n\}$, where each $l_i$ might comprise of multiple speakers. Let $y_1$,...,$y_n$ denote acoustic frames in the target sequence $y$ while $x_1$,...,$x_n$ denote the encoded text sequence $x$ from one of the languages. A typical attention based encoder decoder network such as Tacotron(Wang et al., 2017b) factorizes the joint probability of acoustic frames as product of conditional probabilities. Mathematically, this can be shown as below:

$$P_\theta(y|x) = \Pi_{t=1}^{t=n} P(y_t|x_1...x_m, s_t) \tag{2.5}$$

where $s_t$ is a decoder state summarizing $y_1$,...$y_{t-1}$. Parameters $\theta$ of the model are set by maximizing either the log likelihood of training examples or the divergence between predicted and true target distributions. At each time step t in these models, an attention variable $a_t$ is used to denote which encoded state of $x_1...x_m$ aligns with $y_t$. The most common form of attention used is soft attention, a convex combination from encoded representation of input text. It has to be noted that soft attention in such scenarios is essentially a latent deterministic variable that computes an expectation over the alignment between input and output sequences. Empirically, soft attention provides surprisingly good alignment often correlating with human intuitions. Having said that, to synthesize speech from different languages at test time, the generative process needs to disentangle appropriate individual language attributes from observed data $X_{obs}$ and also compose them to form a coherent utterance in the voice of desired speaker. However, presence of deterministic alignment method limits the ability of models to generalize to such scenario.

On the other hand, variational attention(Deng et al., 2018) provides a mechanism to factorize this alignment and mediate the generative process of $y$ through a stochastic variable $z$. In addition, both soft and hard attention mechanisms can be shown as special cases of Evidence LowerBound(ELBO)(Deng et al., 2018). Therefore, incorporating latent stochastic variables allows us to directly optimize ELBO. In this context, model parameters are set by maximizing the log marginal likelihood of the training samples. But direct maximization of this marginal in the presence of latent variable is often difficult due to expectation involved. To address this, a recognition network $q$ is employed to approximate the posterior probability using reparameterization. It is interesting to note that the encoder in a deterministic Seq2Seq network functions as the recognition network in latent stochastic variable models and is incentivized to search over variational distributions to improve ELBO. Intuitively, the lower bound is tight

when the inferred variational distribution is closer to the true posterior of the data. This has a sense of grounding in our understanding of the task as well. Perhaps there are a set of universal phonemes, around 120, which should be enable us to speak in any language subject to the phonotactic constraints of the language. Having such prior information greatly reduces the model size as opposed to naively using a combination of all phones from all the languages to build a polyglot model.

## 2.4 How to accomplish De-Entanglement?

I posit that designing learning paradigms such that we explicitly control de-entanglement of relevant factors of variation while marginalizing the nuisance factors of variation leads to massive improvements. Such an approach, I claim, leads to further advantages in the context of both generative processes: in terms of generation of novel content and discriminative processes: in terms of robustness of such models to noise and attacks.

I identify four ways to computationally accomplish De-entanglement:

- (1) By employing suitable priors about task or data distribution
- (2) By incorporating additional adversarial or multi task objectives within the model
- (3) By utilizing a different divergence objective
- (4) By employing an alternative formulation of probability density estimation

In this chapter, I will present detailed background on (1) and (2) since they are very closely related to experiments in the chapters that follow.

## 2.5 De-Entanglement by Priors

The most popular approach to obtain isolation of factors of variation in neural models is by employing stochastic random variables. This approach provides flexibility to jointly train the latent representations as well as the downstream network. It has been observed that the latent representations resemble disentangled representations under certain conditions (Chen et al., 2018b; Burgess et al., 2018a; Esmaeili et al., 2018a; Ansari and Soh, 2018). Note that although obtaining such degenerate representations is considered typical, it is not the only manifestation: it also manifests as continuous representations(Ravanelli and Bengio, 2018) and other abstract phenomena(e.g. grounding). I argue that explicitly controlling what and how much gets de-entangled (Burgess et al., 2018a) is better than implicit disentanglement as is followed today(Locatello et al., 2018).

### 2.5.1 Analysis of role of priors in Latent Stochastic Models

The choice of priors plays a significant role in optimization within latent stochastic models. In this subsection, we present an analysis to show that priors control the disentanglement of causal factors of variation in such models. Let us consider the ELBO being optimized in a VAE:

$$E_{q_\phi(z|x,c)}[logp_\theta(x|c,z)] - |D_{KL}(q_\phi(z|x,c)||p_\theta(z|c))| \tag{2.6}$$

where the first term is the reconstruction error while the second is the divergence between approximate and true posteriors. Here are the four phenomenon that are manifested due to choices of priors:

(1) *Disentanglement or Factorization of causal factors of variation*
The KL divergence forces the posterior distribution output by encoder to follow an appropriate prior about the data generation process. Typically, prior space is assumed to be continuous distribution and a unit Gaussian. The global optimum value for the divergence in such cases is 0 and is reached only when both the distributions exactly match each other. Since the prior information about the data generation process typically involves some causal factors of variation of the data, this naturally is assumed to translate to a constraint on the encoder to track such factors. Thus, such models have potential to disentangle or factorize the causal factors of variation in the distribution.

(2) *Marginalization of Nuisance Factors of Variation*
It has to be noted that during training optimization is performed in expectation over mini-batches. Therefore, the expectation of KL divergence can be rewritten as related to the amount of mutual information between the latent representation and the data distribution (Makhzani and Frey, 2017a). As this divergence decreases, the amount of information the encoder can place in the latent space also decreases. As a result, encoder is forced to discard some nuisance factors that may not have contributed to the generation of data. Thus, KL divergence also forces the model to marginalize the nuisance variables.

(3) *Posterior Collapse due to simple priors*
Consider the scenario where the prior is too simplistic, such as the aforementioned unit normal distribution. In such cases, the model is incentivized to force the posterior distribution to closely follow the Gaussian distribution (Chen et al., 2016). Typically the decoders in variational models are implemented using universal approximators such as RNNs. In the context of a TTS systems, decoder segment of the acoustic model along with the neural vocoder act as the decoders. Since such decoders are very powerful, they are able to learn or ignore the priors about data distribution themselves and hence marginalize out the latent representation input from the encoder. In other words, the prediction of next sample is based solely on the marginal distribution at the current timestep which can be implemented by learning a dictionary per time step. Therefore, the encoder is no longer forced to track the causal factors of variation in the data. This is referred to as posterior collapse or mode collapse.

(4) *Loss of output fidelity due to complex priors*
A reasonable and intuitive solution to posterior collapse is making the prior space more complex thereby pressurizing the posterior distribution to track the prior space more closely. For instance, (Burgess et al., 2018b) attempt to accomplish this by adding a hyperparameter $\beta$ to promote disentanglement and gradually increasing channel capacity, something that increases loss. However, it has to be noted that simply making the prior distribution arbitrarily complex also perhaps leads to unreasonable constraints on the decoder. For instance, in scenarios that have categorical distribution as their output (tasks such as language modeling, machine translation, image captioning among others) it is unintuitive to assume that the true prior that generates latent distribution is a Gaussian when the likelihood is based on discrete sequential data in such tasks. Having such strong priors directly affects the reconstruction ability in these models.

Therefore, priors in latent stochastic models play a significant role in the optimization and facilitate disentanglement of causal factors of variation on the one hand, as well as help the ability of the model to reconstruct the data distribution on the other. Having this knowledge enables us to engineer various components to tune the model behavior as per our requirements.

## 2.6 De-Entanglement by Divergences

I identify two paradigms within divergence based approaches that differ by the type of divergence: (b) Generative Adversarial Networks (GANs) and (b) Reinforcement Learning

### 2.6.1 Adversarial Learning

Here are the ways in which adversarial training has been employed to isolate information from the data distribution:

- **Regularization**: In (Song et al., 2020), authors employ a distance covariable based decorrelation regularization. In (Lin et al., 2019), authors make an observation that latent traversal provides a useful signal to detect disentanglement. Based on this, they employ contrastive regularization.
- **Incorporation of additional information**: In (Gadelha et al., 2017, 2019) authors observe that addition of a differential projection module allows inference of underlying 3D shape distribution from 2D views. In (Kaneko et al., 2019), authors incorporate a noise transition model to learn clean label distribution in the presence of noisy labels. In (Wang et al., 2018a), authors attempt to recover the nuisance variable from the representations.
- **Structure**: In (Kim et al., 2019), encoder is trained to discover the multi manifold structure and abstract features of the data. In (Sáez Trigueros et al., 2018) authors train an

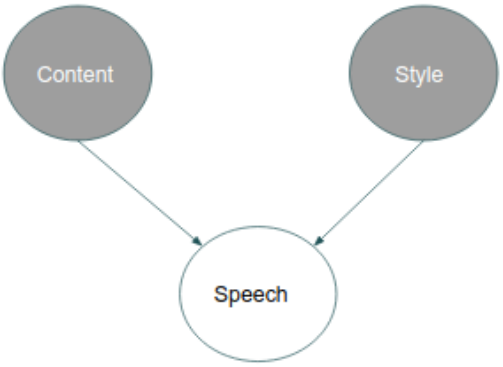



FIGURE 2.1: Graphical Model depicting De-Entanglement in Speech

identity latent space that separates identity related and identity independent attributes. In (Qian et al., 2020) authors encode F0 and speaker identity into different latent spaces and in (Chen et al., 2020) authors separate accent invariant and accent specific characteristics from acoustic features. In (Xiang et al., 2020) authors learn personalized facial landmarks that combine the identity of the target with expressions and poses from a different subject using dictionary learning.

- **Multi objective learning**: (Makhzani and Frey, 2017b) identified that different priors result in different decomposition of information in the latent space. In (Yang et al., 2019a), authors attempt both domain transfer and task transfer to learn a disentangled representation. In (Yang et al., 2019b), the model is trained to accomplish both style transfer and style removal. In (Choi et al., 2020), inference networks are trained to match speaker identity between two different modalities. In (Ghosh et al., 2018), multiple generators are employed to capture diverse high probability modes. In (Ding et al., 2020), authors employ unsupervised learning for latent geometric transformation and supervised learning with adversarial excitation and inhibition mechanism. In (Luo et al., 2020) authors learn four different latent spaces to capture style, content, pitch and rhythm information. In (Jha et al., 2018), authors learn two complementary sub space using weak supervision in the form of pairwise similarity labels.

Perhaps the work closest to this dissertation is (Luo et al., 2020) where authors use two classifiers to separate style and content latent space.

## 2.7 De-Entanglement in Speech

In my dissertation, I view speech utterance as a confluence of two types of information, depicted in the figure 2.1. With this perspective, any real life application that employes speech processing can be viewed as an information processing system. In other words, as the speech utterance passes through different stages in the application such as Source Separation, Voice

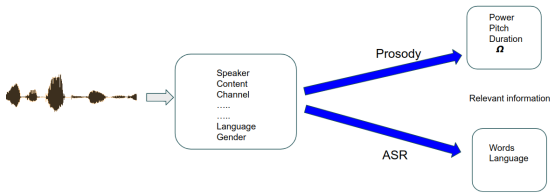



FIGURE 2.2: Model depicting the process of De-Entanglement in Speech

Activity Detection, Speech to Text, etc, information realized as both content and style are manipulated from one form to another.

I believe this provides a richer and more interesting way of modeling speech in the context of a real life application. For example, in the sentence "I need to transfer 500 dollars" spoken by a customer via a mobile phone, style information initially includes information about the channel as the utterance passes through voice activity detection. After this module when the utterance is passed through source separation and speech to text components, style encompasses the speaker and language information while content is realized in the form of phone sequence. Finally when the utterance is passed through intent recognition system, words form the content information while intent of the utterance becomes the style information.

It has to be noted that separating the causal factors of variation in a speech utterance could be accomplished in a latent fashion. Examples of approaches that accomplish such separation are deterministic processes like Matrix Factorization and stochastic processes like Variational Auto Encoder(Burgess et al., 2018b). However, in such approaches there is no grounding for the discovered causal factors. In the world of De-Entanglement, I ground the causal factors in linguistic concepts of content and style.

Another observation is that all of the latent factors of variation that resulted in generation of the data distribution are equally significant for every task. For instance, as depicted in the figure 2.2, some speech features become important for particular tasks while the others are marginalized. If the task at hand is related to prosody, the relevant features are pwer, pitch and duration. De-Entanglement natively supports this by transforming the content and style information types depending on the task at hand. In addition, consider a task like Voice Conversion: De-Entanglement also provides a mechanism to manipulate one type of information, say content manifested as phonetic composition of the utterance, to control another, say style manifested as speaker information.

# 3

# FALCON

*"You are in dire need of an upgrade.*
*Systemic, Top to bottom.*
*100 point restoration. Thats why I am here"*

- Tony Stark, *Captain America Civil War*

FALCON is the toolkit I am developing to support experiments in this dissertation. FALCON is primarily built as an update to our existing framework Festvox(Anumanchipalli et al., 2011) and is intended to function as neural extension to our voice building suite. In this chapter, I first extend the research question posed in chapter 1 and contextualize it with respect to the three challenges. I will describe the desiderata for the toolkit that can assist in testing the hypotheses of De-Entanglement. I then present the architecture of FALCON and highlight two experiments - one in a generative model setting and the other in a discriminative model setting.

## 3.1 Restating the research question

In chapter 1, I have presented the overall goal on this dissertation, to answer the research question: "**How do we build ubiquitous speech technologies?**". In this chapter, I will make it more specific and break the question into three parts contextualized by the individual challenges. For each of the sub questions I will briefly explain the desiderata for the toolkit that allows experimentation and validation of the hypotheses put forth by De-Entanglement.

- ***Scalability Challenge**: How to build speech technologies that are scalable?*
  In this dissertation I use the term scalability to refer to the ability of De-Entanglement to apply to multiple languages and linguistic phenomena. I am specifically constraining myself to investigate code switching. Code switching is a linguistic phenomena where multilingual speakers tend to use more than one language in a single utterance. For instance, consider the sentence

  "*IPL ki opening ceremony Kolkata ke Saltlake Stadium mein hua*".

  This sentence contains linguistic units(words) from two different languages - Hindi and English. Therefore, I am interested in tools that natively support multiple languages and allow flexible experimentation by manipulating and combining the linguistic attributes at various levels.

- ***Flexibility Challenge**: How to build speech technologies that are flexible?*
  In this dissertation I use the term flexibility to refer to the ability of De-Entanglement to apply to detection and generation of various prosodic phenomena observed in human speech. Therefore I am interested in tools that allow me to extract information at different levels from the speech utterance.

  *Detection of Para linguistic Events* - Paralinguistics refer to aspects of spoken utterance that do not involve words. Typically paralinguistic events add additional shades to the meaning already being conveyed by the linguistic units (words). In the context of De-Entanglement, I pose this as the problem of representation learning.

  *Generation of Prosodic Phenomena* - Generative models are attractive from the perspective of measuring flexibility since the outputs can be subjectively evaluated to ensure the hypotheses align with the observations. For instance, consider the following sentences:

  ***JOHN** loves Mary*
  *John **LOVES** Mary*
  *John loves **MARY***

  All the three sentences are characterized by the same linguistic units(words). The linguistic units also follow the same temporal ordering. The difference in the intended meaning of the sentences rises from the prosodic focus on the individual words. De-Entanglement hypothesizes that this prosodic phenomena can be modeled by *style* variable where the linguistic units themselves are modeled by *content* variable. Subjective evaluations make it easy to evaluate whether and to what extent this hypothesis is valid.

  Therefore I am interested in tools that support generative modeling, specifically those that make it easy to incorporate different types of prosodic information. For instance, validating hypotheses about *content* requires experimentation at the level of phonemes whereas for validating hypotheses about *style* it might be important to experiment at sentence and phrase level.

- ***Explainability Challenge**: How to build speech technologies that are explainable?*
  In this dissertation I am restricting myself to one aspect of Explainability - Justification. With respect to De-Entanglement, I build on Neural Module networks and adapt them to accomplish justification of predictions by a model. The specific architecture I have chosen is characterized by the presence of two tasks: a primary task and an auxiliary task whose output is employed by the primary task. Therefore, I am interested in tools that are modular and support multi task learning - tools built so that it is natural to integrate other tasks / applications.

## 3.2 FestX

I refer to the combination of Festival(Black et al., 1998a; Taylor et al., 1998) and Festvox(Anumanchipalli et al., 2011; Black, 2006a) as FestX. Based on the desiderata mentioned in the previous section, I have employed FestX as the toolkit for experiments in this dissertation. Festival encodes speech utterances as Heterogenous Relation Graphs(HRGs) and Festvox contains a suite of routines to manipulate HRGs in order to model speech. At a high level, HRGs in Festival are characterized by objects referred to as *Relations*. Relation objects can be either trees or list of trees. For instance, *SYLLABLE STRUCTURE* object is a list of trees containing *WORD* and *SEGMENT* relations. Relations are composed of *ITEMS* which are the most basic level of representation used in Festival. Items have features implemented as dictionaries.

### 3.2.1 Evolution of FestX

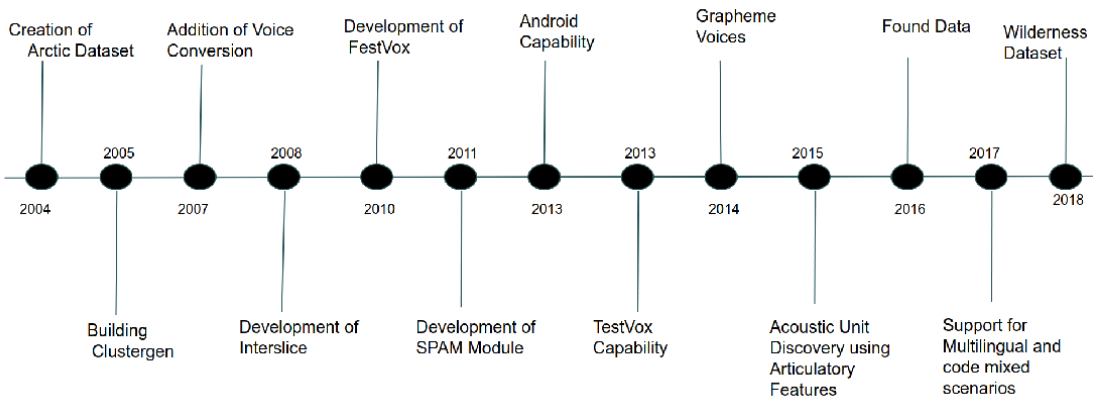



FIGURE 3.1: Timeline of developments to FestX within LTI leading up to FALCON. Note that while there has been lot of work on Unit Selection speech synthesis, I am not depicting those.

The Evolution of FestX is depicted in the figure [1] 3.1. It has to be noted that while there have been several developments in FestX within and outside my organization, I have mentioned those milestones that I consider are closely related to FALCON and De-Entanglement.

[1] Inspired by (a) the timeline of Marvel Phase 4 presented by Kevin Feige at Comic-Con 2019 and (b) Timeline of development of QA systems in CMU presented by Eric Nyberg during LTI Colloquium, 2019.

## 3.3 FALCON

FALCON is the neural extension to FestX. In this section I will first present a background on deep neural generative models. Following this, I present a case for one of the founding objectives of FALCON - unification of generative models for speech. I will then present an architecture of FALCON.

### 3.3.1 Background

Deep Neural Generative models have seen a tremendous amount of progress in the recent past. These models factorize the joint probability of the original data distribution and an optional local variability as a recursive product of conditional distributions. Typical implementation of such models therefore follows an auto-regressive framework although other formulations have been suggested as well. Such models have been shown very effective in addressing one of the major challenges with conventional vocoding techniques - fidelity. The principles used in deep generative models for speech such as dilated convolutions are also employed in varied domains such as in machine reading and named entity recognition(Strubell et al., 2017). These models have been applied to speech as well and this seems a natural progression. Generating natural sounding speech has applications in a multitude of speech processing technologies such as Google Duplex, voice over for characters such as Rocket in Guardians of Galaxy, etc. Hence understandably there has been a rise in interest in applying deep neural generative models for speech generation. There are a lot of works towards improving the other issue - flexibility. Given that speech is the longest form of generative models, they also form key enabling technologies and act as horizontal enabling layers or first class objects in a multitude of technologies going forward. Hence it is crucial to have a good understanding of the underpinnings of such a fundamental technology, from a perspective of speech to begin with, but span across domains as well. We believe that a decent strategy to work toward this goal is to employ inductive biases as scaffolding while building these models.

There have been continuous and significant improvements in both the aspects of speech generation - fidelity and flexibility. Auto regressive models such as (Van Den Oord et al., 2016), flow based models such as (Prenger et al., 2018a) have shown to generate audio that rivals the quality of natural speech. Approaches such as (Taigman et al., 2017b,a) have shown ways to incorporate inductive biases into the generative process. These techniques have been utilized in building highly flexible systems capable of generating different styles(Hsu et al., 2018c; Wang et al., 2017d; Skerry-Ryan et al., 2018; Wang et al., 2018d; Skerry-Ryan et al., 2018) of speech and ability to build voices from noisy(Hsu et al., 2018a) or very minimal data(Chen et al., 2018d).

### 3.3.2 Case for unification of Generative Models

The following tasks can be combined under the generative processes for speech:

- *Text To Speech Synthesis*: Deals with generating speech conditioned on the textual constraints
- *Voice Conversion*: Deals with generating speech conditioned on textual and speaker constraints
- *Speech Enhancement*: Deals with generating a cleaner version of speech.
- Speech Coding
- Source Separation

As pointed out in (Zhang et al., 2019a), the differences between various speech generation tasks seem narrow. We posit that it is natural to assume more progress if we combine all of these. There has been work in this direction. In (Hsu et al.), authors unify speech synthesis and speech enhancement. In (Biadsy et al., 2019a), authors show that a well trained speech synthesis model can perform speech source separation. In (Zhang et al., 2019c), authors blur the lines between multilingual speech synthesis and cross language voice conversion. In (Zhang et al., 2019a), authors present a case to unify voice cloning and text to speech synthesis. They extend Tacotron with a shared dual attention model that is common to both voice conversion and TTS. This is also based on the understanding that speech synthesis is a good proxy for testing the effectiveness in tasks such as speech coding and speech chain style architectures.

In (Hsu et al., 2017), authors pose voice conversion as a unified generative modeling task. We follow their framework of unification but employ priors to facilitate the conditioned generation part. In (Kaneko and Kameoka, 2017; Fang et al., 2018; Hosseini-Asl et al., 2018; Kameoka et al., 2018), authors attempt to learn forward and inverse mappings simultaneously using adversarial and cycle-consistency losses. This allows the model to jointly capture sequential as well as hierarchical structures while preserving linguistic information. My work is most similar in spirit to the approaches in (Chou et al., 2018) and (van den Oord et al., 2017a; Chorowski et al., 2019a). I engineer our encoder and decoder to capture different types of information as in these works.

## 3.4 Architecture and Capabilities of FALCON

FALCON extends FestX and hence maintains the structure of HRG. The graph nature of HRG is applied to design neural end to end systems within FALCON. This is detailed in the subsection 3.4.1. Since FALCON is aimed at building speech technology, the extension ships with its own personal speech based assistant *JUDITH* with a variety of functionality. This is detailed in the subsection 18.4.

### 3.4.1 HRG based Modularity

FALCON is built using three objects - *layers, blocks and models*. I will elucidate these objects using a well known architecture in speech literature - WaveNet(van den Oord et al., 2016; Chorowski et al., 2019a).

Layers denote the fundamental objects that perform a transformation of the input variable. In the context of WaveNet, the dilated convolution layer is implemented as a layer. Blocks often include a composition of layers and form the building blocks in a typical model employed to test hypotheses of De-Entanglement. In the context of WaveNet, stack of dilated convolutions with residual connection and gated combination is implemented as a block within FALCON. Model encompasses organization of blocks. Different models share the blocks and layers and hence the mapping from blocks to models is one to many, mirroring the mapping from layers to blocks. For any experiment, FALCON expects the user to build multiple systems and run multiple experiments with some baselines and multiple topline approaches. Such modular and connected organization allows FALCON to draw conclusions across systems and experiments. An example of this is presented in subsection 18.4.1.

## 3.5 Experiments

In this section I will present two case studies that highlight the functionality of FALCON.

### 3.5.1 Case Study 01: Generation of Expressive Speech

For this experiment, I have used the audiobook data released as part of Blizzard Challenge 2018. The database used was provided by Usborne Publishing Ltd. and consists of the utterances from children's audiobooks spoken by a native British female speaker. We are given about 5 hours of speech data. We have removed the 'bell' sounds which were present in the speech and all the other expressions like 'uh..', 'hm..'. The total duration of the audio is approximately 4.5 hours after segmentation. In this experiment, I compare the acoustic modeling capability of FALCON. To demonstrate this, I build two identical systems which differ only in their acoustic models. Our base voice uses Random Forest based acoustic model while model with FALCON employs our implementation of Tacotron end to end acoustic model.

#### 3.5.1.1 Tokenization and G2P

We consider any text entry separated by white space as a token. From the training data, we have observed that there are instances where the calendar entries such as 1859 are represented both in numeric form as well as the expanded form(eighteen hundred and fifty nine). Further, we noticed that tokens like hyphen played a varied role in the pronunciation of the accompanying word(s). For example, here are the different instances where hyphen was used:

- To indicate hesitation in speaking or transient sounds. This can be observed in the pronunciations for words 'S-s-sorry', 'Cr-r-rock', 'W-w-what'.
- To indicate repitition: 'tap-tap-tap', 'glug-glug-glug'.
- As a placeholder joining two words: broken-hearted bulls-eye chinny-chin-chin
- As a phrase break: 'You cant pretend to not know John Canty - your own father'

We have used a decision list based disambiguator built following (Yarowsky, 1997) to tokenize such occurrences appropriately. Once tokens are obtained, we have analysed the usage of two different phonesets: US phoneset which performed G2P mapping based on CMU pronunciation dictionary and UK phoneset which performed G2P mapping based on Unilex pronunciation dictionary. As we did not find striking differences in the final voice quality between the two phonesets, we have continued the system design with US phoneset.

#### 3.5.1.2 Pronunciations for OOV words

There were words in the training corpus which were absent from the CMU pronunciation dictionary. Most of these words were observed to be proper nouns and therefore, we concluded that it would be better to build a generic model to predict the pronunciations for such words. For this, we have employed word to phone mapping (Elluru et al., 2013) using automatic epsilon scattering method (Black et al., 1998c): We first use epsilon scattering method to align the letters and phones for a set of words in the given database. Each letter is assumed to be specifying a phonetic correspondence to one or more phones and in case a letter is not mapped to any phone then epsilon is used as a placeholder. We first aligned the letter (graphemic) and phone sequences by estimating the probabilities for one letter (grapheme) G to match with one phone P. We then used string alignment to introduce epsilons maximizing the probability of the alignment path of that word. Once all the words have been aligned, the association probability is calculated again and this is repeated until convergence. Once a reliable alignment has been obtained, we use a statistical mapping from letters to phones which can be seen as maximizing the expression:

$$\prod_{i,j \in S,W} Prob(s_i|w_j) \tag{3.1}$$

for each word w where $w \in W^d$ is the word in the database with a vocabulary(W) of size d.

#### 3.5.1.3 Improving the labels

We perform text segmentation of the utterances at the segment level (phone). However, all our subsequent analyses are carried out at a lower level, which is realized by dividing each phone

Table 3.1: Analysis of Improving the labels

| Pass | No. of Moves | MCD | F0 Error | Duration Error |
|---|---|---|---|---|
| 1 | 73898 | 4.671 | 28.829 | 0.943 |
| 2 | 63587 | 4.646 | 28.736 | 0.943 |
| 3 | 57072 | 4.634 | 28.814 | 0.943 |
| 4 | 53376 | 4.641 | 28.795 | 0.946 |
| 5 | 51033 | 4.636 | 28.835 | 0.948 |
| 6 | 48881 | 4.627 | 28.823 | 0.946 |
| 7 | 48460 | 4.626 | 28.911 | 0.946 |
| 8 | 46013 | 4.618 | 29.022 | 0.946 |
| 9 | 44776 | 4.620 | 28.926 | 0.947 |

into three states, corresponding to the begin, middle and end states of a phoneme. Therefore, each frame is labeled as one of these states and these initial labels for the data are obtained using EHMM technique. We then tried to improve the labels using the procedure outlined in (Black and Kominek, 2009): We examine each segment boundary and consider moving it forward or backward ( by one frame) and investigate whether this decreases the distance between original and predicted frame. This process is performed over all the labels and then the models are rebuilt. The distance is measured in terms of unnormalized MCD including the energy coefficient but not the deltas. We have performed 10 iterations over the entire database as the improvement in MCD stopped at that point. The results of this procedure have been outlined in the table 3.1. We have observed that the passes did not necessarily result in an improvement in the prosodic models.

#### 3.5.1.4 Acoustic Feature Extraction

For each of the states obtained from segmentation, we extract acoustic feature vectors over a 5ms frames obtained by applying a hamming window. Spectral representation that we use is MCEPs and were extracted using the SPTK toolkit (Group et al., 2009). The order of MCEP was chosen to be 24 with a frequency warping factor of 0.42 and a small value ( 1.0E-08) was added to the periodogram. For F0, we interpolate between unvoiced section ensuring breaks during silences and then apply a post smoothing using a 25 ms window.

#### 3.5.1.5 Outlier Removal

In the context of audiobook synthesis, selection of appropriate examples for building the data driven statistical models is necessary as the statistics may be skewed due to the presence of outliers. We perform this based on the state durations. For each state, we remove the examples that have values farther than 1.5 times the standard deviation of the mean value for the state.

#### 3.5.1.6 Acoustic Modeling

**Base Voice** For the base voice we have used Random Forest (Black and Muthukumar, 2015) as the model for learning a mapping from the linguistic features to the acoustic features. The central idea is based on feature bagging - to replace the original MCEP prediction tree in the CLUSTERGEN framework with multiple prediction trees trained using random linguistic features. For this, we built 20 different trees for each state, by varying the probability of each feature being picked. Then, to form a forest, we average the predicted values from the trees. Based on the observations from (Black and Muthukumar, 2015), we pick the best 3 trees based on the MCD on a held out development set. Predictions at test time are made by averaging the predictions from the selected individual regression trees.

**FALCON Voice** The acoustic model is based on Tacotron(Wang et al., 2017c) Seq2Seq speech synthesis system We have used phones as the input instead of characters. We have not performed masking the loss value for padded frames as is typically done in Seq2Seq models. We found that forcing the model to predict (zero) padded frames helps the model better predict end of sentence as mentioned in(Wang et al., 2017c). Since adjacent frames seem to be correlated, our decoder predicts 3 frames per timestep. We have used a batch size of 64 to train the baseline model.

#### 3.5.1.7 Pruning Frames for Base Voice

In addition to the outlier removal mentioned in section 2.6 which was performed using state duration, we also perform a frame pruning based on the spectral features and remove the frames that have the predicted values farther than a predetermined threshold value. These frames correspond, in general to the areas where the model consistently makes mistakes. After this, we rebuild the final voice with the pruned frames.

#### 3.5.1.8 Identifying quoted speech type and characters from stories

The data provided for building voices consists of abridged plays such as 'Androcles and the Lion'. In other words, the data is a continuum of discourse that runs between characters in the play. Subsequently, the original speaker attempts to imitate the persona of characters while recording the content. This attempt is manifested in the form of prosodic variations in the provided recordings. Therefore, we hypothesize that it is beneficial to tag the data with the speech type ( quoted vs narrated ), character identity (Zhang et al., 2003) and use this information during acoustic modeling (Theune et al., 2002).

### Quoted vs Non quoted speech

We define a portion of story as quoted if it is quote annotated. In addition, we have also annotated if the portion of text was a continuation from previous sentence or a new one. We have annotated these segments using SABLE and an example is shown below:

```
<QUOTE TYPE="NEW"> It's my daughter, Hermia,  </QUOTE> he explained.
<QUOTE TYPE="CONT"> I want her to marry this man, Demetrius. </QUOTE>
```

An informal inspection of the text has not revealed a significant number of nested quotes leading to 'story within a story'. This might be because the provided content is aimed at children. Therefore, we have not specifically annotated nested quotes. In scenarios where we did encounter them, we have split the sentence into different utterances. During acoustic modeling, we use this information as another label.

### Identifying character type in stories

Associating each utterance to a character provides a way to render the story mimicking the characters. This is essentially dealt as a Named Entity Recognition task (Zhang et al., 2003). Additional linguistic information was used to identify the proper names. In our approach, we borrow this idea but confine ourselves to a maximum of three characters. Nominally, we associate the three characters to (1) Narrator (2) Protagonist and (3) Antagonist. Through an analysis of the text, we have come up with the following basic approach for assigning character labels to text:

- Text without 'quote' attribute is labeled by the tag 'Narrator'.
- For every quoted utterance after the narrator, we alternate between characters 'Protagonist' and 'Antagonist' if the quote is labeled 'NEW'.
- If the quoted text is labeled as continuation ('CONT'), we repeat the label of the character.
- If there is a sequence of more than three utterances tagged as 'Narrator', we drop the speaker state and label the next encountered character with the tag 'Protagonist'.

We have observed that the stories differ in the consistency of speaker - speech relationships. However, for these experiments we have followed the rudimentary approach described above and have not specifically handled this inconsistency.

#### 3.5.1.9 Identifying inter sentential events and intra-sentential relations

Stories are often characterized by flow of emotions. In addition to mimicking the characters, the original speaker has also attempted to render the perceived emotions while recording the content. Similar to characters, the manifestation of these emotions too is in the form of prosodic variations. Therefore, we hypothesize that we can model the flow of emotions in a story by modeling the prosody of reader. In our current submission, we investigate two approaches for realizing this:

- We identify semantic units (or) events that indicate change of state within the sentences.
- We employ Rhetorical Structure theory to identify contrastive relationship between different sentences within the paragraph.

**Event Detection within sentences**

We define events as the semantic units that express a change of state or an action in world. Events are comprised of the predicate denoting the action (usually a verb or a noun) and a set of arguments: entities that act on the predicate (agent) or that the predicate acts on them (patient, theme). Conforming to this definition, event detection is an information extraction task where, given a sentence, we try to automatically detect the predicate and the event arguments.

Because of their rich structure, events are good semantic representations that provide useful information for the prosody model. Most times, predicates contain the most important information the speaker wants to convey in an utterance, something crucial for the prosody model. Although predicates are the main information carrier, event arguments also provide important information. Mostly prominent in dialogue or story-telling scenarios, event arguments might carry supplemental information for a previous utterance. In such cases, the speaker wants to focus on those entities more than the action, which shows the importance of Event Detection and comparison of events across utterances.

To identify events, we consider each sentence independently. For each sentence we provide a list of actions and a set of participating entities, which we use as features in our prosody model. Given the lack of annotated data, our Event Detection system is a primarily rule-based system. Our system uses the Stanford CoreNLP parser (Manning et al., 2014) in order to generate a list of candidate verbal and nominal events per sentence. Then we map each of those candidates to a frame provided by FrameNet (Baker et al., 1998), which represents the semantic type that a word belongs to. Finally, we use a curated subset of FrameNet frames that represent events in order to determine whether or not the mention is an event. In order to extract the event arguments, we use the Dependency Graph in combination with the NER model provided by Stanford CoreNLP.

**Identifying intrasentential relations using Rhetorical Structure Theory**

Discourse theory describes the high level organization of speech and text. Specifically, hierarchical discourse representations such as Rhetorical Structure theory (RST) provide tree shaped parsing of a story that can be used for prosody modeling. In simple words, given spans of text, RST describes the relationship between them. We hypothesize that identifying contrastive rhetoric and emphasizing the contrast leads to soulful synthesis. For this, we use an approach inspired by (Ji and Eisenstein, 2014): We first learn projection function that learns a mapping from surface level representation to the discourse label on a gold set of discourse labels (Feng and Hirst, 2012). We then apply the projection function on current data to obtain discourse labels for each utterance in the story. However, our implementation differs from (Ji and Eisenstein, 2014) in a couple of ways: (1) We use the latent representation obtained using a Recurrent Neural net based language model instead of lexical features. This allows us to also bypass the shift reduce parse based implementation(Marcu, 1999). (2) The projection function we learn is non linear instead of a linear function. The steps we followed are outlined below:

- We segment the story into different Elementary Discourse Units (EDUs). Instead of using sequential data labeling (Hernault et al., 2010), we mark each utterance as a different EDU while parsing stories.
- We build an LSTM based language model on WSJ corpus. We then pass the entire story corpus through trained LM, reinitializing the hidden state after every story. For each utterance, we obtain the hidden representation. In our implementation, the RNNLM had a single hidden layer with 512 units and achieved a test perplexity of 5.32 on WSJ corpus. Thus for each utterance in the story, we end up with 512 dimensional representation.
- We then learn a projection function that maps the latent representation of the utterance to its discourse label.

#### 3.5.1.10 Evaluation

Evaluation was performed in the form of listening tests with 20 native students using (Parlikar, 2012a) with naturalness as criterion in terms of Preference test. The results can be seen in the table 3.2.

Table 3.2: Preference Test for Base Voice vs FALCON Voice

| Config | Clustergen | FALCON | No Preference |
|---|---|---|---|
| Naturalness | 32 | **64** | 6 |
| Accentuation | 36 | **59** | 5 |
| Expressiveness | 37 | **42** | 21 |

### 3.5.2 Case Study 02: Paralinguistic Event Detection from Acoustics

In this case study, I present experiments on detection of three meta linguistic events from acoustic signal using FALCON: 1) Identification of Styrian dialects, 2) Detection of presence of Orca from its vocal characteristics and 3) Classification of Baby sounds. To this end, I present experiments that compare FALCON against our previous baseline towards detection of paralinguistic events - modified version of SoundNet.

#### 3.5.2.1 Baseline System based on utterance level representations

SoundNet (Aytar et al., 2016) is a convolutional network operates on raw waveforms and is trained to predict the objects and scenes in video streams at certain points. After the network is trained, the activations of its intermediate layers can be considered a representation of the audio suitable for classification. It has to be noted that SoundNet is a fully convolutional network, in which the frame rate decreases with each layer. Since we need to predict the emotions with reasonable recall, we cannot extract features from the higher layers of SoundNet directly.

The original SoundNet network has seven hidden convolutional layers interspersed with max-pooling layers. Each convolutional layer essentially doubles the number of feature maps and halves the frame rate. The network is trained to minimize the KL divergence from the ground truth distributions to the predicted distributions. In the original SoundNet architecture, the higher layers have been subsampled too much to be used directly for feature extraction. In order to fully exploit the information in the higher layers, we train a fully connected variant of SoundNet (see Fig. 9.1). Instead of using convolutional layers all the way up, we switch to fully connected layers after the 5th layer. We have also changed the input sampling rate to $16\,\mathrm{kHz}$ to match the provided data.

#### 3.5.2.2 Baseline System based on temporal representations

For acoustic feature extraction we divided each utterance (length is $8\,\mathrm{s}$) into $25\,\mathrm{ms}$ segments with a $10\,\mathrm{ms}$ frame shift. For each frame we extract 13 mel-frequency cepstral coefficients and their deltas and double-deltas obtaining a feature vector of 39 dimensions. We further extract the log pitch ($f0$) and strengths of excitation (5 dim) (Yoshimura et al., 2001). In addition, we also obtain 40 dimensional filter banks and 23 dimensional PLP based features. Filter banks have been obtained using the open source toolkit Kaldi (Povey et al., 2011) with 'dithering' enabled as it was shown to be robust in other experiments. We have also extracted several features using Opensmile toolkit (Eyben et al., 2010) and performed singular value decomposition with the intention of obtaining an acoustic representation. This procedure also results in a dense low dimensional representation. This representation was later used in combination with the high level features we obtained in the spirit of early fusion.

Table 3.3: UAR on Val set. Each model was trained for 100 epochs. BL - Baseline NBL - Normalized Baseline

| Architecture | Features | UAR |
|---|---|---|
| BL (256T + SGD + Batch size 4) | SoundNet | 38 |
| BL + Adam | SoundNet | 36 |
| 5 sec split | SoundNet | 41 |
| BL + 5 sec split | Soundnet | 41 |
| BL + 5 sec split + 0.2 dropout | Soundnet | 42 |
| BL + 5 sec split + Adam | Soundnet | 44 |
| BL + 5 sec split + Adam + 0.2 dropout | Soundnet | 43 |
| BL + 5 sec split + Adam + 0.2 dropout + 16 bsz | Soundnet | 42 |
| BL + 5 sec split + Adam + SI | Soundnet | 44 |
| BL + 5 sec split + Adam + SI + 0.2 dropout | Soundnet | 44 |
| BL + 5 sec split + Adam + SI + selu | Soundnet | 41 |
| BL + 5 and 3 sec split + Adam + SB + selu | Soundnet | 46 |
| BL + 5 and 3 sec split + Adam + SB | Soundnet | 45 |
| BL + 5 and 3 sec split + Adam + 0.2 drop + SB | Soundnet | 44 |
| BL + 5 and 3sec + Adam + 0.2 drop + SB + KNN | Soundnet | 43 |
| BL (1024T +128T + SGD + bsz 4) | AUDEEP | 27 |
| NBL (1024T +128T + SGD + bsz 4) | AUDEEP | 34 |
| NBL (1024T +128T + SGD + bsz 16 + 0.3 drop) | AUDEEP | 36 |
| NBL (1024T +512T + SGD + bsz 16 + 0.3 drop) | AUDEEP | 35 |
| BL (1024T +512T + SGD + bsz 16 + 0.3 drop) | AUDEEP | 34 |
| BL SVM | AUDEEP | 27 |
| BL SVM + Normalized | AUDEEP | 31 |
| BL Linear SVM + Normalized | AUDEEP | 38 |
| Temporal Classification | Low level | 32.2 |
| FALCON | VQVAE Features | **47.2** |

#### 3.5.2.3 Classifier

Using all previously mentioned features, we train a 2 layer bidirectional LSTM network with 512 units in each cell. This is followed by 2 fully connected layers each with 512 units. The final softmax layer dimensions were dependent on the sub challenge. The network is trained by minimizing the expected divergence between the classes using cylindrical SGD (Smith, 2017).

#### 3.5.2.4 Class balancing by data augmentation

We systematically try to increase the data points from the classes with lesser number of examples. Since our utterance level feature extractor realizes different feature representation for audios of different length, we have experimented with augmenting the original data with additional data created by randomly joining audio rom the same class. For this, we have combined all the audio files and split them into longer chunks. We have made 5 second and 3 second chunks in this fashion and augmented their features to the original set.

TABLE 3.4: UAR Blind test summary

| Sub-challenge | Metric |
|---|---|
| Styrian Dialects | 47.25 |
| Baby Sounds | 57.2 |
| Orca | 86.6 |

#### 3.5.2.5 Speaker identity based experiments(System SI)

Since the classifiers we use are discriminative in nature, we experiment with two ways of incorporating speakers or subject specific information:

- (1) We add the identity of the speaker as an extra dimension thus forcing the model to build speaker specific models. For example, in case of decision trees, this forces the model to split at the identity of speaker.
- (2) Normalizing with respect to the speaker, following the procedure typically used in speech recognition.

These systems are referred to as Normalized Baselines.

#### 3.5.2.6 FALCON based VQVAE System

The architecture of our proposed model continues from (Chorowski et al., 2019a), with some modifications. As our encoder, we use the same encoder mentioned in (Chorowski et al., 2019a) that downsamples the input audio by 64. We have used WaveNet(Van Den Oord et al., 2016) as our decoder. Following (Strubell et al., 2017), we have shared the parameters of all the residual layers with common dilation factors. We use Mixture of Logistics loss to train the model and the number of logistics was set to 10. Audio signal was power normalized and squashed to the range (-1,1). To make the training faster, we have used chunks of 4000 time steps. Quantizer acts as a bottleneck and performs a similarity matching to generate the appropriate code from a parameterized learnable codebook. We define the latent space $e \in R^{k \times d}$ contains $k$ $d$-dim continuous vector. The similarity measure is implemented using minimum distance in the embedding space. We have used 128 dimensions to perform the comparison.

#### 3.5.2.7 Evaluation

We have performed an extensive evaluation of all the systems using various hyperparameter settings. The results can be seen in table 3.3.

The evaluation results on blind test set for the three sub-tasks is mentioned in the table 3.4. Based on the preliminary experiments, we have utilized our generative model based approach for evaluating with the blind test set.

## 3.6 Conclusion

In this chapter, I have presented an overview of FALCON, the toolkit I am developing as part of my dissertation. FALCON is built using three core objects - *layers, blocks and models* and allows building systems targeted at solving speech processing based tasks. From the two case studies presented, it can be seen that systems based on FALCON have the potential to (a) outperform the conventional systems in the context of both generative as well as discriminative processes and (b) serve as a decent research toolkit to carry out the experiments in this dissertation.

# Part II

# Scalability

In this part I will provide an overview of experiments within the challenge of scalability. In this dissertation I restrict myself to one phenomena - **Code Mixing** which is related to multilingualism and conversational speech 3.2. I inspect and categorize code mixing from the perspectives of Data, Theory and Applications (Rosenberg, 2018) and present experiments demonstrating how De-Entanglement addresses code mixing.

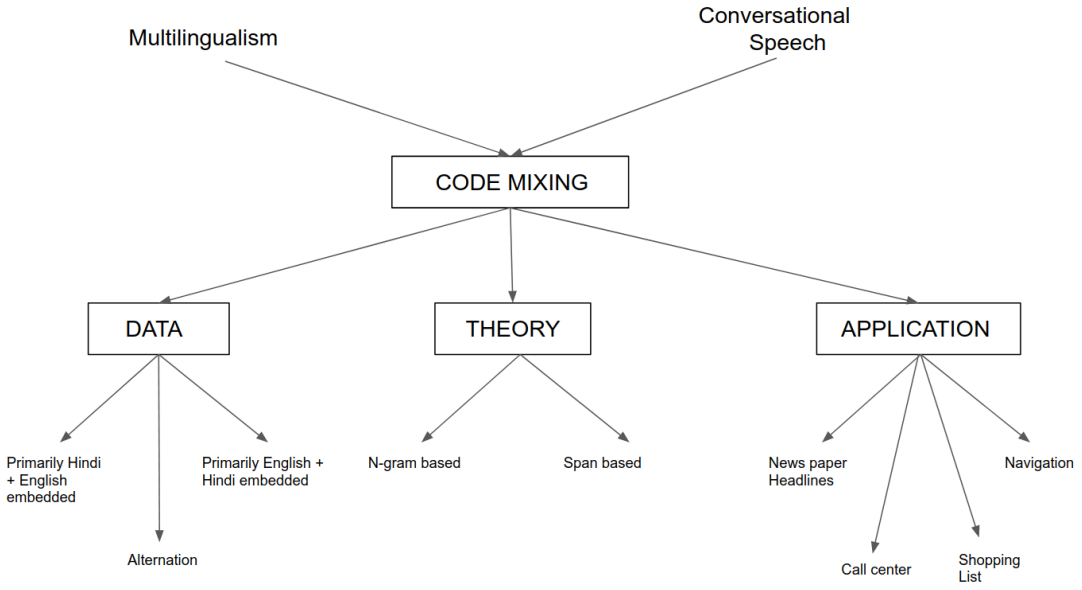



FIGURE 3.2: Taxonomy of Code Mixing from the perspectives of Data, Theory and Applications.

## Code Mixing

Code-switching (or mixing) refers to the phenomenon where bilingual speakers alternate between the languages while speaking. It occurs in multilingual societies such as India, Singapore, etc. and is used both to express opinions as well as for personal and group communications. This can go beyond simple borrowing of words from one language in another and is manifested at lexical, phrasal, grammatical and morphological levels. The technology today - from speech processing systems through conversational agents - assume monolingual mode of operation and do not process code-switched content. However, the mixed content is intuitively the most important part in the content. Since the systems are now handling conversations, it becomes important that they handle code-switching. In terms of speech technologies, this translates to two fundamental requirements:

- Speech Synthesis systems that can synthesize codemixed content.
- Speech Recognition systems that can recognize codemixed content.

In this part, I will present experiments that demonstrate how De-Entanglement provides approaches towards handling both these requirements.

## Speech Synthesis of code mixed content

In this dissertation, I restrict myself to Applications and Data perspectives of code mixing and present approaches towards voice building.

### Applications

While code mixing happens across different scenarios, there are two semi formal scenarios that might make sense to target as first applications:

- (a) News paper headlines where the content is primarily in native language (say, Hindi) with English words interspersed.
- (b) Navigation instructions where the content is primarily in English with named entities in the native language.
- (c) Shopping Lists where the shopping entry is typically in the native language.
- (d) Call center conversations which are characterized by a near fluid transition between the participating languages.

In this dissertation, I present experiments handling scenarios (a) and (b).

### Availability of Data

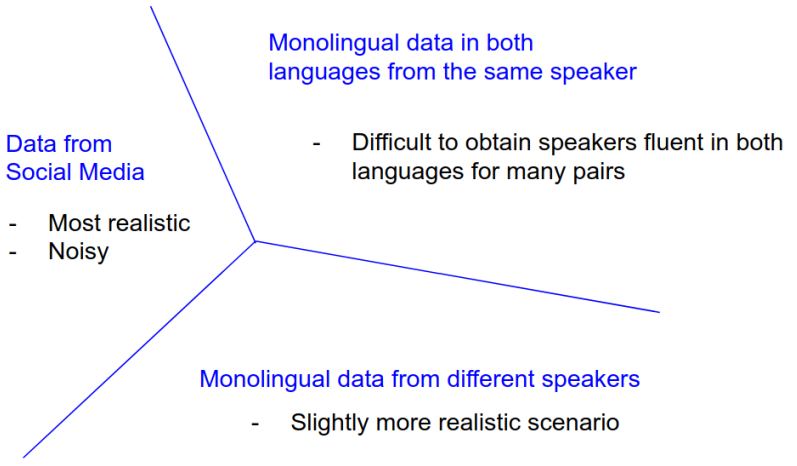



FIGURE 3.3: Figure illustrating code mixing scenarios depending on the availability of data

Figure 3.3 depicts categorization of code mixing from the perspective of available data. Speech synthesis typically uses clean recordings from a speaker in a controlled settings. Given that code mixing happens in social scenarios, it is difficult to get speaker data.

- (a) When we have data only from one language.
- (b) When we have data from both the languages but monolingual in the language - For instance, the voice talent records data first in Hindi and later in English.
- When we have data that is truly mixed - YouTube videos with interviews of contemporary stars.

## Speech Recognition of code mixed content

I present experiments that demonstrate how De-Entanglement of style information can help improve the performance of speech recognition systems targeted at code mixed speech.

## Road map to Part 1

Real world data sources are characterized by presence of background noise. It is important to extract the content information from such data in order to build reliable speech synthesis systems. In chapter 5, I present experiments in this direction demonstrating how De-Entanglement. Specifically, I employ source separation as the task and show that content can be de-entangled from noisy speech signals. In chapter 4, I present experiments on de-entangling style information from code mixed speech utterances. I demonstrate using experiments in language modeling that this approach improves the perplexity of code mixed data. I then present experiments in voice building by jointly de-entangling content and style information from the speech utterances.

# 4

# SCALABILITY - De-Entanglement of Style: A Case Study with Code Switching Style Detection in Conversational Speech

In this chapter, I present a case study about de-entanglement of style information from speech utterance. Multilingual speakers switch between languages displaying inter sentential, intra sentential, and congruent lexicalization based transitions. While monolingual ASR systems may be capable of recognizing a few words from a foreign language, they are usually not robust enough to handle these varied styles of code-switching. There is also a lack of large code-switched speech corpora capturing all these styles making it difficult to build code-switched speech recognition systems. I first look at the first problem of detecting code-switching style from acoustics. We classify code-switched Spanish-English and Hindi-English corpora using two metrics and show that features extracted from acoustics alone can distinguish between different kinds of code-switching in these language pairs.

## 4.1 Introduction

Code-switching refers to the phenomenon where bilingual speakers alternate between the languages while speaking. It occurs in multilingual societies around the world. As Automatic Speech Recognition (ASR) systems are now recognizing conversational speech, it becomes important that they handle code-switching. Furthermore, code-switching affects co-articulation and context dependent acoustic modeling (Elias et al., 2017). Therefore, developing systems for

such speech requires careful handling of unexpected language switches that may occur in a single utterance. We hypothesize that in such scenarios it would be desirable to condition the recognition systems on the type (Muysken, 2000) or style of language mixing that might be expected in the signal. In this work, we present approaches to detecting code-switching 'style' from acoustics. We first define style of an utterance based on two metrics that indicate the level of mixing in the utterance: CodeMixing Index(CMI) and CodeMixing Span Index. Based on these, we classify each mixed utterance into 5 style classes. We also obtain an utterance level acoustic representation for each of the utterances using a variant of SoundNet. Using this acoustic representation as features, we try to predict the style of utterance.

## 4.2 Related Work

Prior work on building Acoustic and Language Models for ASR systems for code-switched speech can be categorized into the following approaches: (1) Detecting code-switching points in an utterance, followed by the application of monolingual acoustic and language models to the individual segments (Chan et al., 2004; Lyu and Lyu, 2008; Shia et al., 2004). (2)Employing a shared phone set to build acoustic models for mixed speech with standard language models trained on code-switched text (Imseng et al., 2011; Li et al., 2011; Bhuvanagiri and Kopparapu, 2010; Yeh et al., 2010). (3) Training Acoustic or Language models on monolingual data in both languages with little or no code-switched data (Lyu et al., 2006; Vu et al., 2012; Bhuvanagirir and Kopparapu, 2012; Yeh and Lee, 2015). We attempt to approach this problem by first identifying the style of code mixing from acoustics. This is similar to the problem of language identification from acoustics, which is typically done over the span of an entire utterance.

Deep Learning based methods have recently proven very effective in speaker and language recognition tasks. Prior work in Deep Neural Networks (DNN) based language recognition can be grouped into two categories: (1) Approaches that use DNNs as feature extractors followed by separate classifiers to predict the identity of the language (Jiang et al., 2014; Matejka et al., 2014; Song et al., 2013) and (2) Approaches that employ DNNs to directly predict the language ID (Richardson et al., 2015b,a; Lopez-Moreno et al., 2014). Although DNN based systems outperform the iVector based approaches, the output decision is dependent on the outcome from every frame. This limits the real time deployment capabilities for such systems. Moreover, such systems typically use a fixed contextual window which spans hundreds of milliseconds of speech while the language effects in a code-switched scenario are suprasegmental and typically span a longer range. In addition, the accuracies of such systems, especially ones that employ some variant of iVectors drop as the duration of the utterance is reduced. We follow the approach of using DNNs as utterance level feature extractors. Our interest is in adding long term information to influence the recognition model, particularly at the level of the complete utterance, representing stylistic aspects of the degree and style of code-switching throughout the utterances.

| Class | CMI | Hi-En Utts | En-Es Utts |
|---|---|---|---|
| C1 | 0 | 6771 | 41624 |
| C2 | 0-0.15 | 13986 | 2284 |
| C3 | 0.15-0.30 | 492 | 2453 |
| C4 | 0.30-0.45 | 8865 | 1025 |
| C5 | 0.45-1 | 2496 | 1562 |

TABLE 4.1: Distribution of CMI classes for Hinglish and Spanglish

| Class | Description | Hi-En | En-Es |
|---|---|---|---|
| S1 | Mono En | 5413 | 27960 |
| S2 | Mono Hi/Es | 0 | 12749 |
| S3 | En Matrix | 626 | 2883 |
| S4 | Hi/Es Matrix | 36454 | 1986 |
| S5 | Others | 8307 | 3345 |

TABLE 4.2: Distribution of span based classes for Hinglish and Spanglish. Note that the term 'Matrix' is used just here notionally to indicate larger word span of the language.

## 4.3 Style of Mixing and Motivation

Multiple metrics have been proposed to quantify codemixing (Guzmán et al., 2017; Gambäck and Das, 2014) such as span of the participating languages, burstiness and complexity. For our current study, we categorize the utterances into different styles based on two metrics: (1) Code Mixing index (Gamback and Das, 2014) which attempts to quantify the codemixing based on the word counts and (2) CodeMixed Span information which attempts to quantify codemixing of an utterance based on the span of participating languages.

### 4.3.1 Categorization based on Code Mixing Index

Code Mixing Index (Gamback and Das, 2014) was introduced to quantify the level of mixing between the participating languages in a codemixed utterance. CMI can be calculated at the corpus and utterance level. We use utterance CMI, which is defined as:

$$C_u(x) = 100\frac{w_m(N(x) - \max_{L_i \in L}\{t_{L_i}\}(x)) + w_p P(x)}{N(x)} \tag{4.1}$$

where $N$ is the number of languages, $t_{L_i}$ are the tokens in language $L_i$, $P$ is the number of code alternation points in utterance $x$ and $w_m$ and $w_p$ are weights. In our current study, we quantize the range of codemixed index ( 0 to 1) into 5 styles and categorize each utterance as shown in Table 4.1. A CMI of 0 indicates that the utterance is monolingual. We experimented with various CMI ranges and found that the chosen ranges led to a reasonable distribution within the corpus.

### 4.3.2 Categorization based on Span of codemixing

While CMI captures the level of mixing, it does not take to account the span information (regularity) of mixing. Therefore, we use language span information (Guzmán et al., 2017) to categorize the utterances into 5 different styles as shown in Table 4.2. We divide each utterance based on the span of the participating languages into five classes - monolingual English, monolingual Hindi or Spanish, classes where the two languages are dominant (70% or more) and all other utterances. The classes S3 and S4 indicate that the primary language in the utterance has a span of at least 70% with respect to the length of utterance. This criterion makes these classes notionally similar to the construct of 'matrix' language. However, we do not consider any information related to the word identity in this approach. As we can see from both the CMI and span-based classes, the distributions of the two language pairs are very different. The Spanglish data contains much more monolingual data, while the Hinglish data is predominantly Hindi matrix with English embeddings. The Hinglish data set does not have monolingual Hindi utterances which is due to the way the data was selected, as explained in Section 4.4.1.

### 4.3.3 Style Modeling using Modified SoundNet

SoundNet (Aytar et al., 2016) is a deep convolutional network that takes raw waveforms as input and is trained to predict objects and scenes in video streams. Once the network is trained, the activations of intermediate layers can be considered as a high level representation which can be used for other tasks. However, SoundNet is a fully convolutional network, in which the frame rate decreases with each layer. Each convolutional layer doubles the number of feature maps and halves the frame rate. The network is trained to minimize the KL divergence from the ground truth distributions to the predicted distributions. The higher layers in SoundNet are subsampled too much to be used directly for feature extraction. To alleviate this, we train a fully connected variant of Soundnet (Wang and Metze, 2017): Instead of using convolutional layers all the way up, we switch to fully connected layers after the 5th layer. We have also change the input sampling rate to 16 KHz to match the rate of provided data.

## 4.4 Experimental Setup

### 4.4.1 Data

We use code-switched Spanish English (referred to as Spanglish hereafter) released as a part of Miami Corpus (Deuchar et al., 2014) for training and testing. The corpus consists of 56 audio recordings and their corresponding transcripts of informal conversation between two or more speakers, involving a total of 84 speakers. We segment the files based on the transcriptions provided and obtain a total of 51993 utterances. For Hinglish, we use an in-house speech corpus of conversational speech. Participants were given a topic and asked to have a conversation in

Hindi with another speaker. 40% of the data had at least one English word in it, which was transcribed in English, while the Hindi portion of the data was transcribed in Devanagari script. We split the data into Hindi and Hinglish by filtering for English words, hence the Hinglish data does not contain monolingual Hindi utterances. Note that this data did contain a few monolingual English sentences, but they were typically single word sentences. Such English utterances were considered to be part of the Hinglish class. The number of Hinglish utterances is 54279.

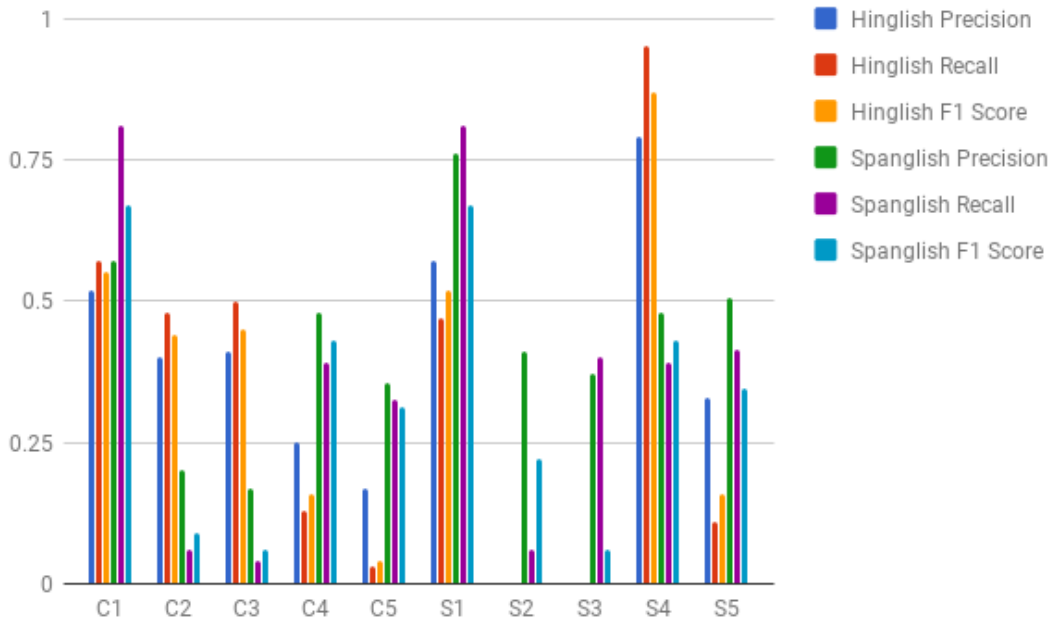



FIGURE 4.1: Precision, Recall and F1 scores for 5 way style classification of Hinglish and Spanglish

### 4.4.2 Style Identification

For style identification we perform the following procedure: We first categorize the utterances into 5 styles based on the criteria described in section 4.3. We pass each utterance through pretrained modified SoundNet and obtain the representations at all the layers. We use the representation from 7th (penultimate) layer as embedding for the utterance. We experimented with combining the representations at multiple layers but found that they do not outperform the representation at the 7th layer alone. Therefore for the purposes of this work, we restrict ourselves to the representation at the penultimate layer. The embedding is obtained by performing mean pooling on the representation. Finally, we train a Random Forest classifier using the obtained embedding to predict the style of mixing.

### 4.4.3 Results and Discussion

Figure 4.1 shows the results for 5 class classification for Hinglish and Spanglish based on CMI (classes C1-C5) and span (classes S1-S5). Some classes (C1, C2, C3, S1, S4 for Hi-En and C1, C4, C5, S1, S4, S5 for En-Es) are easier to predict and are not always the majority classes. In our current implementation, we use a two stage approach for feature extraction and classification. We hypothesize that there might be better approaches to perform each of the components independently. It might also be possible to incorporate a style discovery module in an end to end fashion (Wang et al., 2018b). As we plan to include the predicted style information in our recognition system, we also evaluate our approach using language models. For this, we build style specific language models tested on style specific test sets and include the average perplexity values for all of them in table 4.3. Ground Truth indicates that the model was built on the classes segregated based on approaches described in section 4.3. Predicted indicates that the language model was built based on the classes predicted by the model described in section 4.4.2. We also build a language model on utterances from the majority class for CMI and Span, as well as all the Spanglish data with no style information. As can be observed, the perplexity has a considerable reduction when using style specific information, while the majority style does not lead to the same reduction over the model with no style information. This further validates our hypothesis that style specific models may help decrease LM perplexities and ASR error rates.

Table 4.3: Language Model Experiments

| Language | | | Avg Ppl |
|---|---|---|---|
| Spanglish | CMI | GroundTruth | 54.8 |
| | | Predicted | 56.2 |
| | | Majority Class | 81.2 |
| | Span | GroundTruth | 59.1 |
| | | Predicted | 62.8 |
| | | Majority Class | 80.2 |
| No Style Info | | | 82.1 |

## 4.5 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of style information from speech utterance. Specifically, I present a preliminary attempt at categorizing code-switching style from acoustics, that can be used as a first pass by a speech recognition system. Language Model experiments indicate promising results with considerable reduction in

perplexity for style-specific models. In future work, we plan to improve our feature extraction and classification models and test our language models on code-switched speech recognition.

# 5

# SCALABILITY - De-Entanglement of Content: A Case Study with Blind Source Separation

In order to build language technologies for majority of the languages, it is important to leverage the resources available in public domain on the internet - commonly referred to as 'Found data'. However, such data is characterized by the presence of non-standard, non-trivial variations. For instance, speech resources found on the internet have non-speech content, such as music. Therefore, speech recognition and speech synthesis models need to be robust to such variations. In this work, we present an analysis to demonstrate that it is possible to extract clean content in the original data by employing a priors based approach. Specifically, we present a method to split the latent prior space into continuous representations of dominant speech modes present in the magnitude spectra of audio signals. We propose a completely unsupervised approach using multinode latent space variational autoencoders (VAE). We show that the constraints on the latent space of a VAE can be in-fact used to separate speech and music, independent of the language of the speech. This chapter also analytically presents the requirement on the number of latent variables for the task based on distribution of the speech data.

## 5.1 Background

Speech synthesis has taken some major strides in past few years especially in the form of Text to Speech synthesis (TTS) models. However, most of the work that has been carried out involves

carefully recorded speech data. Generation of such vast amount of data for every application is a daunting task. On the other hand, there is a plethora of speech data that is available on the internet such as news broadcasts, press conferences, audio books etc - also referred to as *Found data*. The only hindrance in utilizing such data for speech based machine learning models is that this Found data is characterized by noise or music in the background. Presence of noise / music degrades the performance of such models. One of the solutions to this problem is source separation - separating out speech from music in the audio. There have been several attempts to accomplish this task using both classical speech processing techniques as well as deep learning models. (Huang et al., 2012) proposed a matrix factorization of the magnitude spectrogram of audio that utilizes the periodicity in music and sparseness in speech to separate the two. However, this technique requires a lot of hyperparameter tuning depending on the type of background music and also degrades the quality of separated speech to some extent. REPET (Rafii and Pardo, 2013) also involves music separation by exploiting its periodic nature but on occasions still leaves a residual music in the background. Most of the work in source separation using deep learning has been supervised (Fan et al., 2017), (Grais et al., 2016), (Soni et al., 2018), (Valentini Botinhao et al., 2016), i.e. they had both noisy and clean versions of the data. However most of the times, especially with Found data, we don't have the clean version of the data.

There has also been some focus on source separation using unsupervised models. (Hsu et al., 2018b) takes the approach of data augmentation by adding different background noise to the clean data and then training an adversarial classifier to make these augmented versions of data indistinguishable from the original speech. However, this method again requires a clean version of data first and additional data augmentation that is representative of the noise in the background. Therefore essentially, this is a semi-supervised approach that requires labels for clean and noisy data. One other semi-supervised approach is using domain adaptation (Ganin et al., 2016) where output is made to follow the clean data domain while making the encoding for clean and noisy data domain indistinguishable using a adversarial classifier. However, this approach requires speech content in both clean and noisy version of data to be very similar for domain adaptation to occur.

We propose a completely unsupervised approach using multinode variational autoencoders (VAE) combined with robust principal component analysis (RPCA) (Huang et al., 2012) as a post-processing step. Our goal is to enable the use of Found data for downstream TTS applications. Therefore, the data we target is predominantly speech with music in the background. We apply this approach on two datasets:- Wilderness(Black, 2019) and Hub4. Wilderness consists of Bible recordings in 699 languages while Hub4 is a news broadcast dataset in English. Both of these datasets contain music/noise in the background. We show that the proposed approach separates out the dominant mode, speech, from a noisy audio and improves the performance of the downstream tasks irrespective of the language of the speech.

This chapter is organized as follows: section 1 discusses the variational autoencoder framework, section 2 talks about source separation using VAE, section 3 addresses the extension to multinode VAE architeture and section 4 discusses post-processing using robust principal component analysis (RPCA). Section 5 analyses the source separation capacity and architectural requirements of the proposed model. Section 6 reports the performance of the proposed model for source separation and for downstream TTS applications. We conclude in section 7.

## 5.2 Variational Autoencoder

Variational autoencoder model in this chapter follows the standard formulation consisting of an inference network with a speech encoder $p(z|x)$ and a latent space decoder $p(x|z)$, where $x$ and $z$ represent the input and the latent space random variables respectively.

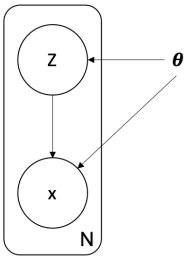



FIGURE 5.1: Latent Variable Model - Variational Autoencoder

The figure 5.1 depicts the latent variable model for variational autoencoder. The true posterior density is intractable.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{5.1}$$

We then approximate the true posterior $p(z|x)$ with a variational distribution $q(z|x)$ that has a prior $p(z)$. The objective can be represented by the evidence lower bound (ELBO) or variational lower bound on the likelihood of the data.

$$\log p(x) \geq \mathcal{L}(x) \tag{5.2}$$

$$\mathcal{L}(x) = \mathbb{E}_q[\log(p(x|z))] - D_{KL}(q(z|x)||p(z)) \tag{5.3}$$

where $\mathcal{L}(x)$ denotes the variational lower bound on the likelihood of the data and $D_{KL}$ is the Kullback-Leibler divergence. We write the first term as a mean squared error (MSE) between the reconstructed and the original data and the prior p(z) follows a standard normal distribution $\mathcal{N}(0, I)$.

## 5.3 VAE for Source Separation

As shown in the previous section, a variational autoencoder reconstructs the input data conditioned on the latent space. The latent space is constrained to follow a certain prior distribution, such as Gaussian distribution. (Dai et al., 2018) shows that this formulation is equivalent to minimizing the alternative lower bound function

$$\begin{aligned} minimize \;\; & n \cdot rank[\, L \,] + \parallel S \parallel_0 \\ & M = L + S \end{aligned} \tag{5.4}$$

where M is the original data matrix, L is a low-rank matrix and S is a sparse matrix and $\parallel . \parallel_0$ denotes the $l_0$ norm. This is shown to be equivalent to an RPCA problem if an optimum solution exists otherwise it's known to smooth out undesirable erratic peaks from the energy curve. (Dai et al., 2018) also presents some interesting results on VAE and it's separation properties. We rewrite some of the results what it means in our context here.

- This formulation of variational autoencoders is shown to perform robust outlier removal in the context of learning inlier points constrained to a manifold of unknown dimension. In simple terms this means, VAE has the property to remove sparse components in the input data distribution and accordingly reduces the latent space to a required (unknown) dimension.
- VAE also help smooth out undesirable minima from the energy landscape of the optimization problem which differentiates it from traditional deterministic autoencoders.

Since our goal is to enhance speech synthesis and speech recognition performances on the 'found' data, we target audio data that is predominantly speech with some music (almost uniform) in the background, for instance, news broadcasts and audio books. We will later show that the presence of the background music can effect the speech synthesis performance drastically. It's been shown that speech and music distributions in audio are quite distinct (RJ and SR, 2012) (RV, 2005). As a result VAE has the tendency to remove the sparse outlier - music from the audio.

In case of multiple speakers in the input audio there can be multiple modes in the speech distribution as well. This can be solved by have multiple nodes in the latent space. This is possible because all latent variables are initialized at random and pick multiple speech modes from the distribution. Later in the results section, we are going to analyze this and the requirement on the number of nodes in the latent space depending on the input data distribution. We also talk about how the performance of the output speech changes based on the intensity/loudness of the music in the background.

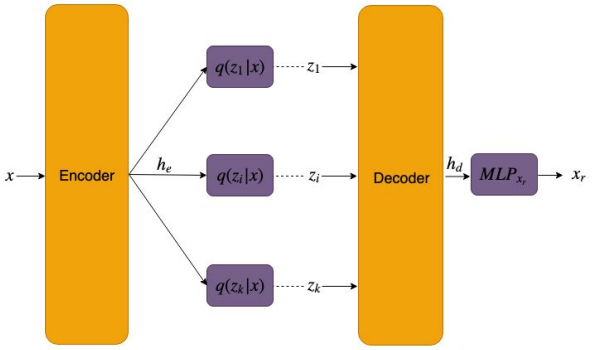



FIGURE 5.2: Multi-node VAE model. Dashed lines represent sampling using reparametrization. Encoder and Decoder are Bi-LSTM networks. Purple blocks are fully connected layers.

## 5.4 Multi-node VAE Model Architecture

The Figure 5.2 depicts the multi-node variational autoencoder architecture. It consists of an Bi-LSTM encoder ($LSTM_E$) for the inference network that captures latent space distributions $p(z_1|x), p(z_2|x), \ldots, p(z_k|x)$ where $x$ is the magnitude spectrogram of the input audio, $z_1, z_2, \ldots, z_k$ are the latent variables and $k$ is the number of latent variables. The reconstruction network is a Bi-LSTM decoder ($LSTM_D$) which generates the reconstructed input distribution at each time-step $p(x_r^t|x_r^{t-1}, z_1, z_2, \ldots, z_k)$ conditioned on the reconstructed input from the previous time-step and the latent space.

$$h_e^t, c_e^t = LSTM_E(x^t, h_e^{t-1}, c_e^{t-1}) \tag{5.5}$$

$$\mu_i^t = MLP_{\mu_i}(h_e^t)\ \forall i \in 0, \ldots, k \tag{5.6}$$

$$logvar_i^t = MLP_{\sigma_i}(h_e^t)\ \forall i \in 0, \ldots, k \tag{5.7}$$

$$q(z_i|x) = \mathcal{N}(\mu_i, \exp(logvar_i)) \tag{5.8}$$

$$h_d^t, c_d^t = LSTM_D(\phi^t, z_1^t, \ldots, z_k^t, h_d^{t-1}, c_d^{t-1}) \tag{5.9}$$

$$\phi^t = MLP_\phi(h_d^{t-1}) \tag{5.10}$$

$$x_r^t = MLP_{x_r}(h_d^t) \tag{5.11}$$

where Bi-LSTM refers to bidirectional long short term memory recurrent neural network, MLP refers to a multi-layer perceptron network, $h_e, c_e$ represent hidden and cell states of the encoder LSTM, $h_d, c_d$ represent hidden and cell states of the decoder LSTM and $\phi$ represents the context from the previous time-step of the decoder. The initial hidden state and the cell state of the decoder LSTM are learnable parameters. The latent variable model for the multinode

variational autoencoder is shown in figure 5.3

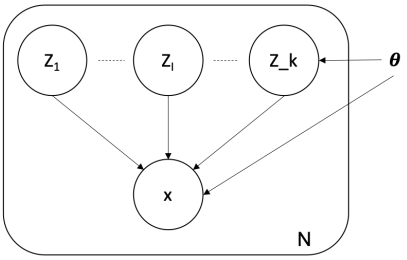



FIGURE 5.3: Latent Variable Model - Multinode Variational Autoencoder

The modified learning objective for a multinode VAE can be represented as an extension of equation 5.3 as:

$$\begin{aligned}\mathcal{L}(x) \geq \mathbb{E}_q[\log(p(x|z_1, z_2, \ldots, z_k))] \\ - \sum_{i=1}^{k} D_{KL}(q(z_i|x)||p(z_i))\end{aligned} \tag{5.12}$$

## 5.5 Speech Enhancement

The output of the VAE network from the above formulation removes the music from the audio however, replaces the music content with random noise instead of silence. There can be multiple post processing or speech enhancement techniques used to eliminate this residual noise such as speech enhancement neural networks or classical speech processing methods. Here in this chapter we use robust principal component analysis (RPCA) (Huang et al., 2012) to eliminate the background noise as it gives control over the quality of speech versus the amount of background noise. We follow the original formulation from the paper by expressing the speech separation as a matrix factorization problem. It represents the magnitude spectrogram of the audio signal as a sum of low rank matrix and a sparse matrix. The assumption here is that non-speech component (background noise) is low rank while the speech component is sparse.

$$\begin{aligned}minimize \parallel L \parallel_* + \lambda \parallel S \parallel_1 \\ M = L + S\end{aligned} \tag{5.13}$$

where $M \in \mathbb{R}^{n_1 \times n_2}$ is the magnitude spectrogram of the VAE output, $L \in \mathbb{R}^{n_1 \times n_2}$ is a low rank matrix, $S \in \mathbb{R}^{n_1 \times n_2}$ is a sparse matrix, $\parallel . \parallel_*$ is the nuclear norm and $\parallel . \parallel_1$ is the $L_1$ norm. $\lambda > 0$ is a hyperparameter that controls the rank and sparsity of $L$ and $S$ respectively. It is recommended in (Huang et al., 2012) to use $\lambda = 1/\sqrt{max(n_1, n_2)}$ to obtain the best result. However, we only need to enhance the audio a little while retaining the speech quality so we use $\lambda = 0.3/\sqrt{max(n_1, n_2)}$. Instead of the hard mask in (Huang et al., 2012) we used

a soft mask as it resulted in a better quality and a smoother speech. The idea is to have a high value for the speech mask where the magnitude of the speech component is much greater the magnitude of the non-speech component.

$$\begin{aligned} |S| &> g|L| \\ |S|^2 &> g^2|L|^2 \\ |M|^2 &= |S|^2 + |L|^2 \\ |S|^2 &> g^2|M|^2 - g^2|S|^2 \\ |S|^2 &> \frac{g^2}{1+g^2}|M|^2 \\ |S| &> \sqrt{\frac{g^2}{1+g^2}}|M| \\ \frac{|S|}{|M|} &- \sqrt{\frac{g^2}{1+g^2}} > 0 \end{aligned} \tag{5.14}$$

where $g \geq 0$ is the gain factor. We came up with a Sigmoid looking threshold for the mask which is still close to the hard mask but results in smoother speech transitions.

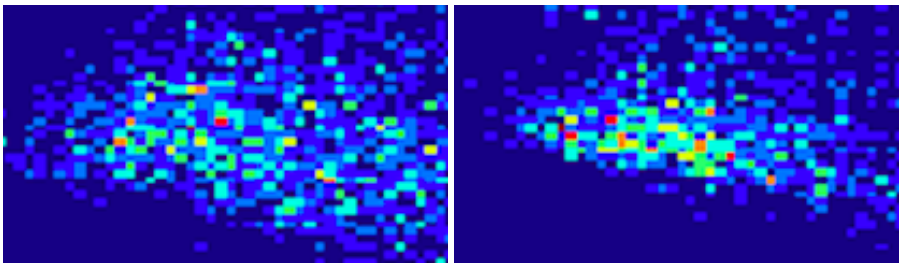
$$W = \frac{1}{1 + \exp\left(-\alpha(\frac{|S|}{|M|} - \sqrt{\frac{g^2}{1+g^2}})\right)} \tag{5.15}$$

where $W \in \mathbb{R}^{n_1 \times n_2}$ represents the speech mask and the obtained speech spectrogram is

$$X_{speech}(i,j) = W(i,j)M(i,j)\ \forall i,j \tag{5.16}$$

## 5.6 Experiments

We applied the multinode VAE model on two datasets:- Wilderness and Hub4. Wilderness dataset consists of Bible recordings in 699 languages with music in the background. We carried out full experiments on two languages:- Dhopadhola (an African language) and Marathi (an Indian language). The results presented here are based on the model that was trained on languages different than the ones that are reported/tested. Hub4 consists of news broadcast recordings in English with various forms of noise in the background, such as music, clapping, roaring etc. We used about 2 hrs of training data for both datasets consisting 1 hr of speech only data and 1 hr of speech-music data.



(a) (b)

FIGURE 5.4: Input Data Distributions for (a) Wilderness (b) Hub4. The red dots show the high density regions in each distribution.

For these experiments, the VAE model consists of input magnitude spectrogram of dimension 512, Bi-LSTM encoder and decoder with hidden size of 512, each of the fully connected layers for latent variables and decoder context from the previous time-step of dimension 64 and the final output layer with a dimension same as the input dimension. We trained for 50 epochs with annealing weight for KL-Divergence loss, this is explained in detail below. For both datasets, we used an ADAM optimizer with a learning rate of 1e-3.

The input data distributions for the two dataset are shown in Figure 5.4. These 2-dimensional distributions are obtained after applying PCA to the magnitude spectrogram and plotting the histogram of the first two components. This figure shows the high density regions of the two distributions. As we can observe, the wilderness distribution has one significant high density region while the hub4 distribution consists of multiple high density regions. Hence, Hub4 data will have more dominant speech modes than the wilderness data. This is probably because there are multiple speakers in Hub4 as well as news broadcast speech has more variance as compared to bible recordings. This gives us an approximate idea that Hub4 multinode VAE model will require more nodes in the latent space than the model for Wilderness dataset.

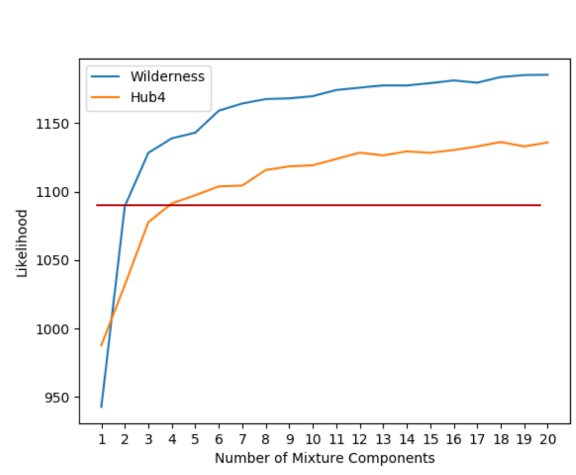
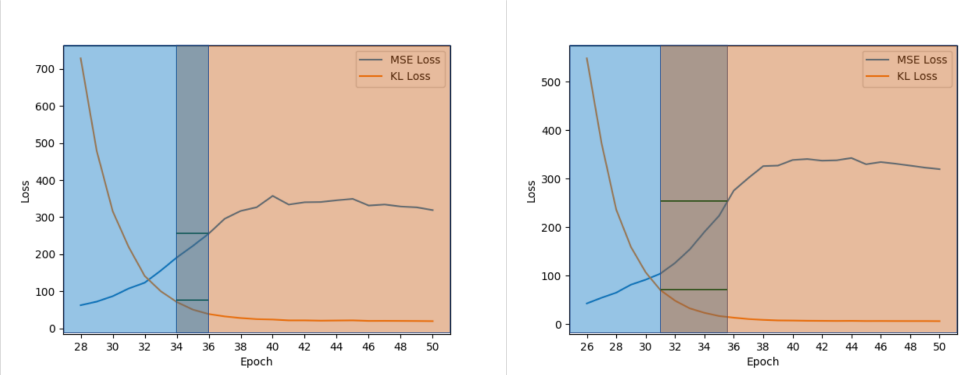



FIGURE 5.5: Gaussian Mixture Fit for Wilderness and Hub4




(a) (b)

FIGURE 5.6: Training Loss (a) 1-Node VAE: Wilderness (b) 3-Node VAE: Wilderness. The left shaded blue region and the right shaded orange region show the required MSE loss and KL loss threshold respectively to obtain good speech. Overlap region represents the model parameters where speech separation occurs.

Figure 5.5 shows the likelihood of fitting Gaussian Mixture Models as a function of the number of cluster centers. As discussed earlier, speech modes are dominant in the target data so fitting $n$ clusters in the curve can be thought of as having $n-1$ nodes/clusters in the VAE for the speech and 1 cluster as the residual non-speech data. The multinode VAE model for the wilderness data obtained good results with just 1 VAE node or 2 clusters as can be confirmed from the graph where likelihood values are high for just 2 clusters. On the other hand, multinode VAE model for Hub4 gave good results with 3 nodes or 4 clusters. Now, as we increase the number of nodes, the peak performance does not change much, however, we attain the same peak performance for more model states:- MSE loss vs KL loss. This will be explained using training loss curves. It would have been better to use some validation parameter but since model performance for human hearing can only be analyzed by listening to the speech, we use the training metrics.

The Figure 5.6 and 5.10 give an idea of speech separation capacity of the model as we increase the number of latent variables. During training of the multinode VAE model, we anneal the KL divergence loss for latent space exponentially. We do the annealing for latent variables simultaneously. Initially, KL divergence loss is assigned a very small weight and then increased exponentially. So, it first increases (not shown in the plot as it's out of the range of the plot) and then decreases eventually while the MSE reconstruction loss first decreases and then increases slightly. During this process there is a small window where both the losses are low enough and we are able to extract out speech from the audio. This window is determined by the threshold values for both losses. If both loss values are below their respective thresholds, we observe speech at the output.

The blue and orange shaded regions in figures 5.6 and 5.10 depict the loss values below the

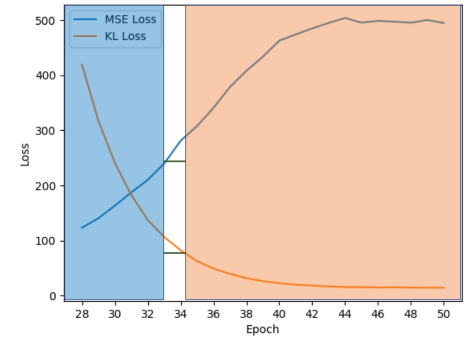



FIGURE 5.7: 1-Node VAE: Hub4

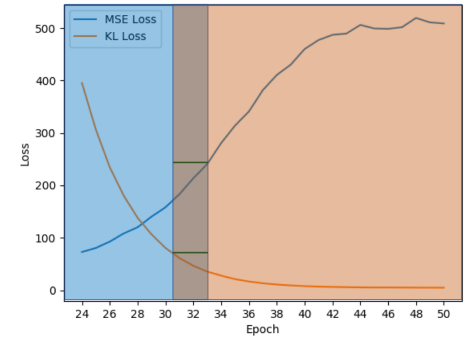



FIGURE 5.8: 3-Node VAE: Hub4

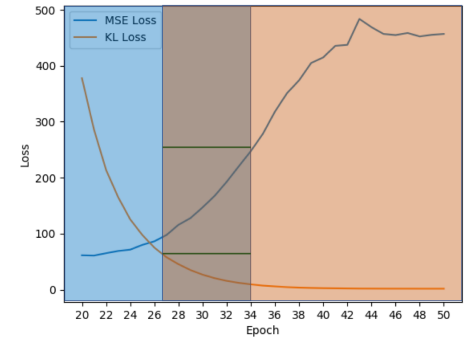



FIGURE 5.9: 8-Node VAE: Hub4

FIGURE 5.10: Training Loss (a) 1-Node VAE: Hub4 (b) 3-Node VAE: Hub4 (c) 8-Node VAE: Hub4. The left shaded blue region and the right shaded orange region show the required MSE loss and KL loss threshold respectively to obtain good speech. Overlap region represents the model parameters where speech separation occurs.

threshold for MSE loss and KL divergence loss respectively. Therefore, their intersection as indicated by the overlap region represents the model parameters that result in speech and music separation. For visualization, the MSE loss in the figure is averaged over all the samples as well as in the time dimension of the audio while the KL divergence loss is averaged over all the latent variables as well as across all samples. Using these loss definitions results in a threshold

value of 250 for MSE loss and a threshold value of 60 for KL divergence loss for both datasets. These are just soft experimental values and may change for other datasets as well as a different loss definition. The key idea is that there exists a window where speech separation occurs.

As shown in the figure, for Wilderness data we obtain this window with just one node in the latent space. As we increase the number of nodes in the latent space, we don't see any significant improvement in the quality of the output speech, however, we do obtain a wider window where this separation occurs. As for the Hub4 dataset, we don't observe any such window with one latent node, however, we do obtain a separation window with three latent nodes and an even wider window with eight latent nodes. Observe that, these results align with the results derived using input data distribution and GMM fitting analysis. Therefore, to be totally certain about the existence of a separation window, we can always add a few more latent variables than what we obtain from our analysis of input distribution.

## 5.7 Observations

Let's explore the nature of the output on either side of the separation window. On the blue/left side of the window MSE loss is very low while the KL divergence loss is high, this results in the output that is close to the original input that consists of both speech and music. On the orange/right side of the window, MSE loss is high while the KL divergence loss is low. This causes the network output to be a really noisy version of the speech component of the audio. We also observed that, as the intensity/loudness of music in the background increases in an audio or for a part of the audio, the speech separation performance for that part of the audio begins to deteriorate slightly. For example, in case of advertisement segments between news broadcasts where music tends to dominate the segment. In such cases, we can still hear some traces of music in the background when we use the same VAE model as we do for the rest of the data. However, this is not a concern as the downstream applications we target don't generally depend on data with such high intensity music. As mentioned in (Dai et al., 2018), a traditional autoencoder is not able to perform the outlier removal. We verified this fact experimentally. We removed any constraint on the latent space and trained the model to minimize the reconstruction loss. We observed that the model was able to reconstruct the audio completely including both speech and music. Therefore, a traditional autoencoder is not able to smooth out the energy contour and fails to remove any outliers. We also tried to experiment with this model on songs and movie clips. As the background music in songs and movies very dense and varies significantly, we observed that our model wasn't able to separate out speech completely. The output speech contained some music in the background and the quality of speech itself was compromised.

## 5.8 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of style information from speech utterance. For this, I employ source separation as the example task. Specifically, we show that Multinode VAE model helps to remove the background noise/music in the 'Found data' irrespective of the language of the speech. Extensive studies on different type of speech and music data verify the effectiveness and robustness of the proposed approach. Performance of this model on Text-to-Speech synthesis applications shows the potential of such an approach that can be further extended to other speech based machine learning models such as Automatic Speech Recognition (ASR). Such an efficient source separation technique can help overcome a major cause for under utilization of 'Found data'. This could mean that acoustic based machine learning models can be drastically improved by leveraging the data found on the internet. Since this approach works in a unsupervised fashion, it eliminates the need to obtain labeled data which has been major hindrance to effective utilization of 'Found data'. Since 'Found data' is abundant, this could also possibly further accelerate the research in this area.

# 6

# SCALABILITY - De-Entanglement of Content and Style: Building code mixed voices using bilingual data

In this chapter, I present an approach to jointly de-entangling both content and style information building on the observations made from chapters 4 and 5. For this, I will employ Text to Speech(TTS) as an example task. TTS systems today are not fully capable of effectively handling such mixed content despite achieving high quality in the monolingual case. I present various mechanisms for building mixed lingual systems built using a mixture of monolingual corpora and are capable of synthesizing such content. First, I explore the possibility of manipulating the phoneme representation: using target word to source phone mapping with the aim of emulating the native speaker intuition. I then present experiments at the acoustic stage investigating training techniques at both spectral and prosodic levels. Subjective evaluation shows that our systems are capable of generating high quality synthesis in code mixed scenarios.

## 6.1 Introduction

Data driven statistical parametric speech synthesis systems have displayed a continued improvement over the recent past, in terms of speech quality (King, 2016; Prahallad et al., 2014). These improvements can be attributed to developments in the aspects such as speech parameterization (Kawahara, 2006; Morise et al., 2016; Soong and Juang, 1984), modeling(Black

and Muthukumar, 2015; Wu et al., 2015b), post filtering(Takamichi et al., 2014; Muthukumar and Black, 2016; Chen et al., 2015a) and have led to deployment in consumer grade systems(Wilkinson et al.). Currently, such Text to Speech (TTS) systems assume that the input is in a single language and that it is written in native script. However, due to the rise in globalization, phenomenon such as code mixing / code switching are now seen in various types of text ranging from news articles through comments/posts on social media, leading to co-existence of multiple languages in the same sentence. Incidentally, these typically are the scenarios where TTS systems are widely deployed as speech interfaces and therefore these systems should be able to handle such input. Even though independent monolingual synthesizers today are of very high quality, they are not fully capable of effectively handling such mixed content that they encounter when deployed. These synthesizers in such cases speak out the wrong/accented version at best or completely leave the words from the other language out at worst. Considering that the words from other language(s) used in such contexts are often the most important content in the message, these systems need to be able to handle this scenario better.

In the next subsection, we first discuss briefly the issue of codemixing and then highlight the kind of codemixing that we are dealing with.

### 6.1.1 Code Mixing

CodeMixing is a phenomenon where linguistic units such as phrases, words and morphemes of one language are embedded into an utterance of another language (Gella et al., 2014). This is a common phenomenon in multilingual societies such as in India where English has transitioned from the status of a foreign language to that of a second language. Moreover, due to the diversity in terms of the regionality and the proficiency, the patterns of codemixing found are rather different from one another, often leading to confusion. (Muysken, 2000) states that there are, in general, three types of mixing:

- *insertion* or *embedding* of content (lexical items or entire constituents) from English into a regional language.
- *alternation* between structures from both the languages.
- *congruent lexicalization* of material from different lexical inventories into a shared grammatical structure.

(Muysken, 2000) also identifies that the lexical items that can be inserted during mixing are adverbarial phrases, single nouns and determiner-nouns. We performed an informal analysis on a Hindi recipe blog on the web and found that while the content was all in ASCII, around 15 percent of the words were English words (though often misspelt), and almost all of them were nouns or adverbs, in line with the observation in (Muysken, 2000). A similar analysis of a Telugu blog showed that around 20-30 percent of the text is in English (ASCII), and again, most

Table 6.1: Overview of Systems with variation in Grapheme to Phoneme Mapping.

| Level | Config | Description |
|---|---|---|
| Text | P2P_Mono | Phone to Phone in Monolingual System |
| Text | W2P_Mono | Word to Phone in Monolingual System |
| Text | Translit_Mono | Word to transliteration in Monolingual System |
| Spectral | Separate | Combined phoneset with language tag |
| Spectral | Shared | Combined phoneset without language tag |
| Spectral | Word | Language of word as question |
| Spectral | WC | Tri context for word is used as question |
| Prosodic | BL | Baseline statelevel Duration Prediction |
| Prosodic | Ratio | Ratio used for modification |
| Prosodic | Gauss | Gaussian used for modification |
| Prosodic | OLA | Only Look ahead features |
| Prosodic | OPF | Only phonetic features |

of them being nouns. Such mixed text data poses a variety of challenges to the speech synthesis system due to their innate characteristics such as contractions, non-standard pronunciation, and non-standard sentence constructions, etc.

Current approaches handling this scenario fall into three categories: phone mapping, multilingual or polyglot. In phone mapping, the phones of the foreign language are substituted with the closest sounding phones of the primary language, often resulting in a strong accented speech. In a multilingual setting, each text portion in a different language is synthesised by a corresponding monolingual TTS system. This typically means that the different languages will have different voices unless each of the voices is trained on the voice of same multilingual speaker. The polyglot solution refers to the case where a single system is trained using data from a multilingual speaker. Similar approaches to dealing with codemixing have been focused on assimilation at the linguistic level, and advocate applying a foreign linguistic model to a monolingual TTS system. The linguistic model might include text analysis and normalisation, a G2P module and a mapping between the phone set of the foreign language and the primary language of the TTS system (Tomokiyo et al., 2005; Campbell, 2001; Badino et al., 2004). Other approaches utilise cross-language voice conversion techniques (Mashimo et al., 2001) and adaptation on a combination of data from multiple languages(Latorre et al., 2005). Assimilation at the linguistic level is fairly successful for phonetically similar languages (Badino et al., 2004), and the resulting foreign synthesized speech was found to be more intelligible compared to an unmodified non-native monolingual system but still retains a degree of accent of the primary language. This might in part be attributed to the non-exact correspondence between individual phone sets.

In this work, we investigate approaches to build mixed-lingual speech synthesis systems based on separate recordings in the individual languages with the ability to appropriately synthesize the 'embedded' lexical items of English , thereby leading to a more natural output. Specifically, we build on the work done in (Wilkinson et al.) and investigate various techniques to train Indic voices that can speak both the primary language and also high-quality English, for the

common situation in which English text is encountered in a primarily Indian language document. We present systems at three different levels: Text level, acoustic modeling level and prosody modeling and try to answer the questions: (1) What modifications should we do at the G2P level so that the current systems can handle mixed text ? (2) How to train effective acoustic models that can handle mixed phone set ? and (3) What are the changes required at the prosodic level for generating natural sounding prosody? In section 7.6, we present the formulation and description of approaches at the text, spectral and prosodic levels. In section 16.3, we explain our experiments followed by evaluation and conclusion in section 7.7 .

## 6.2 Mixed Lingual Systems

In this section, we first present the formulation of our text based approaches and then describe our systems at spectral and prosodic levels.

### 6.2.1 Data

We have used speech and text data from 4 languages to build the systems described in this work: Hindi,Telugu, Marathi and Tamil. For Hindi, we have used the Mono and English parts of the male speaker from speech data released as a part of resources for Indian languages (Baby, 2006). We noticed that the Hindi utterances were longer and therefore used all the 1,132 prompts from the Arctic set but only the first 600 prompts from the Mono set so that both Hindi and English utterances are of equal duration (approximately an hour each). For the remaining languages, we have used the speech and text data that has been collected for (Wilkinson et al.). In all these cases speech data was sampled at 16 kHz and recorded in a high quality studio environment. For Telugu and Tamil we have used the recordings from female speakers and for Marathi, from male speaker thereby ending up with systems from two male and two female speakers overall. For evaluation, we have used the test sentences from multilingual category from (Prahallad et al., 2014) for the respective languages.

### 6.2.2 Approaches for Grapheme to Phoneme Conversion

Grapheme-to-phoneme conversion (G2P) is one of the first tasks in speech synthesis and essentially is a conversion from a word in orthography to its spoken form or pronunciation. This can be seen, in an oversimplification, as maximizing $Prob(P/G)$ where P is the phoneme sequence and G is the grapheme sequence of a single language. However, in case of 'embedding', the G contains graphemes from both native language as well as English. In this case, a phone to phone mapping is employed to map the phones from English to the native language.

However, in practice, this method results in a strong foreign accent while synthesizing the English words. (Elluru et al., 2013; Rallabandi et al.) proposed a method to use a word to phone

mapping instead, where an english word is statistically mapped to Indian language phones. This can be seen as maximizing the expression:

$$\prod_{i,j \in S,W} Prob(s_i|w_j) \tag{6.1}$$

where $w \in W^d$ is a word in source language with a vocabulary(W) of size d. The intuition in this can be seen as

$$\prod_{i,j,k \in S,M,W} Prob(m_k|w_j) * Prob(s_i|m_k) \tag{6.2}$$

where $m \in M$ was referred to as the mental mapping of the native speaker. In this work, we take a more direct approach and investigate the use of transliterations as the phoneme internal representation. We hypothesize that there is a single model which has generated both the transliteration and the phoneme itself. This serves as a more concrete mapping problem and can be seen as maximizing the expression

$$\prod_{i,j,k \in S,T,W} Prob(t_k|w_j) * Prob(s_i|t_k) \tag{6.3}$$

where $concat(t) \in T$ is the transliterated form. We have used this approach previously, (Sitaram et al., 2015a), but this the first time we are systematically comparing the three possible G2P approaches in such scenarios.

#### 6.2.2.1 Systems built

The systems we built at this level are mentioned in table 6.1. We have built a monolingual system (P2P_Mono) with phone to phone mapping as a baseline method. We then built systems W2P_Mono and Translit_Mono with word to phone and transliteration applied on the English words respectively.

#### 6.2.2.2 Pipeline

We follow a three step procedure. First, we identify the language of each individual word in the sentence. This is using a very simple method- based on the orthography of the word. We then apply the appropriate grapheme to phoneme conversion technique to the English words and obtain pronunciation, taking into account the corresponding postlexical rules. The final step is to generate speech using the sequence of phonemes.

### 6.2.3 Approaches for Acoustic Modeling

There are two dimensions in which we can vary the input features for synthesis. First at the phone level itself, choosing to explicitly separate the phones by original language (we add a language id suffix to the phone name), or taking the union of the phones across the languages (e.g. the data for English /t/ and Hindi /t/ are treated as one class). Secondly we provide contextual features to identify the language that the phone actually appears in (e.g. is it in an English or a Hindi word – and also the language id of the surrounding words). In the second case of language contextual tagging, these features may allow pronunciation distinctions between longer phrases in a particular language and isolated words in a codemixed utterance.

#### 6.2.3.1 Systems built

The systems we built at this level are mentioned in table 6.1. The system 'Separate' uses a combined phoneset, where the phones of English are explicitly separated from the phones of Indic by adding a language tag _E or _I denoting English and Indic respectively. The system 'Shared', as the name indicates, uses a combined phoneset obtained by the union of phones from English and Indic phonesets - if a phone is common in both the sets it is retained as is, and the disjoint phones are added separately. We then build systems 'Word' and 'WC' incorporating contextual features to identify the language. These systems differ in the amount of context used. The system 'WC' uses a tri word context while the system 'Word' uses a single word context.

### 6.2.4 Approaches for Modeling Prosody

In bilingual sentences, it was shown that the context of source language will influence the embedded target language words, which will change the original prosody of the target language (Zhang and Tao, 2008). In order to establish a mixed-language speech synthesis system based on separate corpora, it is therefore important to consider such influences to generate natural mixed-lingual prosody. From the corpus, we didnot observe much differences in pitch between the monolingual and mixed lingual synthesis systems, as in (Zhang and Tao, 2008). We suspect that this might be due to the proficiency level of bilingual speech, which is used on a daily basis in India. However, we did observe marked differences in the duration of the words, specifically at the point of switch from source language to the target language. In this subsection, we present our approaches to build combined prosodic models using separate monolingual speech and text data. We specifically look at two different ways of achieving this: (1) By manipulating the durations predicted by the baseline model and (b) Incorporating extra features while building the model, thereby modifying the prediction model itself.

- **Ratio based Manipulation System** (Zhang and Tao, 2008) In this system we first obtain the predicted durations from the baseline prosodic model and then transform the durations of English segments to account for the contextual effect that might be caused by the source language. The multiplication factor $\lambda$ was obtained using the mean durations of the segments during the training stage.

$$dur_{new} = \lambda * dur_{pred} \tag{6.4}$$

- **Gaussian based mapping system**

  We have observed that there seem to be two separate gaussian distributions followed by the phones from arctic and the indic recording sets. Therefore, we built system which modifies the durations of individual phonemes based on the following:

$$dur_{new} = \lambda * \frac{\sigma_{indic}}{\sigma_{arctic}} * (dur_{pred} - \mu_{arctic}) + \mu_{indic} \tag{6.5}$$

  where $\mu$ and $\sigma$ indicate the mean and standard deviation of the individual phonemes respectively.

- **System with only look ahead models** - These are the models that donot take the previous contex

### 6.2.5 Approaches for Acoustic Modeling

There are two dimensions in which we can vary the input features for synthesis. First at the phone level itself, choosing to explicitly separate the phones by original language (we add a language id suffix to the phone name), or taking the union of the phones across the languages (e.g. the data for English /t/ and Hindi /t/ are treated as one class). Secondly we provide contextual features to identify the language that the phone actually appears in (e.g. is it in an English or a Hindi word – and also the language id of the surrounding words). In the second case of language contextual tagging, these features may allow pronunciation distinctions between longer phrases in a particular language and isolated words in a codemixed utterance.

#### 6.2.5.1 Systems built

The systems we built at this level are mentioned in table 6.1. The system 'Separate' uses a combined phoneset, where the phones of English are explicitly separated from the phones of Indic by adding a language tag _E or _I denoting English and Indic respectively. The system 'Shared', as the name indicates, uses a combined phoneset obtained by the union of phones from English and Indic phonesets - if a phone is common in both the

sets it is retained as is, and the disjoint phones are added separately. We then build systems 'Word' and 'WC' incorporating contextual features to identify the language. These systems differ in the amount of context used. The system 'WC' uses a tri word context while the system 'Word' uses a single word context.

### 6.2.6 Approaches for Modeling Prosody

In bilingual sentences, it was shown that the context of source language will influence the embedded target language words, which will change the original prosody of the target language (Zhang and Tao, 2008). In order to establish a mixed-language speech synthesis system based on separate corpora, it is therefore important to consider such influences to generate natural mixed-lingual prosody. From the corpus, we didnot observe much differences in pitch between the monolingual and mixed lingual synthesis systems, as in (Zhang and Tao, 2008). We suspect that this might be due to the proficiency level of bilingual speech, which is used on a daily basis in India. However, we did observe marked differences in the duration of the words, specifically at the point of switch from source language to the target language. In this subsection, we present our approaches to build combined prosodic models using separate monolingual speech and text data. We specifically look at two different ways of achieving this: (1) By manipulating the durations predicted by the baseline model and (b) Incorporating extra features while building the model, thereby modifying the prediction model itself.

- **Ratio based Manipulation System** (Zhang and Tao, 2008) In this system we first obtain the predicted durations from the baseline prosodic model and then transform the durations of English segments to account for the contextual effect that might be caused by the source language. The multiplication factor $\lambda$ was obtained using the mean durations of the segments during the training stage.

$$dur_{new} = \lambda * dur_{pred} \tag{6.6}$$

- **Gaussian based mapping system**

  We have observed that there seem to be two separate gaussian distributions followed by the phones from arctic and the indic recording sets. Therefore, we built system which modifies the durations of individual phonemes based on the following:

$$dur_{new} = \lambda * \frac{\sigma_{indic}}{\sigma_{arctic}} * (dur_{pred} - \mu_{arctic}) + \mu_{indic} \tag{6.7}$$

  where $\mu$ and $\sigma$ indicate the mean and standard deviation of the individual phonemes respectively.

  - **System with only look ahead models** - These are the models that donot take the previous context into account while predicting the duration of the current state.
  - **System with only phonetic feature based models** - These are models trained without taking the names of phones into account and considering only the context in which they occur.

  t into account while predicting the duration of the current state.

- **System with only phonetic feature based models** - These are models trained without taking the names of phones into account and considering only the context in which they occur.

## 6.3 Experiments

All the systems were built using standard Clustergen (Black, 2006b) voice building procedure. Systems P2P_Mono and P2P_Multi were built using phone matching and systems W2P_Mono and W2P_Multi were built using epsilon scattering method (Black et al., 1998c), the idea in which is to estimate the probabilities for one grapheme G to match with one phone P, and then use string alignment to introduce epsilons maximizing the probability of the word's alignment path. We have followed the same procedure outlined in (Elluru et al., 2013). To transliterate the English words from the Romanized script to the native script as a part of the system Translit_Mono, we have used Brahmi-Net transliteration (Kunchukuttan and Bhattacharyya, 2015) which considers this problem as a phrase based translation. The sequences of characters from source to the target language are learnt using a parallel corpus trained using Moses (Koehn et al., 2007). This system supports 13 Indo-Aryan languages, 4 Dravidian languages and English including 306 language pairs for statistical transliteration. The systems at spectral and prosodic levels were built varying the question sets in the clustergen(Black, 2006b) voice building process appropriately.

Table 6.2: Results from Preference Test for Spectral Mapping Experiments among the systems using separate and shared phonesets

| Config | Separate | Shared | No Preference |
|---|---|---|---|
| Hi-En | 78/400 | **286**/400 | 36/400 |
| Te-En | 56/250 | **172**/250 | 22/250 |
| Ma-En | 48/250 | **167**/250 | 35/250 |
| Ta-En | 63/250 | **144**/250 | 47/250 |

### 6.3.1 Evaluation

Evaluation was performed in the form of listening tests using (Parlikar, 2012a). We have conducted two types of listening tests: (1) Rating the naturalness in terms of Mean Opinion Score (MOS) on a scale of 1(least natural) to 5(highly natural) and (2) ABX Preference test where the

Table 6.3: Results from Preference Test for Spectral Mapping Experiments among the systems using different levels of word context. Both these systems use shared phonesets

| Config | Word | WC | No Preference |
|---|---|---|---|
| Hi-En | 4/50 | 7/50 | **39**/50 |
| Te-En | 16/50 | **22**/50 | 12/50 |
| Ma-En | 11/50 | **26**/50 | 13/50 |
| Ta-En | 13/50 | **19**/50 | 18/50 |

Table 6.4: MOS Scores for Naturalness in prosodic modeling based experiments

| Config | Baseline | Ratio | Gauss | OLA | OPF |
|---|---|---|---|---|---|
| Hi-Eng | 3.9 | 3.8 | 3.8 | 3.6 | 3.4 |
| Tel-Eng | 3.6 | 3.7 | 3.5 | 3.4 | 3.5 |
| Mar-Eng | 3.7 | 3.7 | - | - | - |
| Tam-Eng | 3.4 | 3.3 | - | - | - |

users need to mention their preference towards either of the systems or state that they prefer neither. The systems using grapheme to phoneme based approaches and prosodic mapping have been tested using the former while the rest of the systems, i.e spectral modeling systems have been evaluated using preference tests. All the listening tests involved test sentences generated using the Multilingual test set (ML) from (Prahallad et al., 2014).

### 6.3.2 Discussion

#### 6.3.2.1 Front End

The word to phone mapping based systems seem to outperform the rest of the systems across all the languages consistently. The transliteration based systems seem to be performing better than the basic phone mapping based systems in the languages they were deployed in, but seem to be lagging behind the word to phone mapping systems. An informal preference test showed that the transliterated system does reduce the accented nature of phone mapping procedure for some words, but the reduction itself was not found to be as much as that obtained by the word to phone based modeling approach. Therefore it appears that using a separate orthographic system might not necessarily result in the best quality synthesis. One reason for this might be that the errors in the transliteration process itself act as barriers hindering the system from reaching its full potential in terms of voice quality.

#### 6.3.2.2 Spectral Modeling

The evaluation results for the spectral systems are presented in table 6.2 and 6.3. Each of the systems was used to generate 50 sentences from the test set which were evaluated students who were native speakers of the respective languages. The systems in Hindi were evaluated by 8 students, leading to 400 observation points while those from remaining languages were

evaluated by 5 students resulting in 250 observation points. We observe from table 6.2 that across all the 4 languages, the systems using a shared phoneset are preferred in a significant manner. From the results in 6.3, where all the systems were built using the shared phoneset due to the observed preference from 6.2, it appears that using the word context definitively doesnot deteriorate the system; at the same time it does not lead to a substantial imporvement either. We analysed the sentences that were not preferred by either of the systems in this case and found atleast two interesting characteristics common across the languages: (1) The English parts of the sentences seems to be a bit dull and flattened out compared to the native language counterparts. We hypothesize that this might be due to the manner in which the models were trained: using separate corpora as opposed to a multilingual corpus which has codemixed sentences, leading to a train test mismatch no matter how shared the phonesets are. The model when predicting the English segments might therefore be tending towards the mean of the training observations due to lack of proper context. It might be interesting to see if this artifact can be corrected/addressed by either using a codemixed database or by using some form of adaptation. (2) There seems to be an uncharacteristic gap between the English words which have two parts, ex: shortcut as short PAU cut , download as down PAU load, etc. We have anticipated this behaviour at the point of switch between the languages and therefore explicitly tried to model the context, but we didnot expect this in case of English words. In addition, we have observed that the stress pattern in these instances became a bit wierd.

#### 6.3.2.3 Prosodic Modeling

In this case, none of the systems that we have tried were successful in completely eliminating the seemingly disjoint(and therefore sudden variations in) speeds in the English and the Indic parts of test sentences. This is also clear from the evaluation results for the front end systems are presented in table 6.4. Surprisingly, none of the systems were able to outperform the baseline system, indicating that incorporating modifications artificially in the duration of the sentences is easily noticeable as unnatural by human evaluators.

## 6.4 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of content as well style information from speech utterance. Specifically, I present approaches towards build mixed-lingual speech synthesis systems based on separate recordings and present systems at three different levels. From evaluations, we have identified interesting issues which occurred as a result of the train-test discrepancy.

# 7

# SCALABILITY - De-Entanglement of Content and Style: Building code mixed voices using monolingual data

While the approaches presented in chapter 6 generate high quality code mixed speech, they make an assumption about the availability of bilingual voice talents who have similar proficiency in both the languages. This is an unrealistic constraint. In this chapter, I present various training strategies for building mixed lingual systems using only a monolingual corpora. I specifically discuss two strategies : (1) to generate the cross lingual training data and then build bilingual voices using techniques presented in chapter 6 and (2) to generate code mixed speech using latent variable models. Subjective evaluation shows that our systems are capable of generating high quality synthesis in code mixed scenarios.

## 7.1 Introduction

Although building voices using such a combination of multilingual corpora appears as a simple extension of multispeaker or multilingual speech synthesis, generating code mixed content is a deceptively non trivial task since there is a mismatch between training and the testing scenarios: Even though the model has access to data from both the participating languages during training, code mixed content it is exposed to at test time - as seen from the example sentences - is a novel composition of linguistic units from both the languages. In this work, we investigate approaches for building codemixed speech synthesizers using monolingual data by

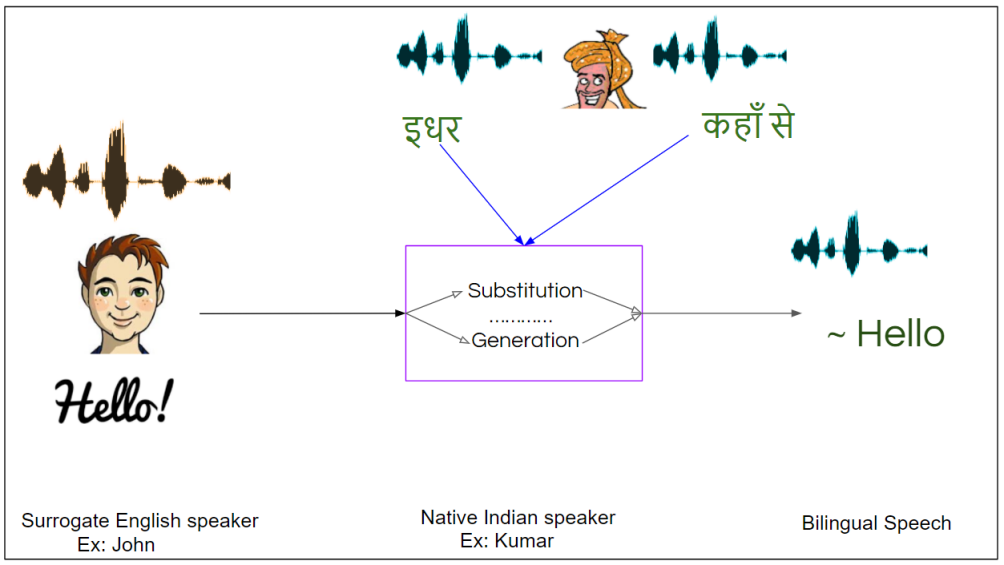



FIGURE 7.1: Illustration for the procedure of obtaining bilingual data using monolingual data from the native speaker by Frame Manipulation.

operating at the acoustic level. Specifically, we want to augment the voice building process of a monolingual Hindi speaker, e.g. Kumar, using acoustic data from a native English speaker, e.g. John with the intention of building a speech synthesizer that can render *Hinglish* content with the same/comparable quality as the voice would render a Hindi sentence.

### 7.1.1 Generating code mixed speech by first creating a pseudo dataset

This set of approaches have resemblance to (Gibson and Byrne, 2011; Sundermann et al., 2006) and (Qian et al., 2011). In (Sundermann et al., 2006), the authors use a frame selection and replacement method inspired by the principles of unit selection algorithm to obtain parallel corpus for the task of voice conversion. We use similar heuristics while selecting frames during the search algorithm but we search over different levels of speech representation. In (Gibson and Byrne, 2011), the authors use an external speaker to create an English model set for each Indian speaker. We have similar intuitions but we use a frame based approach which is at a much lower level of acoustics. The only similarities between our work and that of (Qian et al., 2011) is in the way of formulating the process of voice building as a two step process and that we operate at the frame level.

### 7.1.2 Generating code mixed speech directly using Variational Inference

Models with latent random variables (referred to as latent stochastic variable models hereafter) provide flexibility to jointly train the latent representations as well as the downstream network. They are expected to discover causal factors of variation present in the distribution of original data, so as to generalize at inference time. However, while training latent stochastic

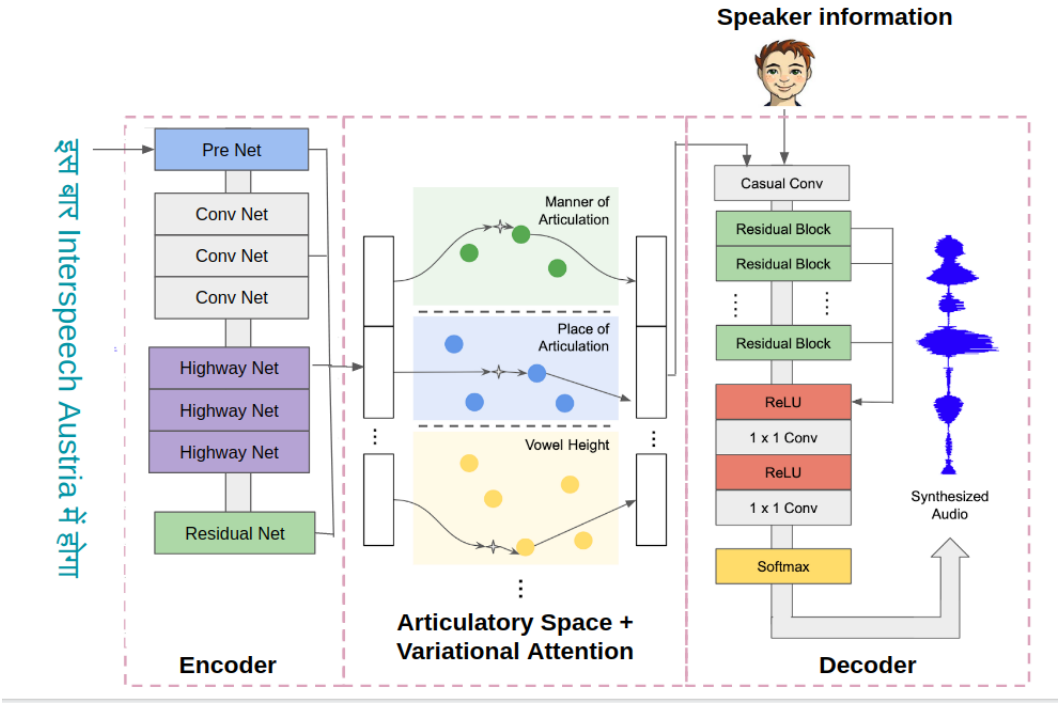



FIGURE 7.2: Illustration of our procedure for generating a code mixed utterance. Text from different languages is converted into a common representation space by Tacotron encoder. The encoded representation is hashed to a latent code based on a discrete articulatory prior bank. The code is passed to the decoder, followed by a WaveNet using speaker embeddings as global conditioning that generates audio.

variable models, optimizing the exact log likelihood can be intractable. To address this, a recognition network is employed to approximate the posterior probability using reparameterization (Kingma and Welling, 2014). We make an observation that articulatory information about speech production presents a discrete set of independent constraints. For instance, manner and place of articulation are two articulatory dimensions characterized by discrete sets(labial vs dental, etc). Based on this, we condition the recognition network in latent stochastic variable models to conform to articulatory prior space by using a bank of discrete prior distributions. We show that such priors help encode language independent information thereby facilitating synthesis of code mixed content.

## 7.2 Relation to previous works

Most of the current approaches accomplishing code mixed synthesis using monolingual data (Elluru et al., 2013; Chandu et al., 2017) operate primarily at the linguistic level: they either map the words/phones of the foreign language with the closest sounding phones of the native language or use transliteration (Sitaram et al., 2015b; Sitaram and Black, 2016a). While such text based approaches have been shown to be effective to generate foreign accents (Tomokiyo et al., 2005; Campbell, 2001; Badino et al., 2004) and are a good start for building code mixed systems, operating only at the linguistic level results in foreign accented pronunciations for

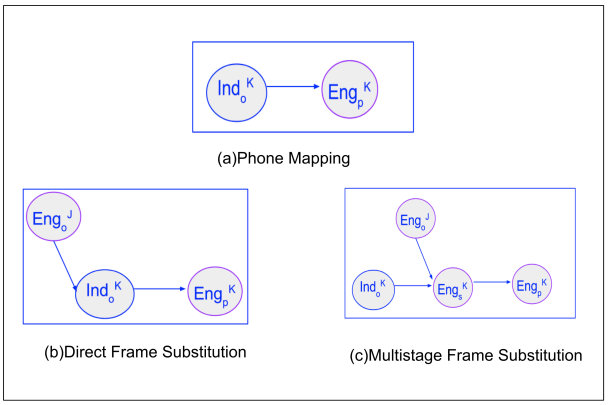



FIGURE 7.3: *Graphical Illustration of the approaches frame substitution.*

the English words in the sentence leading to a perception of unnaturalness. This might in part be attributed to the non-exact correspondence between phonesets of the individual languages being mixed. Approaches related to cross lingual synthesis on the other hand, employ techniques that operate at higher levels such as acoustics: for e.g. In (Qian et al., 2011), the authors follow a two step procedure of first warping the source speakers' speech parameter trajectories (in L1) towards the target speaker and then 'tiling' them with the data (in L2) to form a pseudo training corpus which is subsequently used to train a bilingual speech synthesis system. Similar practices can be found in the literature for voice adaptation (Kain and Macon, 1998; Kurimo et al., 2010; Oura et al., 2010; Yamagishi et al., 2009; Latorre et al., 2005) and voice conversion (Sundermann et al., 2006).

In section 7.6, we present the formulation and description of approaches. In section 7.4, we explain our the systems followed by evaluation and conclusion in section 7.7.

## 7.3 Building Two Stage Mixed Lingual Systems by Frame Manipulation

Our voice building process employs the phone sharing approach as outlined in (Rallabandi and Black, 2017) - where the combined phoneset is formed by the union of phones from both the languages - to build acoustic models. In this section, we present our approaches to generate the required bilingual acoustic data using just monolingual recordings. Specifically, in each of the subsections, we introduce the approaches we follow to artificially generate spectral content in English. We then follow the outlined procedure to build a bilingual voice using the native (L1) recordings and 'pseudo English' (L2) recordings.

### 7.3.1 Preliminaries - Notations and outline

Let us consider that the spectral representation for a single monolingual utterance in an Indian language(say, Hindi) spoken by a native Indian speaker (say, Kumar) is denoted by $Ind_o^K$. We

first build a spectral bank by combining all the available utterances (dropping the notation for utterance) and denote the same by $Ind^K$. Further, lets denote by $V^K_{mono}$ the monolingual voice built using $Ind^K$. We then generate a whole English corpus (say, Arctic) using $V^K_{mono}$, and use $Eng^K_{s,u}$ to denote each utterance. It is to be noted that we are not extracting spectral representation from the synthesized English utterances; we predict the spectral representation using the acoustic model of $V^K_{mono}$ and take the representation as is. We also denote by $Eng^J_{o,u}$ the spectral frames in the utterance u from the English corpus recorded by a (different) native English speaker (say, John) and $Eng^J_o$ denote the spectral bank obtained by concatenating all the English utterances. All of our approaches perform some form of perturbation either on $Eng^K_{s,u}$ or on $Eng^J_{o,u}$ using either $Ind^K$ or $Eng^J_o$ with an aim of generating a 'pseudo' English utterance and we denote it by $Eng^K_{p,u}$. Using the original recordings $Ind^K_{o,u}$ and $Eng^K_{p,u}$, we build the final bilingual voice $V^K_{bi}$.

### 7.3.2 Frame Substitution

The key idea in this approach is to generate $Eng^K_{p,u}$ by an iterative search over either $Ind^K$ or $Eng^J_o$. A figurative illustration of this approach can be found in figure 11.1. We propose two variants of this approaches: Direct Frame Substitution(DFS) and Multistage Frame Substitution (MFS). We refer to the approach of searching over $Ind^K$ as DFS. In this approach, for each $Eng^J_{o,u}$, we search for the closest acoustic frame from $Ind^K$. On the other hand, in MFS we iterate over each acoustic frame in each $Eng^K_{s,u}$ and search for the closest frame from $Eng^J_o$. We describe below the specific search and selection criteria used in each of the approaches. We start with two approaches that perform search over the MCEP spectral representation, followed by two approaches that operate on a transformation of the MCEP spectral representation.

#### 7.3.2.1 Direct Frame Substitution(DFS)

For this approach, we have used 25 dimensional MCEPs as the speech signal representation and Arctic for English spectral bank. The search algorithm for this approach can be mathematically visualized as:

$$\begin{aligned} Eng^K_p = \{Ind^K_o[idx] \mid idx = argmin([\|Ind^K_o - Eng^J_{o,u}\|_2] \\ +\|Ind^K_o[idx-1] - Ind^K_o[idx]\|_2)\} \end{aligned} \tag{7.1}$$

It can be observed that the formulation of 7.1 is similar to the formulation of unit selection algorithm. The first term in the linear combination of the cost functions represents the target cost while the second term is analogous to the concatenation cost ensuring the acoustic continuity. (Sundermann et al., 2006) notes that the weighting on these terms affects the quality of the downstream synthesis task. Therefore, we have tuned the weights for these terms using a

development set. In addition, we have ignored the first dimension(c0) as it mostly corresponds to energy and was observed to dominate the search process in our vanilla implementation. We further calculate the delta features and use voicing on the frame as the first decision before search: We search only the frames with matching voicing. Once the nearest frame is selected, we apply copy and apply VTLN transformation to the strengths of excitation corresponding to the selected frame as they were reported to have speaker information (Tsujioka et al., 2016).

#### 7.3.2.2 Multistage Frame Substitution(MFS)

Although DFS search criterion for obtaining the acoustic frames, we end up with the durations from the English speaker. In order to compensate this, we formulate MFS where we first generate $Eng^K_{s,u}$ and then use search algorithm to search for and substitute with the closest frame from the corresponding $Eng^J_{o,u}$. It can be seen that $Eng^K_s$ and $Eng^J_o$ form a parallel set of acoustic data. In a setting of voice conversion, the acoustic frames are aligned using dynamic programming and then joint acoustic model is built. Instead of alignment, perform a search operation in this case. For this approach, we have used the similar configuration as was used for DFS except two parameters: (1) terms in the Search criterion and (2) Number of cepstral coefficients used in search. The search algorithm for this approach can be mathematically visualized as:

$$\begin{aligned} Eng^K_p = \{Ind^K_s[idx] \mid idx = argmin([\|Ind^K_s - Eng^J_{o,u}\|_2] \\ +\|Ind^K_s[idx-1] - Ind^K_s[idx]\|_2)\} \end{aligned} \tag{7.2}$$

In this case, the search was performed based on only the first 8 cepstral coefficients and their deltas as it was observed to be more effective in our vanilla implementation.

#### 7.3.2.3 Systems using compact spectrum(DCS and MCS)

It was observed in (Mohammadi and Kain, 2014) that a compact spectral representation obtained using the framework of Auto Encoders proved useful for the task of voice conversion. Inspired by the same, we investigate if the formulation of such speaker independent compact spectrum improves the search algorithm. The mathematical formulation for this approach is equivalent to that of 7.1. We also follow the same procedure after obtaining the closest acoustic frame. However, the search is now performed along the dimensions of compact spectrum rather than the MCEPs. To obtain the compact representation, we train a denoising auto encoder following the architecture as described in (Mohammadi and Kain, 2014). The dimensionality of the compact spectrum was chosen to be 10 for DCS and 14 for MCS, after tuning on a validation set.

### 7.3.3 Frame Generation

In this subsection, we introduce the approaches we use to 'generate' $Eng^K_{p,u}$ instead of obtaining it by substitution. We employ Encoder Decoder neural networks as our models and introduce stochasticity in the form of latent variables at two levels: 1) At the input level, in the form of denoising encoder decoder model and 2) At the hidden layer level in the form of variational models. Decoders in both these models are frame level generators : They generate output frame by frame. However, the encoders in these models use both sequence level as well as frame level information while generating the embeddings input to the decoder. In the subsequent section, we describe our formulation of these approaches.

#### 7.3.3.1 Latent Frame Generation 01 (LFG 01)

In this approach, we use the denoising auto encoder model with the same architecture as was used for DCS and MCS to train each of the spectral banks $Ind^K$ and $Eng^M$. In addition to just using the frame level inputs, we augment the models with sequence level information using bidirectional context embeddings: We pass a left to right and right to left LSTM networks through the whole utterance and concatenate the embeddings from each LSTM network. After training, we input $Ind^K_s$ to generate $Ind^K_p$.

#### 7.3.3.2 Latent Frame Generation 02 (LFG 02)

Variational Auto encoders (VAEs)(Kingma and Welling, 2014) directed graphical models that implicitly represent the joint distribution between the data and a set of hidden variables as a product of two terms: a) prior over an 'inferred' set of latent variables and b) distribution of the visible variables conditioned on the inferred latent variables. We refer the readers to (Doersch, 2016) for detailed explanation. The architecture is trained by maximizing the variational lower bound on the marginal likelihood of the data and can be expressed mathematically as:

$$logp_\theta(x) \geq \mathbf{E}_{q_\phi(z|x)}[logp_\theta(x|z)] - KL[q_\phi(z|x)||p(z))] \tag{7.3}$$

where the first term on the right hand side is the variational lower bound (ELBO) and the second term corresponds to the KL divergence between the distribution estimated by the recognition model $q_\phi$ and the prior probability p(z). We make three modifications to this vanilla VAE model: We first add sequence level information to this frame level model by adding a bidirectional LSTM similar to the procedure in LFG 01. While training this model, we observed that the KL term reduces to near zero value very quickly leading to a state where the decoder network ignore the latent representation. In (Zhou and Neubig, 2017a), the authors use scheduled sampling to prevent this. However, we observe that the problem still remained in our case,

leading to very buzzy synthesis from the generation. We believe that this is because the joint distribution in the latent dimensions is too simplistic in the case of phonemes between English and Indian languages, as they share a large number of phones. This coupled with the simplistic assumption of a Gaussian distribution might be complicating the work of decoder. (Zhao et al., 2017a) proposes that removing the KL term completely would help maximizing the Mutual information between the latent representations. While that is practical for detection tasks, it is not suitable for generation tasks as we lose the ability to sample from the distribution. Therefore, to alleviate this, we incorporate a term $\lambda$ parameterized by an MLP and tie the KL term to optimize to $\lambda$. Therefore, the second term in 7.3 now becomes $\lambda$ + 'KL'. Finally, we make a modification to the latent representation z. Instead of using all the dimensions of z in the inference network, we leave the last L dimensions and use just len(z) - L in the inference network. The intuition behind this is to account for the variability of other factors such as language/channel identity: This is a way to give more flexibility to the latent representation to assign L dimensions to account for all the other factors but constrain it to assign the information relevant for inference in the len(z) - L dimensions.

## 7.4 Systems and Evaluation

We have used speech and text data from Hindi to build the systems using the Mono segment of the male speaker from speech data released as a part of resources for Indian languages (Baby, 2006). As baseline for comparison, we have built a voice using monolingual recordings and employing phone mapping. Once we obtain $Ind_p^K$ using each of the proposed approaches, we build a full voice using phone sharing approach outlined in (Rallabandi and Black, 2017). For fundamental frequency, we follow a z score mapping in the same was as is done during alignment in non parallel voice conversion. Evaluation was performed in the form of listening tests with 20 native Hindi students following the convention of Blizzard Challenge evaluations using (Parlikar, 2012a) with naturalness as the criterion in terms of Mean Opinion Score (MOS) on a scale of 1(least natural) to 5(highly natural). All the listening tests involved test sentences generated using the Multilingual test set (ML) from (Prahallad et al., 2014). The evaluation results are depicted in figure 7.4. When building the systems DFS and DCS, we observed that directly copying the unvoiced frames as pointed out by (Sundermann et al., 2006) improved the speed of the search process significantly without a perceivable degradation in the quality. An informal analysis on the outputs from the systems MFS and MCS revealed that the characteristics of the English speaker were retained in certain areas within the utterance, resulting in a slightly stylized version. This might be due to the search algorithm favoring the frames in $Ind^K$ which are close to the latent distribution of the frames of $Eng^J$. We want to investigate this further and hope to uncover techniques that can provide more control.

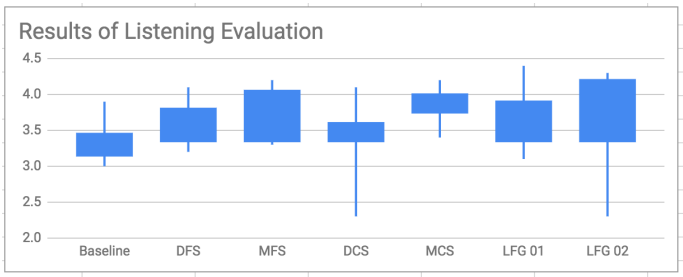



FIGURE 7.4: *Evaluation of various systems. While all systems outperform the baseline, the Multistage substitution techniques seem to have a clear advantage. In the models using generation, LFG 02 shows considerable improvement over LFG 01 indicating the fruitfulness of incorporating modifications to the vanilla models. However, the model also obtains low MOS scores indicating that the errors made are perceptually significant.*

## 7.5 Directly building Mixed Lingual Systems by employing Variational Inference

TABLE 7.1: Articulatory Features

| Feature name | Possible Classes | Cardinality |
|---|---|---|
| vowel or consonant | + - 0 | 3 |
| vowel length | s l d a 0 | 5 |
| vowel height | 1 2 3 0 - | 5 |
| vowel frontness | 1 2 3 0 - | 5 |
| lip rounding | + - 0 | 3 |
| consonant type | s f a n l r 0 | 7 |
| place of articulation | l a p b d v g 0 | 8 |
| consonant voicing | + - 0 | 3 |

We make an observation that dealing with speech presents a characteristic advantage - speech has both continuous as well as discrete priors. The generative process of speech assumes a Gaussian prior distribution which is continuous in nature. However, the language which is also present in the utterance can be approximated to be sampled from a discrete prior distribution. Exact manifestation of this in linguistics can be at different levels: phonemes, words, syllables, sub word units, etc. From the analysis presented in previous subsections, we posit that it helps encoder effectively disentangle the latent causal factors of variation if we use background knowledge about the data distribution while designing the priors. In other words, incorporating appropriated priors provides us with an opportunity to control what gets disentangled (or) decomposed (or) factorized in the latent space. In our context, an appropriate requirement from the encoder is to generate language agnostic yet phonetic representations such that a speaker dependent decoder can synthesize code mixed content. Therefore, we engineer our prior space to account for phonetic information in the utterance by representing the prior as a discrete latent variable bank, similar to filterbanks used for feature extraction from speech. Each discrete latent variable has a different set of states reflecting one of the articulatory dimensions. The specific design of our latent space is highlighted in the table

7.1. Voice building procedure with these priors is depicted in figure 7.2. We have used the articulatory dimensions according to the definitions in Indic voice building process of (Black, 2006b). Although some of them might be redundant, for this initial study we have retained all the articulatory dimensions. Without loss of generality, we assume that the individual latent articulatory dimensions are independent of each other. The divergence between the true prior and approximate prior now becomes:

$$D_{KL}(q_\phi(z_{enc}|p)||p(z_{code})) = \sum_{i=1}^{N}[$$
$$E_{q_\phi(z^i_{enc}|p)}[logq_\phi(z^i_{enc}|p)] - E_{q_\phi(z^i_{enc}|p)}[logp(z^i_{code})]]$$

where N is the number of articulatory dimensions and i denotes the index of individual articulatory dimensions. $z_{code}$ denotes the parameterized codebook and $z_{enc}$ denotes the representation output by the encoder.

## 7.6 Experiments

### 7.6.1 Data

We have used speech and text data from three Indian languages Hindi, Telugu and Marathi released as a part of resources for Indian languages (Baby, 2006) to build our synthesis systems. From our baseline voice building process, we found male speaker from Hindi to be the most reliable voice in terms of quality. Therefore, all of our systems use English recordings from Mono segment of this speaker as English set - as a scaffolding. For other two languages, we use only monolingual data from the speakers. In other words, to generate code mixed Telugu sentence, the systems have access to English content but from a different speaker. As baseline for comparison, we have built a CLUSTERGEN voice using monolingual recordings employing phone mapping. Evaluation was performed in the form of listening tests with 20 native students following the convention of Blizzard Challenge evaluations using (Parlikar, 2012a) with naturalness as criterion in terms of Mean Opinion Score (MOS) on a scale of 1(least natural) to 5(highly natural). All the listening tests involved test sentences generated using the Multilingual test set (ML) from (Prahallad et al., 2014). The evaluation results are depicted in table 15.2.

### 7.6.2 Implementation Details

We have built two systems employing variational attention: VQTacotron with vanilla vector quantization and VACONDA - with articulatory prior on the latent space. The architecture of our models continues from (Baljiker et al., 2018), with some modifications. We have used WaveNet(Van Den Oord et al., 2016) as our decoder. Following (Strubell et al., 2017), we have shared the parameters of all the residual layers with common dilation factors. We use Mixture

of Logistics loss to train the model and the number of logistics was set to 10. Speech signal was power normalized and squashed to the range (-1,1). To make the training faster, we have used chunks of 8000 time steps. Our quantizer performs vector quantization to generate the appropriate code from a parameterized codebook. We define the latent space $e \in R^{k \times d}$ contains $k$ $d$-dim continuous vector. Quantization is implemented using minimum distance in the embedding space. We have used 128 dimensions to perform the comparison in system VQTacotron. The number of classes was chosen to be 64, approximating 64 universal phonemes. For system VACONDA, we use a linear mapping to first project the 128 dimensional vector to 160 dimensions. We then perform comparison with respect to individual articulatory dimensions each of which is 16 in size. The speaker embedding is shared between the decoder of our acoustic model and WaveNet. We have noticed the lengths of utterances in the Indic datasets being too big to train attention from scratch. Therefore we have initialized attention using alignments performed within Festvox using HMM aligner. All the models were built at phone level since that was observed to be the most stable configuration even though our phones do not cover all the variants (ex. we do not have explicit phones for geminates). We have used quantization penalty and commitment loss terms as mentioned in (Chorowski et al., 2019b). In addition, we have also normalized each latent embedding vector to be on a unit sphere.

Table 7.2: MOS Scores for Naturalness in prosodic modeling based experiments

| Config | Clustergen | VQTacotron | VACONDA |
|---|---|---|---|
| Hi-Eng (Male) | 3.9 | **4.31** | 4.28 |
| Tel-Eng(Female) | 3.6 | 3.9 | **4.1** |
| Mar-Eng(Male) | 3.7 | 4.0 | **4.0** |
| Mar-Eng(Female) | 3.4 | 3.9 | **4.0** |

### 7.6.3 Observations

An informal analysis on the outputs from the proposed systems revealed that the characteristics of the English speaker were retained in certain areas within the utterance, resulting in a slightly stylized version[1]. We want to investigate this further and hope to uncover techniques that can provide more control. While most of the systems using CLUSTERGEN (Rallabandi and Black, 2017) make errors in the prosodic features such as irregular duration shifts at the boundaries between languages, the proposed approaches have smooth transitions at the boundaries. However, we have observed marked differences in the pronunciations by the proposed approaches. For instance, the phone 'S' from the word 'Stanford' when heard in isolation is indistinguishable from other fricative sounds. Since we specifically deal with articulatory priors in VACONDA, a reasonable assumption to make is that this issue will be bypassed by the model. However, this characteristic is common across voices built using both VQTacotron as well as VACONDA.

[1]The samples can be found here http://www.cs.cmu.edu/~srallaba/IS2019_CodeMixedTTS/.

## 7.7 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of content as well as style information from speech utterance. Specifically, we investigated approaches to build mixed-lingual speech synthesis systems based on separate recordings and present two types of systems: In the two step frame manipulation based approach, we have identified interesting issues which occurred due to the nature of the task as well as the nature of the algorithms used. In the direct approach we incorporate stochastic latent variables into attention mechanism and subject the latent variables to match articulatory constraints. Subjective evaluation shows that our systems are capable of generating high quality synthesis in code mixed scenarios.

# 8

# SCALABILITY - Applications: Synthesis of Navigation Instructions

## 8.1 Introduction

Navigation systems that can render instructions in the form of synthesized speech in addition to a visual interface are an important application of TTS Systems where being hands-free is critical. The text that needs to be synthesized in the navigation domain contains many named entities, such as names of roads and landmarks. The language that the TTS system is trained on may not be the same as the language that local place names are derived from. This may lead to pronunciation that does not seem natural, which may affect the usability of such systems.

Text for instructions is typically rendered in a single script. That is, although names of roads and landmarks are derived from a particular language, they are represented in the language that the TTS system is speaking in. For example, instructions being spoken by an Indian English TTS system for navigation in Bangalore will contain location names transliterated into the Roman script. In text that contains foreign named entities, language identification can be applied to categorize words so that corresponding phonetic rules are applied to each set accordingly. This scenario is different from code-mixing in the sense that only certain words, specifically proper nouns, belong to the native language. However, the influence of English still prevails in the names of the places as well, for example: 'road', 'park', 'mall', 'plaza' etc. An example navigation instruction that is collected between two locations in Delhi is:

*Turn\Eng left\Eng at\Eng Mukhiya\Hin Market\Eng Chowk\Hin onto\Eng Karawal\Hin Nagar\Hin.*

In this example, the words followed by '\Eng' and '\Hin' are English and Hindi words respectively. As stated before, there is a mixture of languages in the names of the places as well. For example, in 'Mukhiya Market Chowk', 'market' is an English word while the others are derived from native language Hindi. In this work, we extend our previous work on synthesizing code-mixed text using a monolingual voice to the domain of synthesizing navigation instructions using a bilingual voice. We build systems to synthesize navigation instructions using a Hindi-English bilingual voice for location names derived from Hindi, Kannada and Telugu. In addition, we conduct subjective listening tests to compare our system with a baseline monolingual system. Studies show that the performance of a driver is impacted by his cognitive load (Jonides, 1981). This may compromise the ability of the driver to perceive safety-critical events. Considering that TTS systems for navigation instructions are deployed in real time, it is imperative to aid the user with the provision of more natural auditory instructions. With the proliferation of ride sharing applications like Uber and Ola in countries like India, many individuals working as full-time drivers are now using navigation apps that have TTS systems. In some cases these drivers choose to use such apps voluntarily, while in other cases, the use of such apps is mandated by the cab company. Many of these drivers are semi-literate and have low English proficiency, and we conduct interviews and listening tests with them to evaluate our system. We also conduct listening tests with another set of users mostly comprising of graduate students, who have high English proficiency. In the remainder of the work, we refer to an English navigation instruction as native to a language if it has words derived from that language as the names of the places.

This chapter is organized as follows. Section 2 relates our work to previous work. Section 3 describes data collection. We describe our proposed technique in Section 4. Section 5 details subjective listening tests conducted with the two groups of users. Section 6 concludes.

## 8.2 Relation to Prior Work

Previous work in synthesizing multilingual speech can be classified into three approaches: bilingual TTS systems in which two speech databases are used from the same speaker to build a single TTS system, polyglot systems that create combined phonesets and phone-mapping based approaches. Bilingual TTS systems have been proposed by (Liang et al., 2007) for English-Mandarin code switched TTS. Microsoft Mulan (Chu et al., 2003) is a bilingual system for English-Mandarin that uses different frontends to process text in different languages and then uses a single voice to synthesize the text. Both these systems synthesize speech using native scripts, that is, each language is written using its own script. Polyglot systems (Traber et al., 1999) enable multilingual speech synthesis using a single TTS system. This method involves recording a multi language speech corpus by someone who is fluent in multiple languages. This

speech corpus is then used to build a multilingual TTS system. The primary issue with polyglot speech synthesis is that it requires development of a combined phoneset, incorporating phones from all the languages under consideration.

Another type of multilingual synthesis is based on phone mapping, whereby the phones of the foreign language are substituted with the closest sounding phones of the primary language. This method results in a strong foreign accent while synthesizing the foreign words, which may or may not be acceptable. Also, if the sequence of the mapped phones does not exist or does not occur frequently in the primary language, the synthesis quality can be poor. To overcome this, an average polyglot synthesis technique using HMM based synthesis and speaker adaptation has been proposed (Latorre et al., 2006). Such methods make use of speech data from different languages and speakers.

Recently, we proposed a framework for speech synthesis of code-mixed text (Sitaram and Black, 2016b) (Sitaram et al.) in which we assumed that two languages were mixed, and one of the languages was not written in its native script but borrowed the script of the other language. Our framework consisted of first identifying the language of a word using a dictionary or HMM-based approach, then normalizing spellings of the language that was not written in its native script and then transliterating it from the borrowed script to the native script. Then, we used a mapping between the phonemes of both languages to synthesize the text using a TTS system trained on a single language. We performed experiments on German-English and Hindi-English. We also conducted experiments to determine which language's TTS database should be used when synthesizing code-mixed text.

In this work, we extend our previous work in two ways: (1) Our current system is a bilingual system built using speech from two monolingual speech datasets and a combined phoneset, thereby removing the need for phone to phone mapping (2) We formulate our proposed approach and determine its effectiveness in the domain of navigation instructions.

## 8.3 Data

We used the Google Maps API to collect navigation directions from the locations where the following are the native languages: Hindi, Telugu, Kannada, Gujarati, Bengali, Marathi and Tamil. While we conducted listening tests for Hindi, Kannada and Telugu, this method is easily extensible to the other languages as well. The choice of these languages was based on access to native speakers in these languages to perform subjective testing. The navigation instructions used in GPS applications are in English, and so the syntactic structure of these instructions remains in English. The names of the places including native language words are considered words from the *embedded language* into English, which is the *matrix language*, in the matrix language-embedded language theory of code-mixing. Language Mix Ratio (LMR) is defined as the ratio of the number of words from the embedded language to the number of words in the

matrix language. Table 8.1 includes details about the data, including the LMR after using the language identification module mentioned in the following section.

TABLE 8.1: Navigation Instructions Data

| Language | # distinct routes | # sentences | LMR |
|---|---|---|---|
| Hindi | 399 | 4,806 | 0.2392 |
| Telugu | 1,974 | 19,976 | 0.1576 |
| Kannada | 8,898 | 108,178 | 0.1471 |
| Gujarati | 1,995 | 17,649 | 0.0942 |
| Bengali | 2,448 | 24,909 | 0.1852 |
| Marathi | 2,363 | 23,614 | 0.1977 |
| Tamil | 3,322 | 37,428 | 0.1612 |

The navigation domain has less spelling variations compared to general cross script code-mixing in social media observed from our previous work, where normalization is crucial. The navigation data we collected has fairly standardized spellings for the names of the places, although the native words of the places are transliterated into English.

## 8.4 Proposed Technique

Our proposed technique is similar to the pipeline we follow for synthesizing code-mixed text - first, we identify the language of each individual word in the sentence. Then, we transliterate the words that are not in English to the native script. This mixed script multilingual instruction is sent to corresponding G2P systems based on the language of lexical items. Finally a multilingual synthesizer is used to generate vocal navigational instruction. This section briefly outlines these stages.

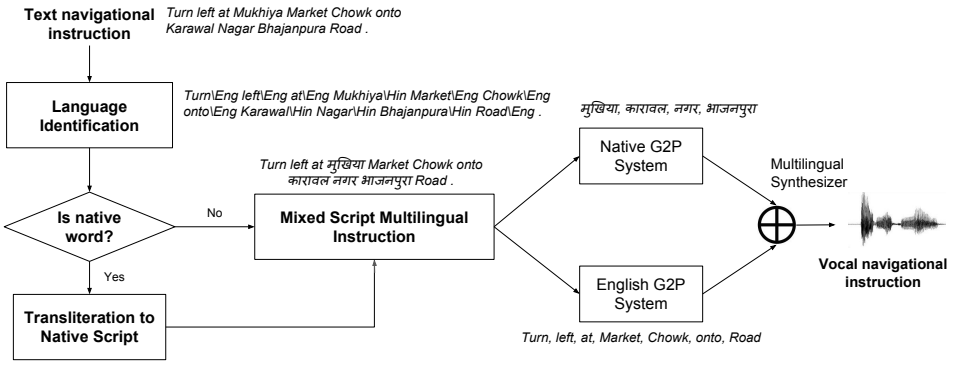



FIGURE 8.1: Architecture of the system with example of Hindi navigation instruction (Note that the language of the word 'Chowk' is misidentified and transliteration of 'karawal' is incorrect)

### 8.4.1 Language Identification

In this stage, the task is to identify names of places in the native language in each of the navigation instructions. One way is to use POS taggers and Named Entity Recognition tools to identify the names of locations in the instructions. We have attempted mapping named entities from wikipedia full text dumps with the ones found in navigational instructions by using soundex encodings. This method has a good coverage of important places but did not work well for local street names. In addition, as discussed in the introduction, we often find that place names contain English words like 'mall', 'park', 'station' etc, which need to be pronounced with English pronunciation rules. Hence, we identify the language of each word in the navigation instructions. We used an off-the-shelf system for language identification (Bhat et al., 2014) which uses character ngrams as features. Due to the specificity of the domain, we also attempt to mitigate errors made by the system by labeling common words like 'road', 'bus', 'main' as English words. This system covers all the languages of our interest, except for Marathi. Since this system is not trained to identify Marathi and English, we proxy Hindi for Marathi for language identification task. Though this is not ideal, it serves as a solution to differentiate English and non-English words.

### 8.4.2 Transliteration

To map the representation of native words to their corresponding phonemes used in the front end, these words are transliterated from the Romanized script to the native script. (Bhat et al., 2014) modeled transliteration as a structured prediction problem using second order Hidden Markov Models. In our initial experiments using Soundex codes, we mapped these transliterated words to words from large text of monolingual script (including wiki text dump, wiki titles and web pages from relevant queries) to derive a locality name. Even a very large amount of text had coverage issues with respect to proper nouns. We experimented with transliteration as a sequence to sequence problem by training an LSTM to convert English sequences for the names of the places to native script. We used 1000 parallel examples of Hindi words written in Devanagari and Romanized scripts from the FIRE task data (Choudhury et al., 2014) for this task. When trained on 800 samples and tested on 200 samples, the character level accuracy is 35.64%, while the word level accuracy is much smaller. The problems of recurring and invalid sequences of characters were addressed by building a language model of the native script. In similar lines of (Black et al., 1998c) which uses decision tree based letter to sound rules, we adapted this approach for the task of transliteration and for the same test set, we got a word level accuracy of 26%.

Brahmi-Net transliteration (Kunchukuttan et al., 2015) considers this problem similar to a phrase based translation problem, through which sequences of characters from source to the target language are learnt, where the parallel corpus is trained using Moses. This system supports 13 Indo-Aryan languages, 4 Dravidian languages and English including 306 language pairs for statistical transliteration. Using this, the accuracy corresponding to the correctness

of the entire word for the 200 test examples is 32.65%. Since this yielded higher accuracies at the word level, we proceeded with this scheme using their REST API to transliterate words into their native script.

### 8.4.3 Synthesis

The final step is to synthesize the navigation instructions that are transliterated into the appropriate script. Once we transliterate native language words, we synthesize the sentence using the bilingual TTS voice.

Speech data from Mono and English sets of the male speaker released as a part of resources for Indian languages (Baby et al., 2016) was used for these experiments. We used all the 1,132 prompts from the Arctic set recorded by a male Indian English speaker and used only the first 600 prompts from the Hindi set so that both Hindi and English utterances are of equal duration (approximately an hour each), as the Hindi utterances were longer. The speech data was sampled at 16 kHz and recorded by professional speaker in a high quality studio environment. For combining the English and Hindi phonesets, we used a simple phone clustering approach: the phones common in English and Hindi were retained as is and the phones present only in English were added resulting in a common phoneset. By doing this, we bypassed the phone-mapping process, which was shown to result in accented speech (Elluru et al., 2013) and would have limited the phones that could be used to those in the target language's phoneset. For getting pronunciations of native language words, we used the Festvox Indic frontend (Parlikar et al., 2016), which provides a g2p mapping between all Indian language UTF-8 code points and a phoneme from a common Indic phoneset. For some languages, rules like stress assignment, schwa deletion and voicing rules are implemented in the frontend. To build the voice, we followed the standard CLUSTERGEN (Black, 2006b) Statistical Parameteric Synthesis voice building process.

## 8.5 Evaluation

To perform preference testing, we synthesized navigation instructions using two methods. The first method was to retain all the lexical items in English. The second method used the proposed technique ie. language identification, transliteration and g2p using the native script. Both the methods used the same TTS voice trained using the bilingual data.

### 8.5.1 Preference testing

We conducted a user preference study to compare the baseline system to our proposed approach. We randomly sampled 20 navigation instructions in each of Hindi, Kannada and Telugu languages from the data collected and synthesized them. We used the Testvox web-based

framework (Parlikar, 2012a) for conducting these listening tests. Examples of these synthesized files can be found here [1]. We asked five native speakers each of Hindi, Telugu and Kannada to perform the listening test. We gave each speaker navigation instructions with location names derived from their mother tongue. We asked them to pick the sample that sounded more natural and understandable, with an option of choosing 'No preference' as well. Table 8.2 presents the results of this preference testing for three languages; Hindi, Kannada and Telugu. We can see that there was a significant preference for our proposed system in all three languages.

In addition to preference testing, we also did an informal study for intelligibility. For each of the languages, one student was asked to transcribe 20 navigation instructions and we recorded the number of times that the person had to listen to it to transcribe the sentence accurately. On an average, the transcriber had to listen 1.70 times for Hindi, 1.75 for Telugu and 2.15 for Kannada navigation instructions.

Table 8.2: Subjective listening tests

| Language | Prefer Baseline | Prefer Proposed | No Preference |
|---|---|---|---|
| Hindi | 17% | 70% | 13% |
| Telugu | 4% | 76% | 20% |
| Kannada | 19% | 69% | 12% |

The language identification module that we are using has an accuracy of 88.08%, 92.27% and 91.89% for Hindi-English, Telugu-English and Kannada-English language pairs. Some words are very ambiguous and the limited context may not be enough to identify the language correctly, particularly if the language identification system is trained on data from another domain. For example, the word 'to' is identified as a Hindi word as it is very common in Hindi (meaning: 'then'), however in the navigation instructions it is always an English word. We observed the following errors in the Kannada native words. The language identification system apart from using n-gram character features, also takes into account the context information from surrounding words. Hence the same word can be identified in different languages based on context. One such example is 'Jaraganahalli', which was identified as Kannada and English in two different instructions. Erroneous transliteration introduces some errors, for example, for words like 'Hosakerehalli' and 'Gubbi Thotadappa Road'. People acquainted with these locations however could still recognize them. As observed from Table 8.2, our system is preferred to a great extent in Telugu in comparison to other languages that we conducted this study. This could be because Telugu words are relatively longer than in the other languages, and hence English pronunciations of long Telugu words may be even more distracting.

[1] http://www.cs.cmu.edu/ kchandu/navigation/index.html

### 8.5.2 User study with drivers

In addition to conducting listening tests with users with high English proficiency and familiarity with speech-based systems, we also wanted to conduct user studies with a population of drivers who use navigation apps. These drivers are typically semi-literate and have low English proficiency and relatively low exposure to technology.

We conducted interviews and listening tests with 11 subjects who are full-time drivers in Bangalore. We briefed the drivers about the goals of the project, collected demographic data from them and asked them about their experience with GPS-based navigation systems, particularly about the TTS part of the systems. The drivers were given a mobile topup recharge of INR 50 (around 0.8 USD) as compensation for participating in the study. The entire interview was conducted in Kannada, the local language in Bangalore, although the TTS system itself was the bilingual voice described above. All the drivers in the study reported that they were familiar with locations in Bangalore, and almost all of them had lived in Bangalore for at least five years. Most drivers said that they had low English proficiency, with almost all of them saying that they could not speak or write English, but they could read and understand some English. All the drivers were multilingual, with some drivers knowing as many as five languages - Kannada, Tamil, Telugu and Hindi being the most common languages that drivers knew, with some drivers knowing some English and one driver also knowing Urdu. After the initial interview to collect demographic information, the drivers were given the same listening task as the previous study, with location names in Bangalore. Each driver listened to ten pairs of audio files using the Testvox interface. They were asked to choose the system that they could understand better, and one of the authors helped them navigate the web based listening test and answered any questions they had. Table 8.3 shows their listening preference between the baseline system and our proposed approach.

Table 8.3: Subjective listening tests with drivers

| Prefer Baseline | Prefer Proposed | No Preference |
|---|---|---|
| 34% | 60% | 6% |

From Table 8.3, it is clear that drivers had a strong preference for the proposed system. In many cases, they also pointed out specific words that they could understand better in the proposed system. The proposed system produced some extra schwas in some words which made it sound slightly unnatural, but the drivers did not point this out. In some cases, the drivers also pointed out that the (incorrect) pronunciation of a particular word in the monolingual system was similar to what they heard in the current navigation app that they used.

After the listening test, we asked drivers open-ended questions about their experience with navigation apps and suggestions for improvement. Some drivers had driven Ola and Uber cabs and had more experience with navigation apps, while others used them only when they went out of town, did not know a route or wanted to find out about traffic conditions. Surprisingly,

almost all drivers preferred the navigation instructions to be in English, rather than the local language or their native language. Their reasons for this were that the instructions used minimal English which they already understood, and they wanted the instructions to be in a language that their passengers could understand, so that there was more transparency with customer. They did however say that they knew of other drivers who knew no English who used the navigation app with the voice on mute because they could not understand it.

## 8.6 Conclusion

In this work, we presented techniques to synthesize navigation instructions in mixed language, where the instructions are rendered in one language and the names of locations are derived from another language. Such scenarios are common in multilingual countries like India where English is a widely-used language. For this work, we extended previous work in synthesizing code-mixed text, in which we first perform language identification and then transliterate native language words into the native script to derive appropriate pronunciation rules. We bypassed the step of mapping phones cross-lingually by using a bilingual TTS system to synthesize mixed-language navigation instructions. We performed experiments synthesizing navigation instructions with named entities derived from three Indian languages - Hindi, Telugu and Kannada. In subjective listening tests, there was a significant preference for our proposed approach compared to a monolingual Indian English system. We also performed a listening test and open-ended interviews with drivers with low English proficiency and found a preference for our proposed approach.

## 8.7 Acknowledgements

The authors would like to thank participants of the listening tests and interviews conducted in Bangalore and Pittsburgh. We also would like to thank Dr. Anoop Kunchukuttan and Professor Pushpak Bhattacharya for patiently assisting us with BrahmiNet. We also sincerely thank Hear2Read volunteers.

# Part III

# Flexibility

In this part I will provide an overview of experiments within the challenge of flexibility. I argue that flexible speech technologies should support the following scenarios:

- Detection of various para linguistic phenomena
- Synthesis of various extra linguistic effects

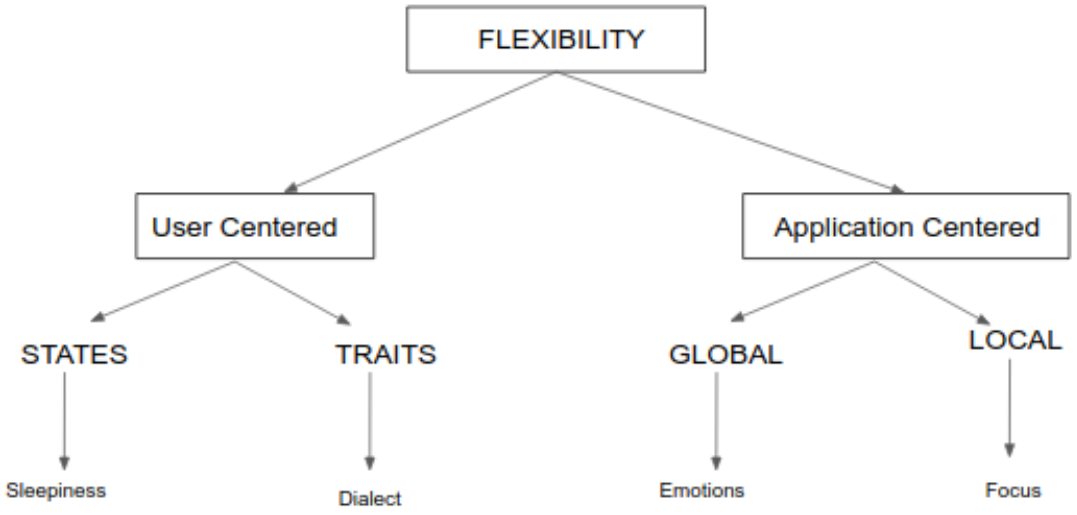



FIGURE 8.2: Taxonomy of Flexibility from the perspectives of Detection and Generation.

## Road map to Part 3

In chapter 9, I present experiments that demonstrate de-entanglement of style information using utterance level representations and then by using divergences in chapter 10. I then show how employing appropriate priors can help de-entangle content information from a speech utterance in chapter 11. Finally in chapter 12, I present experiments targeted at accomplishing local control at the word level in a neural generative model for speech by de-entangling both content and style information.

# 9

# FLEXIBILITY - De-Entanglement of Style using Utterance level Representations : A Case Study with Paralinguistic Event Detection

Recognizing paralinguistic cues from speech has applications in varied domains of speech processing. In this work I present approaches to identify the expressed intent from acoustics in three scenarios: 1) prediction of self-assessed affect and 2) detection of atypical affect 3) categorization of children's cries. Since emotion and intent are perceived at suprasegmental levels, we explore a variety of utterance level embeddings. The work includes experiments with both automatically derived as well as knowledge-inspired features that capture spoken intent at various acoustic levels. Incorporation of utterance level embeddings at the text level using an off the shelf phone decoder has also been investigated. The experiments impose constraints and manipulate the training procedure using heuristics from the data distribution. We conclude by presenting the preliminary results on the development and blind test sets.

## 9.1 Introduction

Applications of Computational Paralinguistics have grown rapidly over the last decade and span both human-human as well as human-machine interactions. The ComPare Paralinguistics challenges have been playing a significant role in driving progress in the diverse use of paralinguistics. Besides the traditional task of affect recognition using suprasegmental nonverbal aspects of speech, novel tasks were introduced, such as, the detection of speaker traits,

deception, conflict, eating and autism (Schuller et al., 2013, 2010, 2015, 2017). These challenges have shown that paralinguistic information can be used not only to identify affect but also clues that are helpful to detect abnormalities indicating disorders. Paralinguistic information also has applications in other domains of speech processing such as dialog systems, speech synthesis, voice conversion, assistance systems, and eHealth systems.

In this work, we present our approach to three of the INTERSPEECH 2018 ComPare sub-challenges (Schuller et al., 2018b): prediction of 1) self-assessed affect, 2) atypical affect and 3) types of crying. The *Self-Assessed Affect **(S)** Sub-Challenge* and the *Atypical Affect **(A)** Sub-Challenge* aim to classify affect from speech. In **(S)** ground-truth labels are provided by the speaker itself. The prediction of affect from speech oriented by the own assessment, could be used as a support in eHealth systems for individuals with affective disorders, such that a therapist can monitor the emotional state of their clients. In **(A)** the goal is to determine the affect of mentally, neurologically, and/or physically disabled individuals. The challenge is that some disorders also affect way people express their emotions. However, having a system able to detect distress in workplaces of disabled individuals can be helpful to make supervisors aware to suggest breaks or divide tasks in smaller ones, improving the emotional state of workers and therefore their concentration. The *Crying **(C)** Sub-Challenge* focuses on using paralinguistic information to identify affect in vocalisations of infants. Experts in the field of early speech-language development labeled audio-video clips into three classes of vocalisations: neutral/positive , fussing, and crying.

Typical approaches for classification and prediction of paralinguistic features include extraction of low level descriptive features followed by a machine learning model. Examples of low level descriptors are Mel-Frequency Cepstral Coefficients (MFCCs), log Mel-scale filter banks energies (FBANK) and several suprasegmental acoustic features that can be extracted using the openSMILE tool (Eyben et al., 2010). These features act as general purpose feature set and are expected to achieve competitive results in a wide range of paralinguistic problems. However, derived neural representations using unsupervised learning have shown impressive results on many speech and image based tasks recently (Aytar et al., 2016). These features usually embed the task relevant information from the entire utterance in a compact form. Also end-to-end learning models have been employed in affect classification using Long Short-Term Memories (LSTMs) or Gated Recurrent Units (GRUs) (Trigeorgis et al., 2016; Schuller et al., 2018b).

Motivated by this, we explore different utterance level representations and end-to-end approaches in the context of sub-challenges. Specifically, we investigate the significance of using both utterance level acoustic and derived linguistic features. We further employ data augmentation using utterance emphasis (see section 9.2.3.4) and random utterance segmentation (section 9.2.3.2), as a strategy to cope with class imbalance. For obtaining linguistic features we first obtain the text for each of the utterances using a pretrained English ASPire model. We then train a Recurrent Neural Network language model on the obtained text at the phone level and use the representation at the hidden state as the embedding of the utterance. Apart from this, we explore the applications of various Convolutional Neural Network models and chart

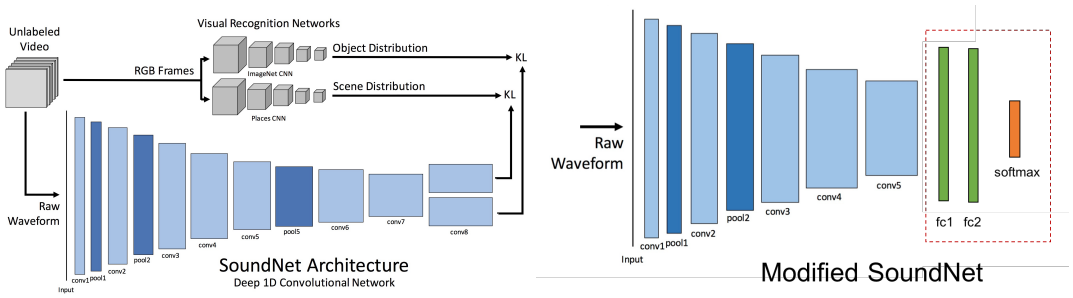



FIGURE 9.1: Original SoundNet architecture (Aytar et al., 2016) on top and modified SoundNet architecture at the bottom. The modified version uses 2 layers of 512 fully connected (fc) units and a softmax layer of 3 units.

their performance. It has to be noted that even though acoustic and phonetic embeddings use identical inputs, they differ in the higher level features learned internally. Therefore we believe that they complement each other producing a superior fusion result.

## 9.2 Framework

In this section, we present different features and classifiers used for all three sub-challenges. We used two different classification models: 1) Bidirectional LSTM using low-level features which uses temporal information , and 2) Random Forest classifier or SVM Classifier using high-level features, which are utterance based, combined with utterance level embeddings.

### 9.2.1 Temporal classification

#### 9.2.1.1 Low level features

For acoustic feature extraction we divided each utterance (length is $8\,\mathrm{s}$) into $25\,\mathrm{ms}$ segments with a $10\,\mathrm{ms}$ frame shift. For each frame we extract 13 mel-frequency cepstral coefficients and their deltas and double-deltas obtaining a feature vector of 39 dimensions. We further extract the log pitch ($f0$) and strengths of excitation (5 dim) (Yoshimura et al., 2001). In addition, we also obtain 40 dimensional filter banks and 23 dimensional PLP based features. Filter banks have been obtained using the open source toolkit Kaldi (Povey et al., 2011) with 'dithering' enabled as it was shown to be robust in other experiments. We have also extracted several features using Opensmile toolkit (Eyben et al., 2010) and performed singular value decomposition with the intention of obtaining an acoustic representation. This procedure also results in a dense low dimensional representation. This representation was later used in combination with the high level features we obtained in the spirit of early fusion.

#### 9.2.1.2 Classifier

Using all previously mentioned features, we train a 2 layer bidirectional LSTM network with 512 units in each cell. This is followed by 2 fully connected layers each with 512 units. The final softmax layer dimensions were dependent on the sub challenge. The network is trained by minimizing the expected divergence between the classes using cylindrical SGD (Smith, 2017).

### 9.2.2 Utterance-based classification

Recently, end-to-end approaches have shown impressive results on many speech based tasks (Trigeorgis et al., 2016). Specifically combinations of CNN and fully connected layers with a global pooling layer have obtained human level recognition rates on speaker and language recognition tasks. The global pooling layer functions as averaging sequential inputs therefore aggregating frame level representations to utterance level. This is advantageous for end-to-end learning.

#### 9.2.2.1 Extracting high level acoustic representations using Modified SoundNet

SoundNet (Aytar et al., 2016) is a convolutional network operates on raw waveforms and is trained to predict the objects and scenes in video streams at certain points. After the network is trained, the activations of its intermediate layers can be considered a representation of the audio suitable for classification. It has to be noted that SoundNet is a fully convolutional network, in which the frame rate decreases with each layer. Since we need to predict the emotions with reasonable recall, we cannot extract features from the higher layers of SoundNet directly.

The original SoundNet network has seven hidden convolutional layers interspersed with max-pooling layers. Each convolutional layer essentially doubles the number of feature maps and halves the frame rate. The network is trained to minimize the KL divergence from the ground truth distributions to the predicted distributions. In the original SoundNet architecture, the higher layers have been subsampled too much to be used directly for feature extraction. In order to fully exploit the information in the higher layers, we train a fully connected variant of SoundNet (see Fig. 9.1). Instead of using convolutional layers all the way up, we switch to fully connected layers after the 5th layer. We have also changed the input sampling rate to 16 kHz to match the provided data.

#### 9.2.2.2 Linguistic features

An informal analysis of the recordings indicated that the content being spoken plays a non trivial role in the valence of the utterance. A simple manifestation of this is the distribution of filled pauses and hesitations in the provided data across the classes. In the Self Assessed Affect dataset, examples belonging to the class ‘low’ have higher number of such irregularities

compared to the other two classes. Note that these features are not extracted for the Crying dataset. Therefore we hypothesize that using an off the shelf phoneme decoder to recognize such events might be beneficial. For this we first obtain the text at the phoneme level for each of the utterances using a pretrained English ASPire model from the toolbox Kaldi (Povey et al., 2011). We then train a Recurrent Neural Network language model on the obtained text at the phoneme level and use the representation at the hidden state as the embedding of the utterance.

#### 9.2.2.3 Classifier

We obtain the prediction scores from our models using either a Random Forest Classifier or a one-vs-rest classifier implemented using a binary SVM classifier depending on the performance. It is a known fact that SVM models perform better on sparse data than does trees in general. Therefore depending on the data augmentation techniques, we choose the classifier.

### 9.2.3 Data Manipulation & Enhancement

In this section we present various data engineering approaches that make the data more suitable for our models. Specifically, we explore approaches that aim to (a) obliterate the imbalance in class, (b) extract derived features which might help in distinguishing between the classes, (c) downsizing and normalizing on the duration of clips, etc.

#### 9.2.3.1 Class balancing by data restriction

In order to address the class imbalance present in the original data, we reduce the number of samples used for the classes that are dominant in the dataset. We hypothesize that the skewness of the original data causes low recall for classes that are in minority. Therefore, we study the effects of attempting to artificially balancing the classes by using less samples of dominant classes.

#### 9.2.3.2 Class balancing by data augmentation

The objective function we minimize in this approach is the expected divergence between the classes. An analysis of the original data points to the imbalance between the classes: For example, in Self Assessed Affect subchallenge, there are almost 3 times less number of examples for the 'low' class compared to the other classes in the training set. To alleviate this, we look at approaches to augment the existing data. Since our model operates on the sequence of frames, we hypothesize that segmenting the audio data into chunks (Agrima et al., 2017) exposes the model to different distributional properties. We obtain 4 times the original data for the class with less number of examples in Self Assessed Affect challenge by chopping the original signal between (0-2), (0-4), (0-6) and (0-8) seconds.

#### 9.2.3.3 Deriving Speaker Identity

Speaker normalization and adaptation have been widely documented as significant for a speech recognition system. As the original data did not have speakers tagged per utterance, we have tried to do speaker recognition using length normalized i Vector. i-Vectors are low-dimensional representation of GMM supervectors in a single subspace which have been formulated to include all characteristics of speaker and inter-session variability. Mathematically, given an observation set $X_s$, the adapted mean super-vector $m_s$ is modeled as,

$$m_s = m_0 + \mathbf{T}w_s + \theta \tag{9.1}$$

where $m_0$ is the Universal Background Model (UBM) supervector, and $\theta$ is the residual term which accounts for the variability not captured by T. Following Garcia-Romero and Epsy-Wilson (Garcia-Romero and Espy-Wilson, 2011), we perform a within class covariance normalization followed by length normalization of i vectors. These have been shown to 'gaussianize' the distribution and improve the performance of PLDA. iVectors have been extracted after log energy based voice activity detection on the utterances. This system was built within framework of Kaldi toolkit(Povey et al., 2011).

#### 9.2.3.4 Improving contrastiveness of features

We have tried to improve the contrastive nature among the classes artificially. An informal analysis of the recordings from Self assessed affect subchallenge led to the observation that the utterances with high valence were also relatively at a higher speed compared to the utterances with lower rate. Therefore we increased the rate of speech for the high valence utterances by 10 percent while simultaneously decreasing the rate of speech for low valence utterances by 10 percent. We performed similar perturbations with respect to pitch: boosting the pitch of the samples from 'high' class and lowering the pitch for the samples from 'low'. The samples for 'medium' class have not been subjected to any modification.

### 9.2.4 Early Fusion - Combining different representations

We have experimented with a feature level fusion of Soundnet layer 5 and ResNet50 (He et al., 2015) features extracted from the audio files. Resnet has been trained on around 1.28 images from the Imagenet dataset and has a top 5 error of 3.57% beating all other CNN image classifiers. We aim to systematically study the strategies of combining representations from multiple feature extractors.

## 9.3 Datasets

### 9.3.1 Self-Assessed Affect Recognition

The dataset used in this sub-challenge is the Ulm State-of-Mind in Speech (USoMS). It contains recordings of 100 students. The labels were obtained from the subjects themselves obtaining 3 classes: low, medium, and high.The class distributionfor combined train and dev sets are: 716 high, 698 medium, and 174 low. This highlights skewness in the data distribution.

### 9.3.2 Atypical Affect Recognition

The dataset comprised of a total of 10677 audio files out of which there are 3342 training, 4186 validation files and the remaining test files. There are four target classes that pertain to the four emotions - neutral, happy, sad and angry. The distribution of classes is again skewed with 5209, 1708, 516 and 175 being the total numbers of neutral, happy, sad and angry labels on the train and validation sets.

### 9.3.3 CRYING

This dataset is obtained from the Cry Recognition In Early Development (CRIED) database. It consists of 5588 vocalizations of 20 infants sampled at 44.1kHz in mpeg format. The objective is to identify three mood-related types of infant vocalization - neutral/positive, fussing and crying. The class distribution is as follows: 2292 cases of neutral/positive mood, 368 files of class fussing and the remaining 178 belonging to the class crying. The dataset is clean of vegetative sounds such as breathing sounds, smacking sounds, hiccups and so on. Further details about the datasets can be obtained from (Schuller et al., 2018a).

## 9.4 Experiments

In the following we present the preliminary results obtained using the systems we investigated on the Self Assessed Affect sub challenge. We further present the results of UAR for blind tests for all the three sub-challenges.

### 9.4.1 Class balancing by data restriction(System CBR)

We systematically try to reduce the data points from the classes with higher number of examples. The results from this experiment are depicted in Table 9.1.

TABLE 9.1: UAR for class balancing by data restriction

| Data split | | | UAR[%] |
|---|---|---|---|
| 100% Low | 100% High | 90% Medium | 56.8 |
| | | 70% Medium | 55.0 |
| | | 40% Medium | 52.1 |
| | 100% Medium | 90% High | **59.1** |
| | | 70% High | 56.8 |
| | | 40% High | 51.5 |
| All Data | | | 57.2 |

### 9.4.2 Speaker identity based experiments(System SI)

TABLE 9.2: UAR for Speaker identity based experiments

| | | Normalization | |
|---|---|---|---|
| | UAR[%] | used | not used |
| Speaker ID | used | 62.2 | 54.0 |
| | not used | 61.1 | **64.7** |

Since the classifiers we use are discriminative in nature, we experiment with two ways of incorporating speakers or subject specific information:

- (1) We add the identity of the speaker as an extra dimension thus forcing the model to build speaker specific models. For example, in case of decision trees, this forces the model to split at the identity of speaker.
- (2) Normalizing with respect to the speaker, following the procedure typically used in speech recognition.

The results from these experiments have been depicted in table 10.4.

### 9.4.3 Improving contrastiveness of features(System CTR)

We have explored two ways of artificially increasing the contrastiveness of the features, based on observations on the original data. Since the different classes appear to have a different distribution of artifacts such as hesitation, we have tried to use signal processing techniques to further separate the classes. Specifically, we have used festival toolkit (Black et al., 1998b) to decompose the signal into its spectrum, pitch and then apply class specific modifications to the utterances in the train set. The waveform was reconstructed using the vocoding framework within festvox voice building tools. We have used WSOLA (Verhelst and Roelands, 1993) to accomplish duration based manipulations. The results from these experiments are shown in the table 9.3.

Table 9.3: UAR for Emphasis and Data Augmentation Experiments

| Augmentation [%] | UAR [%] |
|---|---|
| 100 | 54.4 |
| 200 | 58.3 |
| 300 | 57.2 |
| 400 | 58.8 |

### 9.4.4 Blind Test Results and Discussion

The evaluation results on blind test set for the three sub-challenges is mentioned in the table 9.4. Based on the preliminary experiments, system **SI** appears to achieve a significant boost over the baseline before fusion. This seems plausible due to the nature of task at hand: emotions and intent have been known to be speaker specific. System **CTR** surprisingly does not have the expected gain in performance. We hypothesize that even though the premise of improving the class statistics by enhancing contrastiveness is valid, the manner in which we have performed the manipulation might be flaky. For example, given manipulating pitch might not be the best way to improve contrastiveness when the classes are separated by valence. However, we do see improvements with the Atypical affect subchallenge. Specifically, the recall for the class angry seems to improve with very little augmentation. Another observation with respect to system **CBR** is that the 'neutral' class seems to be very sensitive to any subsampling.

Table 9.4: UAR Blind test summary

| Sub-challenge | UAR |
|---|---|
| Self Assessed Affect | 48.3 |
| Atypical Affect | 34.2 |
| CRYING | 71.406 |

## 9.5 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of style information from speech utterance. For this, I employ detection of paralinguistic information as the example task. Specifically, we have explored the usage of both low level and high level features aimed at deciphering the intent from acoustics. In the preliminary experiments, higher level features seem to effectively embed the holistic information required for intent recognition. Since the datasets were highly skewed, we have explored various data augmentation and class balancing techniques. It might be beneficial to design architectures that exploit the nature of data and the constraints of the task.

## 9.6 Acknowledgments

This work was supported in part by a fellowship from the Portuguese Foundation for Science and Technology through the CMU – Portugal Program and the BioVisualSpeech project (grant CMUP-ERI/TIC/0033/2014) to Carla Viegas.

# 10

# FLEXIBILITY - De-Entanglement of Style using Alternative Divergences: A Case Study with Paralinguistic Event Detection

In this work we present our submission to the INTERSPEECH 2019 ComParE Sleepiness challenge. By nature, the given speech dataset is an archetype of one with relatively limited samples, a complex underlying data distribution, and subjective ordinal labels. We propose a novel approach termed ordinal triplet loss (OTL) that can be readily added to any deep architecture in order to address the above data constraints. Ordinal triplet loss implicitly maps inputs into a space where similar samples are closer to each other than different ones. We demonstrate the efficacy of our approach on the aforementioned task.[1]

## 10.1 Introduction

### 10.1.1 Paralinguistics

Paralinguistics refers to the aspects in a speech utterance beyond the linguistic content such as words. Paralinguistic cues such as accentuation are used to convey extra information such as emphasis, focus, expressiveness, and more. Applications of Computational Paralinguistics, the automatic analysis of such information, have grown rapidly over the last decade, spanning both human-human as well as human-machine interactions.

[1] Code is available at https://github.com/peter-yh-wu/ordinal

The ComPare Paralinguistics challenges have been playing a significant role in driving progress in the diverse use of paralinguistic information. Besides the traditional tasks such as emotion recognition using suprasegmental verbal and non-verbal aspects of speech, novel tasks such as the detection of speaker traits, deception, conflict, eating, and autism (Schuller et al., 2013, 2010, 2015, 2017) have been introduced. Detecting such information has the potential to not only play a role in assisting technologies with identifying affect but also play a role in detecting abnormalities indicating disorders. Paralinguistic information also has applications in other domains of speech processing such as dialog systems, speech synthesis, voice conversion, and more. In this work, we present our approach towards one such paralinguistics task - the detection of sleepiness from speech.

The advent of deep learning has brought forth a surge in high-performing speech models (Amodei et al., 2016; van den Oord et al., 2018). They have shown tremendous improvements in all the aspects of natural language processing (NLP), including speech recognition (Amodei et al., 2016), visual question answering (Kafle and Kanan, 2017), speech synthesis (Wang et al., 2017a), and more. The success of deep architectures in a variety of NLP tasks thus motivates their use in related areas including paralinguistics.

However, these models have been susceptible to learning just surface level associations and biases in the observed data, leading to overfitting and vulnerability to adversarial attacks (Chen et al., 2018a; Goyal et al., 2019; Kuhnle et al., 2018; Agrawal et al., 2017c). Therefore, there has been an interest towards learning algorithms that specifically consider intraclass relationships such as Siamese and triplet loss networks. Siamese networks have shown success in training on limited amounts of complex data (Koch et al., 2015). Therefore, we combine a Siamese architecture with ordinal regression techniques in order to effectively train the model based on the given the data constraints.

### 10.1.2 Ordinal Data

A significant amount of data generated by our world, from natural forces to human behavior, is effectively continuous. As a result, humans' tendency to bin continuous data (Tee and Taylor, 2018) has given rise to enormous amounts of ordinal data for applications ranging from healthcare to recommender systems (Marateb et al., 2014; Melville and Sindhwani, 2017). Thus, while humans tend to assign hard labels, the underlying data generally lies on a continuous spectrum. In order to perform effectively, statistical models must be able to capture the underlying data distribution rather than the humans' potentially subjective, and consequently noisy, discrete values. In a limited data setting where using sheer data size to generalize models is not an option, alternative techniques are required to make full use of the available data.

Leveraging the ordinal nature of a dataset as opposed to treating the classes as categorical is one effective approach for extracting more information from a limited set of samples. Many ordinal regression techniques have been proposed throughout the long-standing history of the field and have been traditionally applied to simpler tasks and non-deep models (Chu and

Ghahramani, 2005; Rennie and Srebro, 2005). For complex data that generally require deeper architectures, the large number of parameters in these ordinal techniques can tend to result in overfitting. Thus, simpler approaches are required in order to effectively integrate ordinal techniques into deep networks.

While treating continuous values as ordinals has good bearings intuitively, it is hard to train deep models that can effectively work with such data since standard classification techniques in deep learning are categorical. Hence, they cannot for example take into account the fact that class 2 is closer to class 3 as opposed to class 8. In order to effectively capture this information in a model, one approach is to construct an output distribution that reflects the relationship between classes. Soft labeling is one such technique that has been empirically shown to be effective with noisy ordinal data (Zhang et al.). Our proposed approach builds on this idea of leveraging ordinal relations to generalize from limited noisy data, namely via learning the relative distances between the encoded representations of different data samples.

## 10.2 Related Work

### 10.2.1 Speech Techniques

Typical approaches for classification and prediction of paralinguistic features include extraction of low level descriptive features such as Mel-Frequency Cepstral Coefficients (MFCCs), log Mel-scale filter banks energies (FBANK) and several suprasegmental acoustic features that can be extracted using the openSMILE tool (Eyben et al., 2010) followed by a classification model such as an SVM, decision tree, or neural network. While low level features act as general purpose feature sets, automatically derived neural representations using unsupervised learning (Aytar et al., 2016) have the potential to further increase model performances. Recently there has been a surge in the use of pretrained generative models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and more. These features usually embed the task relevant information from the entire utterance in a compact form. In accordance with this trend, end-to-end learning models have been employed in paralinguistics tasks (Trigeorgis et al., 2016; Schuller et al., 2018b).

### 10.2.2 Ordinal Regression

Each audio sample in the dataset is labeled with a number based on the KSS scale (Shahid et al., 2012). Since numbers on this scale follow a clear ranking, approaches in ordinal regression can be applied to this task. Namely, instead of penalizing all incorrect labels equally as in traditional multi-class classification, we can leverage the intuition that an incorrectly predicted class $\hat{y}$ that is numerically closer to the actual class $y$ should be penalized less than a farther $\hat{y}$. Two primary ordinal regression techniques that have been applied to statistical models include ordistic loss, which represents the output distribution as a mixture of Gaussians, and a thresholding-based

approach which learns the decision boundary between adjacent classes (Rennie and Srebro, 2005). Since both approaches involve many parameters, utlizing them in a deep architecture can lead to overfitting.

Soft labels have been shown to not only work effectively with neural models, but also help with convergence and training on noisy data (Hinton et al., 2015; Zhang et al.). While not originally created for ordinal tasks, empirical results suggest that soft labelling can be effectively applied to ordinal regression problems (Zhang et al.). In this work, we show why soft labelling is particularly effective for ordinal tasks and propose a general deep approach that learns ordinal relationships through soft labels and relative distance constraints.

### 10.2.3 Deep Metric Learning

Deep metric learning (DML) encompasses approaches that capture the similarity between datapoints via deep architectures. One such technique is the triplet loss function (Schroff et al., 2015), which constrains models to map input data from the same class to similar locations in an embedding space and data from different classes to separate locations. Specifically, the loss function for a triple $(x_a, x_p, x_n)$ with respective classes $y_a = y_p \neq y_n$ is given by

$$\|f(x_a) - f(x_p)\|_2^2 - \|f(x_a) - f(x_n)\|_2^2 + \alpha,$$

where $\|\cdot\|$ is the Euclidean norm, $f(x)$ is the encoded representation of $x$, and $\alpha$ is a hyperparameter representing the margin between same-class and different-class pairs.

Previous works have shown the effectiveness of triplet loss and Siamese architectures in limited data settings (Koch et al., 2015). Siamese networks perform well in such cases since they keep the number of model parameters low through weight sharing and effectively increase the dataset size through accepting multiple inputs at a time. Additionally, by encouraging input representations to cluster spatially by their class labels, these approaches can implicitly accentuate features useful for downstream classification tasks.

Like many other DML techniques, triplet loss is designed for categorical data, and consequently does not leverage any properties of ordinal data. We propose an augmented loss function, which we refer to as ordinal triplet loss in later sections, that captures the ordered nature of a collection of data through accounting for the absolute difference between class labels in its relative distance constraints.

## 10.3 Proposed Approach

Our proposed OTL approach is mainly comprised of two parts: soft labelling and an ordinal triplet loss function. Previous works have demonstrated the superiority of soft labels over hard labels for tasks with noisy data (Zhang et al.). In Section 3.1., we show that soft labels

are especially suited for ordinal tasks via a statistical interpretation. The ordinal triplet loss function serves to encourage the model to learn representations specific to the ordinal task at hand by adding a loss constraint to a hidden layer. We discuss the formulation of the loss function in Section 3.2 and how to integrate it into a deep architecture in Section 3.3.

### 10.3.1 Soft Labels

Results from Zhang et al (Zhang et al.) suggest that soft labels are well-suited for tasks with noisy, complex data. We reformulate their approach through a statistical lens in order to evince its particular effectiveness for ordinal tasks.

In a $K$-class ordinal task, we can uniformly scale a class label $k \in \{0, 1, \ldots, K-1\}$ to the interval $[0, 1]$, i.e. mapping class $k$ to $k/(K-1)$, without losing generality. Additionally, through associating a datapoint in original class $k$ with a pair $(k/(K-1), 1-k/(K-1))$ that sums to 1, we can reinterpret the class as a combination of binary labels. In other words, we can interpret the datapoint as being a combination of $k/(K-1)^{th}$ of a class-0 datapoint and $1-k/(K-1)^{th}$ of a class-1 datapoint. Assuming that the binary classes are generated from a Bernoulli distribution, we can express the likelihood of a set of data $\{x_1, x_2, \ldots, x_B\}$ with respective classes $\{y_1, y_2, \ldots, y_B\}$ as

$$\prod_{i=1}^{B} f(x_i)^{\frac{y_i}{K-1}} (1 - f(x_i))^{1-\frac{y_i}{K-1}},$$

where $f(x_i)$ is the model output for datapoint $x_i$. We can thus maximize this likelihood by training the model using the class pairs via cross-entropy loss. During test time, we invert the class-to-soft-label function to retrieve class predictions, namely mapping a pair $(\hat{p}, 1-\hat{p})$ to $\lceil \hat{p}(K-1) \rfloor$, where $\lceil \cdot \rfloor$ is the nearest integer function.

Training the model in this matter naturally penalizes class predictions more the farther they are from the true class, thus capturing the ordinal nature of the data. In fact, due to the curvature of the log likelihood function, loss penalties approximately increase exponentially with respect to distance to the middle class, capturing the central tendency bias inherent in datasets using the Likert scale. It is worth noting that this soft label formulation works with ordered data in general, including continuous data.

### 10.3.2 Ordinal Triplet Loss

Ordinal triplet loss augments the traditional triplet loss function (Schroff et al., 2015) by capturing ordinal relations, thus further utilizing properties in a limited corpus. Namely, the function adds a constraint ensuring that datapoints with farther class labels have larger distances between them in their embedded space. Each input triplet is comprised of an anchor sample $x_a$, another sample $x_s$, and a sample $x_d$ constrained to have a class farther from $x_a$ than $x_s$. In

other words, their respective class labels satisfy

$$|y_a - y_d| > |y_a - y_s| + \alpha,$$

where $\alpha \in \mathbb{N}$ is a hyperparameter. Since $x_s$ does not need to have the same class as $x_a$, the resulting set of possible triplets is noticeably larger than that of the traditional triplet loss formulation. When appropriate techniques described in Section 3.4 are applied to select which triplets to train, this expanded set of triplets can help the model generalize better. The ordinal triplet loss for a triplet ($x_a$, $x_s$, $x_d$) is given by

$$\sigma(\|f(x_a) - f(x_d)\| - \|f(x_a) - f(x_s)\|),$$

where $f(x)$ is the encoded representation of $x$, $\|\cdot\|$ is the Euclidean norm, and $\sigma$ is the logistic function, given by $\sigma(x) = \log(1 + e^{-x})$. Conceptually, the loss function penalizes cases where the model maps the $x$'s to representations where $x_a$ is closer to $x_d$ than $x_s$. The logistic function serves to make the loss function differentiable. Like the soft label approach, ordinal triplet loss can be applied to continuous data as well.

### 10.3.3 Network Architecture

We use an architecture similar to that of Zhang et al (Zhang et al.) to train our model, replacing their loss functions with ordinal triplet loss. Namely, the model receives triplet inputs and jointly optimizes the ordinal triplet loss function, which uses all three inputs, and the soft label cross-entropy loss, which uses only the anchor samples. Each iteration, the model embeds all inputs using an encoder $f$ before applying ordinal triplet loss, and passes the anchor sample embeddings through an MLP $g$ before applying the soft label cross-entropy loss. We add a batch norm layer between $f$ and $g$ to help with convergence. The loss function for a batch $\{(x_1, y_1), (x_2, y_2), \ldots, (x_B, y_B)\}$ is given by

$$\frac{1}{B}\left(\sum_{i=1}^{B} l_s(x_a^{(i)}, y_a^{(i)}) + \beta \sum_{i=1}^{B} l_t(x_a^{(i)}, x_s^{(i)}, x_d^{(i)})\right),$$

where $l_t$ is the ordinal triplet loss function, $l_s$ is the soft label cross-entropy loss function, and $\beta$ is a hyperparameter describing how much to weigh the ordinal triplet loss.

Conceptually, $f$ serves to separate embeddings in a manner that captures ordinal relations in order to help $g$ in the downstream classification task. As with other Siamese architectures (Koch et al., 2015), the weight sharing between elements in each triplet and the increased number of possible inputs via grouping samples into tuples aims to help with training effectively on limited amounts of complex data.

### 10.3.4 Implementation Details

Since the number of possible triplets is cubic with respect to the number of data samples, training using the traditional epoch formulation is impractical. Thus, we choose datapoints using an ordinal version of the triplet loss semi-hard sampling approach (Schroff et al., 2015). Namely, given an $(x_a, x_s)$ pair, we select the $x_d$ with the minimum $\|f(x_a) - f(x_d)\|$ that satisfies

$$\|f(x_a) - f(x_d)\| > \|f(x_a) - f(x_s)\|,$$

as well as the class label constraint $|y_a - y_d| > |y_a - y_s| + \alpha$.

## 10.4 Experiments

We describe in the following sections the experiments we conducted to achieve our best model. Our experiments generally proceeded in four parts: selecting features to train our models, modifying them to improve convergence, experimenting with soft labelling, and finally testing our proposed ordinal triplet loss formulation. All experiments used the Adam optimizer and a learning rate scheduler which decreased the rate by a factor of 0.1 after 10 epochs of no improvement.

### 10.4.1 Feature Selection

Table 1 describes the experiments we conducted to select the best features to use for our model. Features tested include the ComParE baseline features, SoundNet features, MFCCs, and raw waveforms. Of the ComParE baseline features, we observed that ComParE, BoAW-2000, and auDeep-fused yielded the best performances for both neural and statistical models. SoundNet features are extracted from the pretrained network with the same name (Aytar et al., 2016). We used the MFCCs to train a multi-layer LSTM augmented with an attention mechanism. The raw waveforms were used to train a deep network comprised of two convolutional layers followed by a multi-layer LSTM. For the SoundNet and baseline features, we used MLPs structured such that each subsequent layer in the network has approximately half the number of units as the previous one. SVM results for ComParE, BoAW-2000, and auDeep-fused are based on those reported in the challenge paper (Schuller et al., 2019). We observed that of the tested features, the three listed baseline features yielded the best results, as bolded in the table.

### 10.4.2 Data Modification

Table 2 on the next page describes the experiments we conducted to modify the input data. Namely, we tested upsampling and weighting the classification loss by class label frequencies as potential approaches to reconcile the skewed data distribution. We also tested applying PCA on

Table 10.1: Performance on Different Features

| | Model | Spearman (Devel) |
|---|---|---|
| SoundNet | MLP | 0.030 |
| ComParE | SVM | 0.251 |
| | MLP | **0.300** |
| BoAW-2000 | SVM | 0.269 |
| | MLP | **0.313** |
| auDeep-fused | SVM | 0.261 |
| | MLP | **0.329** |
| MFCC | Attention LSTM | 0.018 |
| Raw Waveform | CNN LSTM | 0.031 |

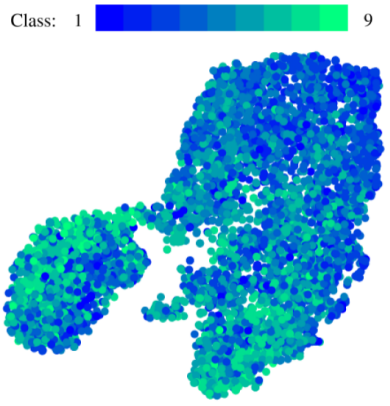



Figure 10.1: t-SNE Visualization of Embedding Space

the input features before feeding them into the model as a potential approach to reduce the high dimensionality of the features. For all the experiments in this section, we used MLPs with the halving property described in the previous section. We observed that these data modification approaches did not consistently improve the model, and thus did not use them in subsequent experiments.

### 10.4.3 Impact of Soft Labels

Table 3 describes the results from using the soft labelling formulation. All experiments in this section also used MLPs with the halving property described earlier. We observe that models trained on soft labels perform noticeably better than models trained on hard labels for two of the three feature types.

Table 10.2: Data Modifications

| | Features | Spearman (Devel) |
|---|---|---|
| Upsampling | ComParE | 0.271 |
| | BoAW-2000 | 0.308 |
| | auDeep-fused | 0.303 |
| PCA | ComParE | 0.279 |
| | BoAW-2000 | 0.325 |
| | auDeep-fused | 0.254 |
| Weighted Loss | ComParE | 0.279 |
| | BoAW-2000 | 0.301 |
| | auDeep-fused | 0.243 |

Table 10.3: Soft Labels

| Features | Spearman (Devel) |
|---|---|
| ComParE | **0.311** |
| BoAW-2000 | **0.333** |
| auDeep-fused | 0.322 |

### 10.4.4 Impact of Ordinal Triplet Loss

Table 4 below summarizes our results using ordinal triplet loss. We train all models in this formulation using the Adam optimizer with learning rate $10^{-7}$, the joint loss described in Section 3.3, batch sizes of 64, and early stopping with a patience of 10. For our models trained via ordinal triplet loss, $f$ is an MLP with input dimensions halved for each subsequent layer, and $g$ is comprised of two fully connected layers. We observe that utilizing ordinal triplet loss yields noticeable improvement in model performance with respect to the BoAW-2000 feature set.

Table 10.4: Ordinal Triplet Loss

| Features | Spearman (Devel) |
|---|---|
| ComParE | 0.308 |
| BoAW-2000 | **0.343** |
| auDeep-fused | 0.323 |

### 10.4.5 Analysis of Results

Figure 1 plots the t-SNE visualization of the training data in our model's embedding space. Lighter points represent data samples with higher class labels. The model is able to successfully learn a space that captures desirable ordinal relations, generally mapping data with closer class labels to closer locations in the embedding space.

## 10.5 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of style information from speech utterance. For this, I employ detection of sleepiness information from speech as the example task. Specifically, In this work, we present ordinal triplet loss as an effective way to train deep architectures on noisy, complex, ordered data. We show that soft labels work particularly effectively in ordinal regression tasks. We propose an ordinal triplet loss function that captures ordinal relations in its embedding space, which we validate empirically on the Sleepiness dataset. Finally, we show that our proposed approach performs well on the Sleepiness dataset. In the future, we are interested in exploring how well our approach performs on continuous data in order to show an effective deep technique on complex regression tasks.

# 11

# FLEXIBILITY - De-Entanglement of Content using Priors: A Case Study with Acoustic Unit Discovery

In this work, we present an approach to automatically discover acoustic-phonetic units from a speech utterance in an unsupervised fashion. We first present an analysis to show that incorporating latent random variables into the neural generative models using suitable priors allows us to control what gets encoded into the latent space. Based on this, we employ articulatory features as discrete prior bank in the latent space and obtain acoustic units that are speaker and language independent. Through experiments the proposed model achieves significantly better fidelity compared to the baseline model with a lower bit rate.

## 11.1 Introduction

A major bottleneck in the progress of many data-intensive language processing tasks such as speech recognition and synthesis is scalability to new languages and domains. Building such technologies for unwritten or under-resourced languages is often not feasible due to lack of annotated data or other expensive resources. A fundamental resource required to build such a stack is a phonetic lexicon - something that translates acoustic input to textual representation. Having such a lexicon, even if noisy, can help bootstrap speech recognition models, synthesis, and other technologies. Typical approaches may involve a pivot language or bootstrapping or

adapting from a closely related high-resource language. But, this can be a deceptively non-trivial task due to linguistic differences which can pose inherent difficulties. For instance, it may be unreasonable to analyze a Sino-Tibetan language using English as a source. Moreover, using an additional language might make the model learn unintended surface level associations or biases between the participating languages that prevent them from generalizing across languages. Associations between these languages over a set of units that may better generalize to other languages. Therefore, in this work we are interested in discovering the appropriate acoustic phonetic units.

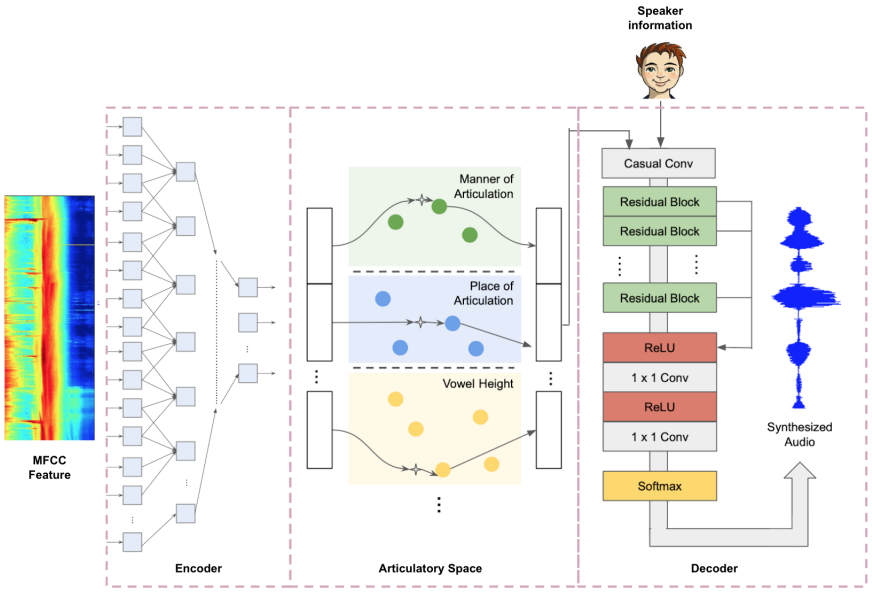



FIGURE 11.1: Illustration of our procedure for automatically discovering acoustic units from a speech utterance. We pass the speech utterance through a downsampling encoder. The encoded representation is hashed to a latent code based on a discrete articulatory prior bank. The code is passed to the decoder, a WaveNet using speaker embeddings as global conditioning that regenerates audio.

In ZeroSpeech Challenge(Ewan et al., 2019) resynthesis is considered a good proxy task to evaluate the performance of systems when training using unsupervised approaches. To accomplish this we use neural generative models. Deep Neural Generative models have seen a tremendous amount of progress in the recent past. These models aim to model the joint probability of the data distribution and the conditioning information as a product of conditional distributions. Typical implementations of such models follow an autoregressive framework, although other formulations have been suggested as well. Such models have been shown very effective in addressing one of the major challenges with conventional vocoding techniques - fidelity. Neural generative models has been shown to generate speech that rivals natural speech when conditioned on predicted mel spectrum (Shen et al., 2017).

Speech has a lot of natural variations in terms of content, speaker, channel information, speaking style, prosodic variations, etc. Accordingly, we are interested in models which have flexibility to marginalize such variations but preserve the phonetic content and distinguish meaningful differences between phonetic units. To accomplish this, we employ sequence to sequence

models with latent random variables (referred to as latent stochastic models hereafter). These models provide a mechanism to jointly train both the latent representations as well as the downstream inference network. They are expected to both discover and disentangle causal factors of variation present in the distribution of original data, so as to generalize at inference time. While training latent stochastic models, optimizing the exact log likelihood can be intractable. To address this, a recognition network is employed to approximate the posterior probability using reparameterization (Kingma and Welling, 2013). When deployed in encoder-decoder models, this approach is often subject to an optimization challenge referred to as KL-collapse (Bowman et al., 2015), wherein the generator (usually an RNN) marginalizes the learnt latent representation. Typical approaches to dealing this issue involve annealing the KL divergence loss (Bowman et al., 2015; Zhou and Neubig, 2017b), weakening the generator (Zhao et al., 2017b) and ensuring the recall using bag of words loss.

In our work, we present an approach to deal with the KL-collapse problem by vector quantization in the latent space. Building on (van den Oord et al., 2017a; Chorowski et al., 2019b), we add additional constraints in the prior space forcing the latent representations to follow articulatory dimensions: The encoded representation is hashed to a latent code based on a articulatory prior bank designed using a discrete codebook. Our decoder is a conditional WaveNet using speaker embedding as global embedding trained to regenerate input audio using the code sequence as local information.

## 11.2 Background

### 11.2.1 Acoustic Unit Discovery

Let us consider a speech corpus X which consists of speakers $\{s_1, s_2..., s_n\}$. The goal of acoustic unit discovery is to come up with a set of units **U** that represent a speech utterance $x \subset$ X allowing robust resynthesis. The elements of such a set also might conform to desirable characteristics such as being injective, consistent and compact, i.e. that different inputs should have discriminant acoustic units, but expected variance such as speaker or dialect should produce the same acoustic units.

There have been numerous attempts to discover such acoustic units in an unsupervised fashion. In (Huijbregts et al., 2011), authors presented an approach to modify the speaker diarization system to detect speaker-dependent acoustic units. (Jansen et al., 2013) proposed a GMM-based approach to discover speaker-independent subword units. However, their system requires a separate Spoken Term Detector. Recently, due to the surge of deep generative model, using unsupervised method such as auto-encoder and variational auto-encoder (VAE). (Badino et al., 2014) designed a stacked AutoEncoder using backpropagation and then cluster the representations at the bottleneck layer. To avoid quick transitions leading to repeated units, they employed a smoothing function based on transition probabilities of the individual states. (Ebbers et al., 2017) extended the structured VAE to incorporate the Hidden Markov Models as latent

model. (van den Oord et al., 2017a; Chorowski et al., 2019b) proposed VQ-VAE and argue that by vector quantization the ""posterior collapse" problem could be circumvented.

### 11.2.2 Disentanglement

There have been many attempts to interpret and manipulate the working of ELBO through analysis or exploitation of the latent space. For instance, in (Chen et al., 2018b), authors decomposed the ELBO term and showed that there are terms measuring the total correlation between the latent variables. (Esmaeili et al., 2018a) introduced a generalization of ELBO by factorizing the latent representations into a hierarchy, while (Ansari and Soh, 2018) presented an approach to accomplish disentanglement by modifying the co-variance matrix of the latent representations. Lastly, (Kim and Mnih, 2018) augmented ELBO using the density ratio trick to accomplish disentanglement. Our work is similar to these in that we analyze ELBO to show that it is possible to control what gets disentangled.

Other works have focused on the prior distribution and causal factors, such as (Hoffman and Johnson, 2016), which posited that to improve ELBO we must also improve the marginal KL e.g. we must have good priors, and (Banijamali et al., 2017), which showed that actively trying to disentangle the causal factors of variation is better than trying to pressure the model to forget the invariant representations. In (Burgess et al., 2018a), authors proposed incorporating a channel capacity term to promote disentanglement of these causal factors. We take inspiration from previous approaches that manipulate the prior distribution, but in our work, specifically incorporate articulatory constraints on the prior space. Doing so has additional benefits such as interpretation of the intermediate model outputs. Our implementation use an information bottleneck, which was shown in (Alemi et al., 2016) to help models become robust to adversarial attacks as well. However, such analysis is beyond scope of the current study.

### 11.2.3 Neural Generative Models for Speech

Artificial generation of speech based on neural approaches has soared in the recent past. There have been continuous and significant improvements in both the aspects of speech generation - fidelity and flexibility. Autoregressive models such as (van den Oord et al., 2016), flow based models such as (Prenger et al., 2018a) have shown to generate audio that rivals the quality of natural speech. Approaches such as (Taigman et al., 2017b,a) have shown ways to incorporate inductive biases into the generative process. (Watts, 2012) developed generic methods to enable the usage of distributional analysis of text at phone, word, and character levels in an unsupervised fashion. These techniques have been utilized in building highly flexible systems capable of generating different styles of speech and ability to build voices from noisy or very minimal data.

### 11.2.4 Speech Chain

There have been attempts to combine the ASR model and TTS system to form a closed-loop speech chain inspired by their closely dependent nature. (Tjandra et al., 2017) proposed the first deep sequence-to-sequence model in close-loop architecture allows us to train our model on the concatenation of both labeled and unlabeled data.

## 11.3 Proposed Approach

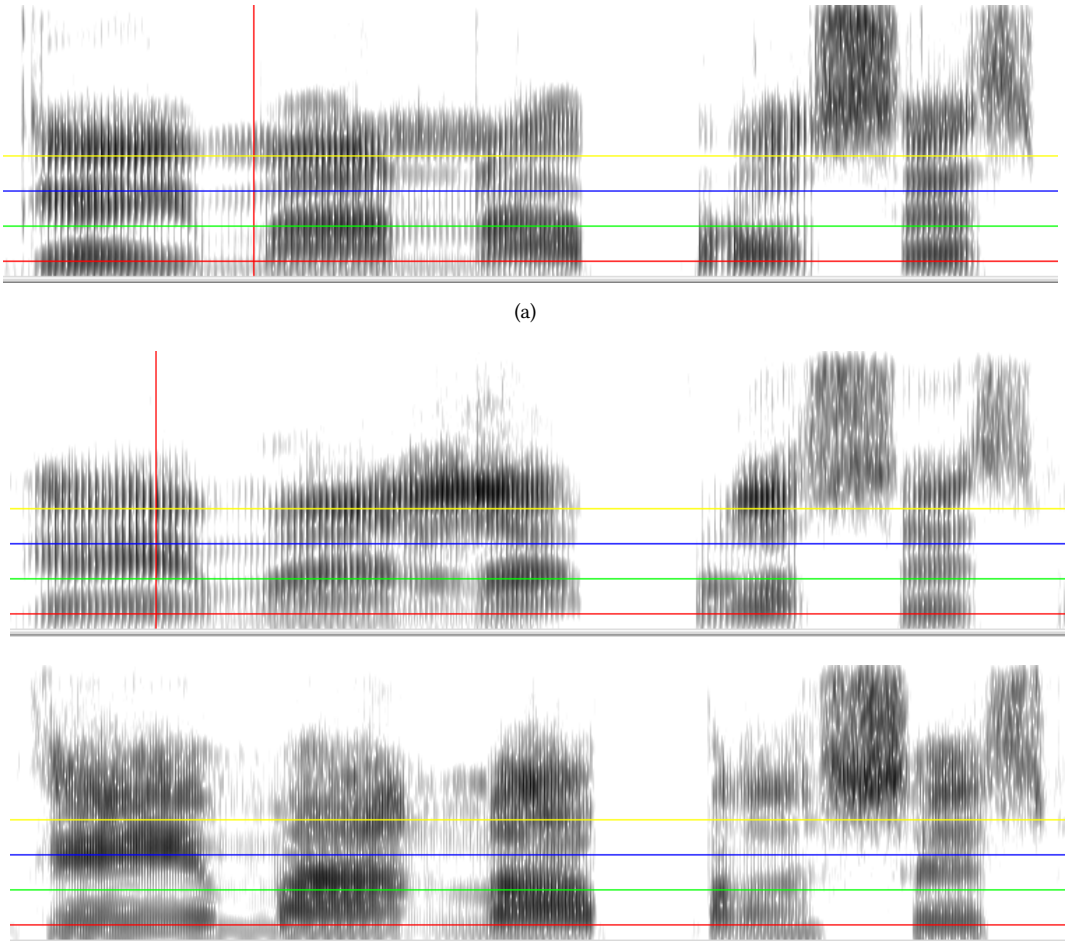
For experiments in this case study, we extend the articulatory priors based model presented in chapter 6. The architecture of our model is built on top of VQ-VAE. It consists of three modules: an encoder, quantizer and a decoder. As our encoder, we use a dilated convolution stack of layers which downsamples the input audio by 64. The speech signal was power normalized and squashed to the range (-1,1) before feeding to the downsampling encoder. To make the training faster, we have used chunks of 2000 time steps. This means we get 31 timesteps at the output of the encoder. The quantizer acts as a bottleneck and performs vector quantization. Quantization is implemented using minimum distance in the embedding space. The number of classes was chosen to be 64, approximating 64 universal phonemes. We use a linear mapping to first project the 128 dimensional vector to 160 dimensions. We then perform comparisons with respect to individual articulatory dimensions each of which is 16 in size. Assuming $z_e(x)$ denotes the encoder output in the latent space, then the input of decoder $z_d(x)$ will be obtained by ${}_jd(e_j, z_e(x))$, where $d$ is a similarity function of two vectors. In this work, we consider Euclidean distance as the similarity metric. Our decoder is an iterated dilated convolution-based WaveNet that uses a 256-level quantized raw signal as the input and the output from vector quantization module as the conditioning. Although using a Mixture of Logistics loss function might yield a better output, we have only used a 256 class softmax in this study. The decoder takes the output from the quantizer along with the speaker label as global conditioning and aims to reconstruct the input in an auto regressive fashion. Following IDCNNs, we have shared the parameters of all the stacks.

## 11.4 Experiments

### 11.4.1 Dataset

ZeroSpeech Challenge 2019: TTS without T is to propose to build a speech synthesizer without any text or phonetic labels (Sakti et al., 2008b,a; Ewan et al., 2019). The systems are required to extract the symbolic representation of the raw audio, and then re-synthesize the audio using these discovered units. There are three datasets in total: (1) *Unit Discovery Dataset* provides audio from a variety of speakers and is used to unsupervised acoustic modeling, (2) *Voice Dataset*

provides audio from the targeted speaker and is used for synthesizer modeling and (3) *Parallel Dataset* is intended for finetuning both the sub-systems. We have not utilized the parallel dataset for our observations in this study. The development language is English and the test language is Standard Indonesian. The system is constrained to not use any pre-existing resource or models. To ensure that the model generalizes out of the box, the hyperparameter will be fine-tuned only on the development dataset, and the model will be trained in test language under the same parameters.



(a)

FIGURE 11.2: Spectrograms of original, generated, and converted speech. The source speaker is female while the target speaker is male.

### 11.4.2 Analysis

In this section, we will discuss different design choices in the architecture, including input features and latent space constraints.

#### 11.4.2.1 Acoustic Unit Discovery

Here we analyze the AUD performance of three different models in ZeroSpeech dataset as shown in Table 11.1. We only show the results in English since we don't have ground truth for the Indonesian language.

TABLE 11.1: Performance of different systems in ZeroSpeech

| | English | |
|---|---|---|
| Model | ABX score | bitrate |
| Baseline | **27.46** | 74.5 |
| Three-stage Model | 34.86 | 68.54 |
| VACONDA | 38 | **58.19** |

As in Table 11.1, the VACONDA achieves the best bit rate among three models. With such small number of unit, we could resynthesize and even convert the speech in a very high quality.

#### 11.4.2.2 Speech Resynthesis and Conversion

The proposed model supports synthesizing the same speech in both the same speaker and a different speaker. Here we show a sample in the test dataset of Indonesian language in Figure 11.2. When we feed the decoder with the same speaker identification, the decoder will generate the original audio. Otherwise, it will perform speech convertion. The three audio shares similar structure. However, the converted audio has denser waveform, suggesting it's a different speaker. For the sampled audio, please visit the our website.

## 11.5 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of style information from speech utterance. For this, I employ automatic discovery of acoustic units as the example task. Specifically, we present an approach to automatically discover acoustic-phonetic units from a speech utterance in an unsupervised fashion. We first present an analysis to show that incorporating latent random variables into neural generative models using suitable priors allows us to control what gets encoded into the latent space. Based on this, we employ articulatory features as a discrete prior bank in the latent space and obtain acoustic units that are speaker and language independent. To validate effectiveness of the discovered units, we perform discriminability tests as part of ZeroSpeech Challenge 2019.

# 12

# FLEXIBILITY - De-Entanglement of Content and Style for Emphasis in Text to Speech

In this work, I present our work towards modeling the prosodic variations in generative models for speech. Specifically I present a conditional variational auto-encoder with hierarchical global and local latent variables aimed at disentangling the tonal information from speech. During inference, our model provides flexibility to either sample an embedding or specify desired prosodic schema at the variational layer to generate varying prosody for the input text. To accomplish efficient modeling of prosodic aspects we incorporate inductive bias into the model architecture in the form of fundamental frequency($F_0$). We show that our model outperforms an unbiased hierarchical baseline in terms of control over the generated prosody. Through extensive subjective evaluations in the form of preference tests, we show that our approach also significantly outperforms AuToBI based system that incorporate tone information at the word level.

## 12.1 Introduction

Humans exhibit both coarse as well as fine grained explicit control over how they speak an utterance. This targeted control on speech - often manifested in the form of prosodic constructions - allows us to effectively convey our intent in a conversation. Examples of controlled speech generation include simple prosodic manipulations such as implying specific meaning, highlighting or expressing interest in something as well as various communication strategies

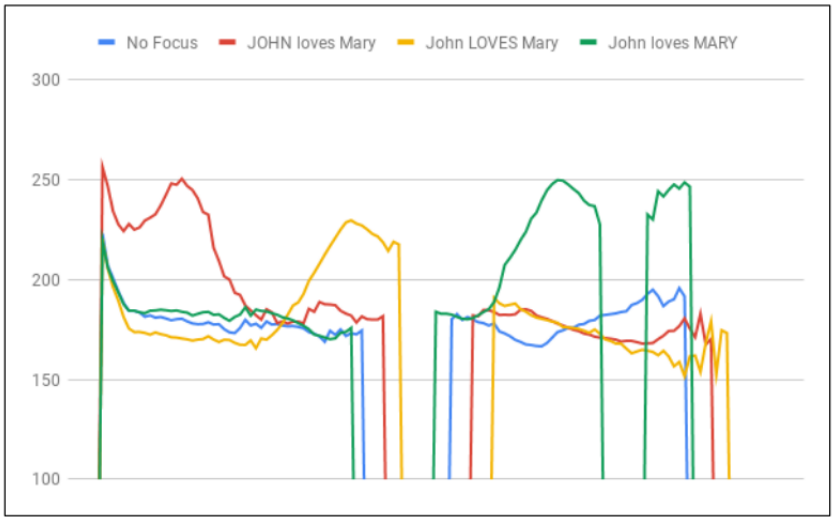



FIGURE 12.1: *Plot of Fundamental Frequency($F_0$) trajectories obtained from generated waves using proposed approach FUE. Variants of the sentence 'John loves Mary' are generated with emphasis on individual words(captialized). The blue trajectory corresponds to $F_0$ when no emphasis was applied to any word. The plot highlights that the proposed approach allows explicit local control at the desired level in the generated speech. We have submitted the generated wavefiles as supplementary material.*

such as contradiction, contrast, complaints or grudging admiration(Ward, 2019). Further, such manipulation in prosody has been shown effective in applications such as Infant Behavior Programs (Morningstar et al., 2019), improving language acquisition(de Carvalho et al., 2019) and promoting rapport(Acosta and Ward, 2011). It seems natural to employ generative models of speech(Wang et al., 2017b; Gibiansky et al., 2017; Ping et al., 2018; Biadsy et al., 2019b) to assist in such scenarios (Wood et al., 2018). However, although there has been tremendous progress in the neural generative models for speech in the context of vocoder fidelity(Prenger et al., 2018b; van den Oord et al., 2016), the notion of controllability in such models is not yet fully evolved. While there have been works towards models aimed at controlling prosody(Hsu et al., 2018c; Wang et al., 2017d), the exerted control is still global or coarse grained in terms of styles of speech(Skerry-Ryan et al., 2018; Wang et al., 2018d), etc. In this work, we propose an approach that allows both global as well as local control over the prosodic variation in the generated speech.

In this work we concern ourselves specifically with Text to Speech (TTS) systems. Typically TTS is formulated as a conditional generative modeling problem. In our approach, we propose to instead formulate it as a conditional variational auto-encoder and incorporate automatically derivable information from speech data into the model architecture. This is motivated by the understanding that the utterances themselves do not always contain all the information needed to comprehend the appropriate prosody information. The missing information is either part of background knowledge about the world - implicit to humans but not annotated in the data - or is provided by accompanying context of the utterance. Formulating the task using variational inference allows us to efficiently capture the distribution of prosody thereby avoiding the averaging effect observed in a typical TTS system due to prosody marginalization. To illustrate

this, consider an example sentence: '*You do not have a pet shark*'. Most prosodic constructions for this sentence involve sarcasm since it is not commonplace to have sharks as pets - world knowledge. Similarly consider the sentence: '*I dont want to be a nun*'. The linguistic unit subject to realization of prosodic stress in this sentence depends on the context information. Finally, consider the example of a TTS system deployed in a screenreader to assist visually impaired students comprehend math equations. Human voice talent would almost certainly place appropriate prosodic cues that help in comprehension of $x^{(y+z)}$ as opposed to $(x^y + z)$. Our formulation allows the model to leverage prosodic information available from the speech signal and capture prosodic distribution.

To accomplish local as well as global prosody control, we incorporate inductive biases into the model architecture in the form of fundamental frequency($F_0$). Specifically, we quantize $F_0$ into multiple bins and constrain the latent space to disentangle these quantized values from acoustics at the level of phonemes. Our model is explained in detail in section 16.2. During inference, the prosody distribution can be utilized to control and generate variability in the output speech. In short, our contributions from this work are: (1) We present EDITH, a hierarchical model that disentangles prosodic features in the form of $F_0$ enabling explicit global as well as local control. (2) We show that EDITH captures reliable representation of local prosody by generating speech with desired variations at the chosen linguistic level.

## 12.2 Related Works

### 12.2.1 Neural Generative Models for Speech

There have been continuous and significant improvements in both the aspects of speech generation - fidelity and flexibility. Approaches such as (Wang et al., 2017b; Gibiansky et al., 2017; Ping et al., 2018; Biadsy et al., 2019b) have demonstrated that Seq2Seq models are capable of reliably learning reasonable associations between the textual and acoustic modalities. These approaches have been utilized in building systems for new languages (Choi et al., 2018) as well as improving the models for existing languages (Lee and Kim, 2019a). Moreover, adaptation of these approaches to various tasks has been investigated (Bollepalli et al., 2018). Auto regressive models such as (van den Oord et al., 2016), flow based models such as (Prenger et al., 2018b) have shown to generate audio that rivals the quality of natural speech. Approaches such as (Taigman et al., 2017c; Taigman et al.) have shown ways to incorporate inductive biases into the generative process. These techniques have been utilized in building highly flexible systems capable of generating different styles(Hsu et al., 2018c; Wang et al., 2017d; Skerry-Ryan et al., 2018; Wang et al., 2018d; Skerry-Ryan et al., 2018) of speech and ability to build voices from noisy(Hsu et al., 2018a) or very minimal data(Chen et al., 2018d). The elegance of such Seq2Seq models comes from the fact that they can be trained without making assumptions based on prior knowledge specific to speech. Therefore, we have employed Seq2Seq based approach as our cornerstone toward building our proposed systems.

### 12.2.2 Prosody

There have been several works depicting the usefulness of prosody in language technologies such as improving rapport(Acosta and Ward, 2011; Shi and Yu, 2018). Therefore, multiple efforts are directed towards both modeling as well as generating prosody in speech synthesis systems. These efforts include both supervised approaches employing manually annotated or automatically derived labels as well as unsupervised approaches aimed at discovering prosody information. While most Seq2Seq approaches aim to model prosody at a coarse level(Hsu et al., 2018a; Wang et al., 2017d; Skerry-Ryan et al., 2018; Wang et al., 2018d), there have been attempts targeting fine grained control as well. In (Wan et al., 2019), authors introduce clockwork hierarchical VAE to predict $F_0$, duration and $C_0$. Our approach of incorporating $F_0$ information at the output of encoder in the form of additional task can be seen similar to this work. However, we use quantized $F_0$s, do not employ clockwork structure in our model and do not explicitly model duration or $C_0$. In (Zhang et al., 2019b), authors employ a Variational AutoEncoder (VAE) to model mel spectrum and derive reference embedding. This embedding is used to model and control prosody. While we use a variant of VAE and perform vector quantization in the latent space to identify appropriate quantization of $F_0$, we directly model $F_0$ and not mel spectrogram. Our work is most similar to (Lee and Kim, 2019b) where authors employ inductive bias in terms of duration and show that it is possible to model and control prosody. In our work, we incorporate inductive bias in the form of $F_0$ and not duration.

### 12.2.3 Disentanglement

In (Chen et al., 2018b), authors decompose Evidence Lower Bound (ELBO) into terms measuring the total correlation between the latent variables. In (Burgess et al., 2018a), authors propose incorporating an additional term referred to as channel capacity to promote disentanglement. Our work is similar to these in that we analyze ELBO to show that it is possible to control what gets disentangled. In (Ansari and Soh, 2018), authors present an approach to accomplish disentanglement by modifying the co-variance matrix of the latent representations. In (Esmaeili et al., 2018a), authors present a generalization of ELBO by factorizing the latent representation into a hierarchy. In (Kim and Mnih, 2018) authors augment ELBO using the density ratio trick and in (Hoffman and Johnson, 2016), authors posit that to improve ELBO we must also improve the marginal KL implying the need for good priors. In (Banijamali et al., 2017) authors show that actively trying to disentangle the causal factors of variation is better than trying to pressurize the model to forget the invariant representations. We take inspiration from these approaches that manipulate the prior distribution to conform to the quantized $F_0 s$.

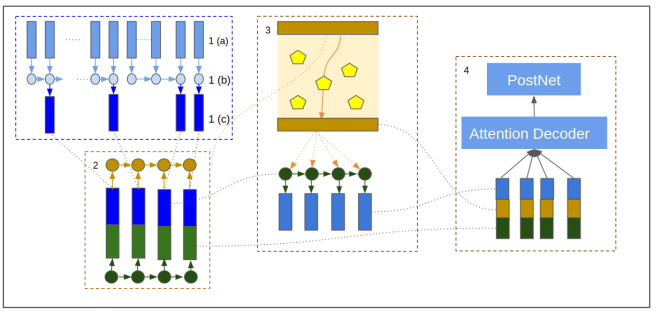



FIGURE 12.2: *Architecture of EDITH. Circles denote LSTM cells, rectangles represent vectors and pentagons represent global latent vectors. (Best viewed in color)*

## 12.3 Emphasis by Disentangling Tonal Heuristics(EDITH)

EDITH learns the joint distribution between pairs of temporal sequences $\{\mathbf{x}, \mathbf{y}\}$ where $\mathbf{x}$ denotes the features and $\mathbf{y}$ denotes the acoustic parameters. Let Ti and To denote the lengths of input and output sequences respectively. Input features $\mathbf{x}$ consist of both linguistic features denoted as $x^{1:Ti}_{linguistic}$ as well as features extracted from the acoustic signal denoted as $x^{1:To}_{acoustic}$. The output features $\mathbf{y}$ consist of linear feature representation $y^{1:To}_{linear}$ as well as mel features $y^{1:To}_{mel}$. It has to be noted that $x^{1:To}_{acoustic}$ = $y^{1:To}_{mel}$. To efficiently model varying prosody and prevent the averaging effect, we incorporate a variational layer. Therefore, EDITH is a conditional variational auto-encoder. During inference, we discard the encoder part of our model. Our model can be summarized by the following set of equations:

$$\begin{aligned} encoded &= \mathbf{H}^{Encoder}(x_{linguistic}, x_{acoustic}) \\ z_g, z_l &= \mathbf{VI}(encoded) \\ \hat{y}_{mel} &= \mathbf{H}^{Decoder}(x_{linguistic}, z_g, z_l) \\ \hat{y}_{linear} &= \mathbf{H}^{postnet}(\hat{y}_{mel}) \end{aligned} \tag{12.1}$$

Design of our encoder is inpired by the encoder from (Wan et al., 2019). We use clockwork hierarchical LSTM to encode $x^{1:Ti}_{linguistic}$ and $x^{1:To}_{acoustic}$. However, our models are clocked at the rate of phones as opposed to syllables. In addition, we do not incorporate any features from word or sentence levels in our encoder to keep the architecture compact. Our variational layer is derived from (van den Oord et al., 2017b) and is employed to generate global and local latent variables $z_g$, $z_l$ respectively. Our decoder is similar to a typical attention based acoustic decoder(Wang et al., 2017b) and includes a postnet. While similar in formulation, EDITH has an important difference from (Wan et al., 2019) in that our local latent variables follow the rate of input as opposed to output as in (Wan et al., 2019). This allows us to exercise more control over the generated prosodic variations.

**Optimization and Learning**: $x_{acoustic}$ is passed through phone rate LSTM. This is shown as block 1 in figure 12.2. $x_{linguistic}$ is passed through phone LSTM. The representations are

concatenated and passed through EDITH Encoder. This is shown as block 2. Outputs from the encoder are passed through the variational layer where vector quantization is performed to pick the most suitable global latent prosodic vector. Conditioned on encoder outputs and the global latent prosodic vector, we predict Ti local prosodic vectors corresponding to predicted local prosodic features. We constrain the local latent variables to correspond to quantized $F_0$ by modeling their prediction as a classification task. These local latent variables thus capture the local variations in prosody while global latent variable is reserved for capturing sentence level variations. Ground truth quantized values for classification are obtained by selecting the maximum bin within the duration of phoneme. This is shown as block 3 in the figure. We then employ dot product attention in our decoder. $y_{mel}$ is generated by decoder conditioned on local, global latent variables and the encoded $x_{linguistic}$. A postnet is employed to generate $y_{linear}$ conditioned on $y_{mel}$. EDITH is optimized to minimize two *L1* losses one each for $y_{mel}$ and $y_{linear}$ and one classification loss for local latent variables. Additionally, to train the vector quantization layer, we minimize encoder commitment loss for $z_g$ and vector quantization loss following (van den Oord et al., 2017b) for both $z_g$ and $z_l$. This can be expressed as below:

$$\begin{aligned} L = \lambda_{linear} \sum_{t=0}^{To} \|y^t_{linear} - \hat{y}^t_{linear}\| \\ + \lambda_{mel} \sum_{t=0}^{To} \|y^t_{mel} - \hat{y}^t_{mel}\| \\ + \lambda_{qF_0} \sum_{t=0}^{Ti} Div(qF_0, q\hat{F}_0) + \lambda_e L_e + L_{VQ} \end{aligned} \tag{12.2}$$

### 12.3.1 Model Interpretation

This approach can be interpreted as VQVAE(van den Oord et al., 2017b). It can also be seen as GST(Wang et al., 2018d) based encoding but our approach has two differences:(1) We do not use a different encoder for spectral information and (2) We explicitly constrain the latent classes to correspond to the quantized $F_0$s. We divide the model into individual blocks or modules. Therefore, it can be seen as an extension to Neural Module Networks(Andreas et al., 2016b). In (Wan et al., 2019), authors introduce clockwork hierarchical VAE to predict $F_0$, duration and $C_0$. Our approach of incorporating $F_0$ information at the output of encoder in the form of additional task can be seen similar to this work. However, we use quantized $F_0$s, do not employ clockwork structure in our model and do not explicitly model duration or $C_0$.

## 12.4 Experimental Setup

**Data**: We have used data from LJSpeech dataset(Ito et al., 2017) to build our systems. We have used all of the 13100 sentences. The text was normalized manually to convert non standard

forms (for ex. 1993) to written forms (nineteen ninety three).

**Baselines**: Our acoustic model is based on Tacotron(Wang et al., 2017b) Seq2Seq speech synthesis system is built using PyTorch(Paszke et al., 2019). We have not performed masking of padded frames as is typically done in Seq2Seq models. We found that not masking helps model better predict end of sentence as mentioned in (Wang et al., 2017b). Since adjacent frames seem to be correlated, our decoder predicts 5 frames per timestep. Our model has three deviations from the original implementation: (1) Phones are used as the input instead of characters. (2) CBHG module in the encoder and postnet has been replaced with with three LSTM layers. (3) We use all the predicted frames at a time step as input to the decoder(as opposed to only the last time step) while predicting the next frames. We have used a batch size of 64 to train the baseline model. To enable control of prosody, we employ quantized $F_0$ values as additional inputs to this baseline model. For this, we first extract $F_0$ values for the dataset and quantize them into multiple bins each spanning 25 *Hz* without any overlap. These quantized $F_0$ values are embedded and added as additional inputs to the baseline model. In other words, this is a conditional generative model with phones and quantized $F_0$s as inputs. Additionally, we also build a model that uses word level prosodic features extracted using AuToBI(Rosenberg, 2010). We refer to this system as **AuToBI**.

**EDITH Hyperparameters**: The encoders of both $x_{acoustic}$ and $x_{linguistic}$ are realized using bidirectional LSTMs. We have used 256 as the hidden dimensions for both these encoders. Both our global and local latent variables are of 256 dimensions. We employ 10 global latent classes. The network to predict local latent variables is implemented using bidirectional LSTMs that takes 512 dimensional input and outputs 256 dimensional vectors. Encoder weight $\lambda_e$ was linearly increased to 0.2 till 10K timesteps and remained constant after that. For quantization of $F_0$, we have followed the same procedure as in Baseline. 25 Hz was chosen as the size of bin. This effectively resulted in a total of 14 bins and thus 14 local latent classes. After every update step, we normalize the local latent variables by the norm. Since these classes correspond to ordinal data in terms of quantized $F_0$s, we believe that normalizing places the vectors on a unit circle.

**SubUtterance Models**: Long utterances present in audiobooks are rich in prosodic variations but also lead to computational overhead in terms of processing speed. Therefore, we have built systems that have access to only part of the utterance by selecting aligned segments of text and acoustics within a full sentence. We note that such an approach is already used for vocoding: Typical vocoders the authors are aware of are trained using aligned chunks of acoustic vectors and corresponding speech samples as opposed to full utterances. Encouraged by this, we build sub utterance based models for both baseline as well as proposed approach. To distinguish from the full sentence models, we refer to these systems as Sub Utterance Baseline(*SUB*) and Sub Utterance EDITH(*SUE*) while referring to the full sentence models as Full Utterance Baseline(*FUB*) and Full Utterance EDITH(*FUE*) respectively.

**Evaluation**: Evaluation was performed in the form of listening tests using (Parlikar, 2012b). We have conducted two types of listening tests: (1) Rating the naturalness in terms of Mean

TABLE 12.1: *Results from Preference and MOS Tests for Emphasis generation. The entries for the preference portion(columns 2 through 6)indicate preference values obtained by the systems in the first column against every other system in the subsequent columns.*

| Config | $FUB$ | $FUE$ | $SUB$ | $SUE$ | AUToBI | MOS |
|---|---|---|---|---|---|---|
| $FUB$ | - | 92 | 396 | 363 | **441** | 4.0 |
| $FUE$(ours) | **345** | - | **424** | **378** | **477** | 4.0 |
| $SUB$ | 91 | 86 | - | 235 | **278** | 3.4 |
| $SUE$(ours) | 64 | 86 | 243 | - | 227 | 3.6 |
| AUToBI | 47 | 19 | 219 | 256 | - | 3.9 |

Opinion Score (MOS) on a scale of 1(least natural) to 5(highly natural) and (2) ABX Preference test on Emphasis where the users need to mention their preference towards either of the systems or state that they prefer neither. For the preference evaluation we have manually curated 50 sentences where the meaning was implied based on prosody. Participants were shown the entire sentence and its implication in parenthesis. An example sentence from our testset is '*It looks like a starfish* (but it really is not).' Every system was used to generate this test set[1]. For baseline and proposed approaches, the phonemes to be emphasized are rendered with embedding vector corresponding to bin 12 while others are rendered with bin 8 The participants are to mention their preference to the system that faithfully generates prosody in line with the information in parenthesis. We had 25 listeners and each participant rated 20 random sentences giving us a total of 500 ratings per pair of systems.

**Discussion**: The preference evaluation results for the proposed approaches are presented in table 12.1. We have excluded the *No Preference* values from this table for brevity. However, they can be estimated based on the values in the table. The full utterance based systems seem to outperform sub utternace as well as AuToBI based systems consistently. Within the full sentence systems, our proposed approach(*FUE*) outperforms the baseline conditional generative model(*FUB*). A sample output generated by conditioning the local latent variables to emphasize individual linguistic units(words) from our approach can be examined in figure 12.1. An informal listening test in the scenarios where full sentence models were not preferred revealed an interesting finding: All these scenarios were when the emphasized word was the first in the sentence. We hypothesize that this might be due to the canonical word order(**SVO**) in English. One approach to handle this could be to incorporate a suitable weighting to consider this effect and we plan to investigate this further. The sub utterance based approaches seem to match the performance of AUToBI systems while clearly under performing their full utterance counterparts. Informal listening evaluations revealed that the sub utterance models seem to have repetition of phoneme units within the generated sentence. We attribute this to the errors in alignment and phoneme boundary estimation and plan to investigate approaches to circumvent this behavior in future work.

[1]in the mentioned example, the systems generated just the part '*It looks like a starfish*' and not the part in parenthesis

## 12.5 Conclusion

In this chapter, I have presented a case study demonstrating De-Entanglement of content as well as style information from speech utterance. Specifically, we have proposed an approach to obtain local and fine grained control over prosody in neural generative models for speech. For this we quantize fundamental frequency, which is highly correlated with prosody information, into multiple bins. We infer this information employing hierarchical global and local latent variables in the model architecture. We show that our approach generates appropriate emphasis at word level and significantly outperforms AuToBI in terms of flexibility.

# Part IV

# Explainability

In this part I will provide an overview of experiments addressing the challenge of explainability. While explainability encompasses several sub topics, I restrict myself to a specific view where the objective is to *reason* about the model behavior. I posit that explainable speech technologies should accomplish the following goals:

- (a) Reasonable Understanding of internal mechanisms in the model.
- (b) Demonstrable Utility of the model for downstream applications.

In this part, I will describe my motivations for selecting these two aspects and then present experiments towards them.

## What and How

### What

I borrow the ideology behind (a) *Reasonable Understanding* from the Explainability draft released by NIST(Phillips et al., 2020). Within De-Entanglement, I propose to employ a two tier architecture - an architecture with multiple models where one of the models attempts to justify the predictions by the overall architecture. Specifically, I propose to explore a generative model that inherently employs **discrete latent** variables. The choice of discrete variables is inspired by the following:

- *Physics* - Brain Greene in his book *Until the end of Time*(Greene, 2020) describes the property of consciousness to be highly integrated and highly differentiated following *integrated information theory* using the example of a Red Ferrari. In his words,

  *... focusing more narrowly on the car's color, note that your experience is decidedly not one of a colorless Ferrari that your mind subsequently paints red. Nor is it of an abstract red environment that your mind subsequently shapes into a Ferrari. Although shape information and color information activate different parts of the visual cortex, your conscious experience of the Ferrari's shape and color are inseparable. You experience them as one.*

  I posit that to realize such highly integrated as well as highly differentiated explanation, it is necessary to incorporate discrete variables into the explanations. Greene further implies that humans invented language as a means of such discretization (and abstracting away) of the details in favour of a big picture view. I posit that such defining and developing abstractions within the context of interpretibility can help comprehending model behavior. For instance, consider a model that learns personality traits from the speech signal as the style information and hypothesizes pseudo phonemes as the content information by design. These learnt representations can be used to interpret the model functionality during inference.

I borrow the ideology behind (b) *Demonstrable Utility* from *Upanishads* of Hindu Mythology. Within De-Entanglement, I propose to employ an architecture that discovers latent units that can be utilized for other downstream tasks in low and under resourced scenarios. The choice of utility is inspired by the following:

- *Upanishads* - Answering a question about what 'intelligence' is, teacher recites the following shloka, in Kenopanishad.

  *śrotrasya śrotra manaso mano yadvāco ha vāca sa u*
  *prāasya prāaścakuaścaku atimucya dhīrā pretyāsmāllokādamtā bhavanti*

  The literal English translation of the verse reads 'It is the ear of the ear, mind of the mind, tongue of the tongue, and also life of the life and eye of the eye.' The teacher here, however, implies a subtext which reads that intelligence is that which enables the ear to hear, eye to see, mind to think, tongue to taste and life to live. In other words, the abstract concept of Intelligence is defined by its utility to perform various sensory functions.

  I formulate an explainable system in a similar fashion, where once the system is explainable, we can employ it to perform various 'utilities'. In my dissertation, I show how the representations learnt in an explainable system can be employed to build systems in low and under resourced scenarios.

### Tasks

I propose to experiment with the proposed architecture in the context of two speech based tasks:

- (1) *Identification of language from acoustics* - Identifying the language of a given speech utterance can benefit the downstream speech and natural language models. I begin this set of experiments with detecting whether is given speech utterance is code mixed or not. For this I to employ the code mixed speech data released by Microsoft Research India as part of Code Switching Workshop accompanying Interspeech 2020. The experimental details can be found in chapter 15.

- (2) *Identification of intents from acoustics* - Understanding the intents from acoustics alone can help rapidly build speech technologies in low resource communities. For this, I employ both natural and synthesized speech as inputs. More details about the experiments can be found in chapter 13.

# 13

# EXPLAINABILITY - Identification of Intents using discovered discrete latent units

Tremendous progress in speech and language processing has brought language technologies closer to daily human life. Voice technology has the potential to act as a horizontal enabling layer across all aspects of digitization. It is especially beneficial to rural communities in scenarios like a pandemic. In this work we present our initial exploratory work towards one such direction - building voice enabled banking services for multilingual societies. Speech interaction for typical banking transactions in multilingual communities involves the presence of filled pauses and is characterized by Code Mixing. Code Mixing is a phenomenon where lexical items from one language are embedded in the utterance of another. Therefore speech systems deployed for banking applications should be able to process such content. In our work we investigate various training strategies for building speech based intent recognition systems. We present our results using a Naive Bayes classifier on approximate acoustic phone units using the Allosaurus library (Li et al., 2020).

## 13.1 Case Study

### 13.1.1 Data Collection Tool

We have created a dataset of multilingual queries using a dummy banking app (ban). It involves a two stage setup. In the first stage, speech data is crowdsourced from fluent Hindi speakers.

| Model | # Unique N-grams | Test Accuracy |
|---|---|---|
| Unigram | 38 | 0.56 |
| Bigram | 292 | 0.48 |
| Trigram | 543 | 0.17 |

Table 13.1: N-gram classification accuracy with Add-1 smoothing

| Model | Delta | Test Accuracy |
|---|---|---|
| Unigram | 5 | 0.69 |
| Bigram | 1 | 0.61 |
| Trigram | 1 | 0.30 |
| Combination | (5, 1, 1) | **0.83** |

Table 13.2: Classification accuracy with Absolute Discounting

Each speaker plays the role of a 'user' and interacts with the automated banking system. This resulted in 100+ task-based dialogs in Hinglish, with five distinct intents. In the second stage, each speaker is asked to acoustically translate their interaction into another language. This way, we obtain pseudo parallel data in two languages from the same speaker. We believe that creating a pseudo parallel dataset will allow us to design semi supervised approaches in the future. Our dataset currently has speech data from six Indian languages - Hindi, Marathi, Gujarati, Punjabi, Telugu and Bhojpuri.

### 13.1.2 Methodology and Results

We chose spoken intent classification as the first task. For this task, we have 25 utterances containing speech samples of 11 people, out of which 4 were female. We have 5 intents in the dataset - **Send Money** (11 utterances), **Check Balance** (9 utterances), **Check Last Transaction** (3 utterances), **Withdraw Money** and **Deposit Money** (1 utterance each) .

We first convert audio into phones using the Allosaurus library(Li et al., 2020). These phones are then used for intent classification. We employ Naive Bayes' Classifier with add-1 smoothing and absolute discounting. We use cross validation where we leave out 2 audio samples for testing. The intents *Withdraw Money* and *Deposit Money* were not used for testing as we only have one sample for both, but are included in the training set. Thus the testing is only done for three intents, but an intent could be classified into any of the 5 classes. The results for Naive Bayes Classifier for add-1 smoothing are shown in Table 13.1.

It can be observed that the accuracy decreases with increasing N for the different N-gram models. We hypothesize that this happens due to the relatively small size of the dataset in our initial exploratory work. We posit the distribution characteristics to be more uniform across N-grams in our future experiments with full dataset.

| Language | Number of Utterances |
|---|---|
| ordering pizza | 711 |
| auto-repair appointment | 484 |
| order ride service | 450 |
| order movie tickets | 549 |
| order coffee | 292 |
| restaurant reservations | 757 |

TABLE 13.3: Class distribution for the Indic dataset.

Further we have also performed absolute discounting with different delta values for each of the Ngram models. The test accuracies improved significantly. We also combined unigram, bigram and trigram models with equal weights with their respective best performing delta values, which gave us the best result as shown in Table 13.2.

## 13.2 DATASETS

We study the performance of our acoustics based intent recognition system for two language families - Indic Languages and Romance Languages. For each family we use a different dataset and each language family has a different intent recognition task.

### 13.2.1 Dataset for Indic Languages

We use Google's Taskmaster-1 Dataset (Byrne et al., 2019) for Indic Languages which contains data for user interactions with an autonomous dialogue system collected using the Wizard of Oz methodology (Hanington and Martin, 2012). The user dialogues are a written transcripts of the conversations in English. The dataset contains labelled intents and slots for the conversation. We extract the sentence responsible for the labelled intent from the dataset and create an intent recognition dataset. We obtain 3243 utterances in total distributed amongst 6 intents as shown in Table 14.1.

After creating the intent classification dataset, we translate the transcripts in Engish into four Indic languages - Hindi, Gujarati, Bengali and Marathi, using the Google Translate API. The translated text was used to synthesize audio using the Google Text-To-Speech API for Hindi, Gujarati and Bengali. CLUSTERGEN (Black and Muthukumar, 2015) was used synthesizing for Marathi voice. The voice quality of Marathi is much worse when compared to the other voices generated from Google's API. The dataset in each language contains two voices - one male and one female. The audios are then passes into Allosaurus (Li et al., 2020) to discover phonetic units and create a phonetic transcription of the audio.

| Intents | Number of Utterances |
|---|---|
| Monday | 743 |
| Tuesday | 718 |
| Wednesday | 757 |
| Thursday | 738 |
| Friday | 763 |
| Saturday | 779 |
| Sunday | 806 |

TABLE 13.4: Class distribution for the Romance dataset.

### 13.2.2 Dataset for Romance Languages

To work with Romance languages, we create an intent recognition dataset from the MultiWoz dataset (Budzianowski et al., 2018). The dataset contains a large number of dialogues between humans and robots where each utterance is associated with a json object containing the conversational context. The context has rich information about the intent of humans. The largest context class in the dataset is about the reservation day whose distributions are shown in the Table 14.3. This class is used as our Romance languages dataset. This dataset is then prepared in a similar way as done for the Indic language, where we translate the original English utterances into 4 different Romance languages - Italian, Portuguese, Romanian and Spanish. The translated text is synthesized with the Google TTS engine, and then transcribed into phonetic units with Allosaurus (Li et al., 2020).

## 13.3 MODELS

A block diagram depicting our acoustics based intent recognition system utilizing a phonetic transcription is shown in Figure 13.1. The input audio is directly fed into a system that can generate hypothesized phonetic units. For our work, we use the Allosaurus library (Li et al., 2020) which is a nearly-universal phone recognition system. For this work, we employ the language dependent phones, which basically means we're providing an identifier to Allosaurus for audio language. The phonetic transcription is then sent to an intent classifier that does the classification purely based on the generated sequence of phones. Such very simple systems can be used to build powerful tools, especially for low resource languages, as shown in (Gupta et al., 2020b).


FIGURE 13.1: Block Diagram showing a general acoustics based intent recognition system.

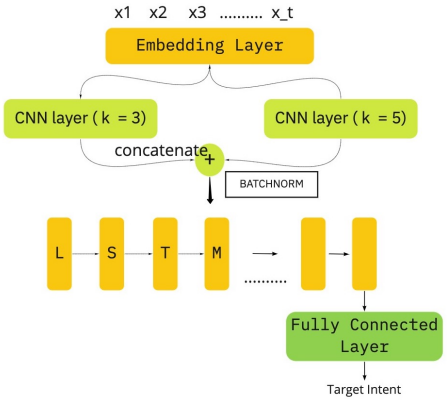



FIGURE 13.2: Block Diagram depicting the architecture of our proposed neural network.

We use a Naive Bayes classifier as our baseline with add-1 smoothing and absolute discounting. We also propose a neural network architecture shown in Figure 13.2 to compare with the baseline results. The architecture is based on LSTMs (long-short term memory) (Hochreiter and Schmidhuber, 1997) for modeling sequential information where the contextual information is encoded using CNNs (convolutional neural network).

The input to the network is a sequence of phones $\mathbf{x} = x_1, x_2, \ldots x_t$, where each phonetic unit is passed to a 128 dimensional embedding vector. The embedding layer converts the input sequence into a dense vector representation which is then sent to two 1-d CNN layers of kernel size k = 3,5. The CNN layers have 128 filters and capture trigram and 5-gram features from the phonetic transcription. The outputs of each of the CNN layers are concatenated to create a 256 dimension long embedding vector where each embedding vector now has contextual information encoded in it.

The concatenated embeddings are passed through the LSTM layer consisting of 128 neurons. The hidden state of the LSTM layer at the final time step is sent to a linear layer for intent classification.

## 13.4 EXPERIMENTS

We test our acoustics based intent recognition system for two sets of languages across two different language families - Indic and Romance language families. We perform monolingual and multilingual training for both baseline and our proposed neural network architecture and test the model performance for multiple languages.

### 13.4.1 Monolingual Training Results

In this section we present results for intent classification architectures trained on a single language. Table 14.2 presents the classification results for Indic languages and Table 13.7 for

| Config | Hin | Guj | Mar | Ben |
|---|---|---|---|---|
| **Hin** | **92.0**(89.3) | 54.7(59.7) | 43.7(36.7) | 54.3(45.3) |
| **Guj** | 52.3(50.3) | **93.3**(91.7) | 52.0(47.0) | 63.0(39.3) |
| **Mar** | 52.0(35.0) | 66.3(49.7) | **87.7**(84.3) | 58.0(37.0) |
| **Ben** | 48.0(41.7) | 54.7(38.3) | 45.7(31.3) | **95.0**(93.0) |

TABLE 13.5: Classification Accuracy for monolingual training for Indic Languages - Hindi (Hin), Gujarati (Guj), Marathi (Mar) and Bengali (Ben). The numbers in the bracket are the baseline results using a Naive Bayes classifier.

| Config | Hin | Guj | Mar | Ben |
|---|---|---|---|---|
| **HGM** | 85.3(84.7) | 90.3(86) | 75.6(78.3) | **80.7(58.3)** |
| **HGB** | 87.3(84) | 90.0(84) | **61.7(54.3)** | 90.3(89.3) |
| **HMB** | 84.3(88.7) | **62(65.4)** | 80.7(76.6) | 88.3(88) |
| **GMB** | **65.3(63.7)** | 86.7(84.7) | 83.0(80) | 92.0(89.7) |

TABLE 13.6: Average Classification Accuracy for a multilingually trained model. The languages in bold are the languages that are not present in the train set. The numbers in the bracket are the baseline results using a Naive Bayes classifier.

Romance languages. The diagonal elements in the tables show the classification accuracy for training and testing performed on the same language. The numbers in the bracket show performance with the baseline (Naive Bayes) classifier. We see that our proposed neural network architecture improves on our baseline significantly.

Cross-lingual testing results for monolingually trained classification models are also shown in Tables 14.2 and 13.7. The performance is relatively poor when the classification model is trained on only one language due to minimal cross-lingual transfer. Language pairs for linguistically similar languages show higher performance. This can be seen for language pairs Hindi-Gujarati and Gujarati-Marathi in Indic language family and pairs Italian-Portuguese and Italian-Spanish in the Romance language family. These language pairs are also geographically close. The cross lingual results are in general better for the Indic Dataset when compared to the Romance dataset. We believe this is because all Indic languages have some amount of code mixing within them. Therefore, there is a larger cross-lingual transfer of features between any pair of languages in the Indic language family when compared to the Romance language.

### 13.4.2 Multilingual Training Results

With the aim of improving performance on a language not present in our training set and simulating a zero resource scenario, we train a multilingual model. The training set size is kept the same and the exact same train-test split is used for accuracy scores as used for monolingual results. Let T = $[L_1, L_2, \ldots L_n]$ be the set of languages we use to train the classifier. Then the training set is divided randomly and equally amongst the '$n$' languages present in the training set.

| Config | Ita | Por | Ron | Spa |
|---|---|---|---|---|
| **Ita** | **88.6**(82.4) | 27.6(24.2) | 28.6(33.9) | 39.0(32.5) |
| **Por** | 30.6(23.3) | **88.6**(74.6) | 28.2(22.2) | 36.0(26.2) |
| **Ron** | 32.6(31.6) | 29.6(17.4) | **86.6**(76.5) | 46.0(33.3) |
| **Spa** | 46.1(35.8) | 43.9(40.2) | 35.4(33.9) | **88.7**(83.3) |

TABLE 13.7: Classification Accuracy for monolingual training for Romance Languages - Italian (Ita), Portuguese (Por), Romanian (Ron and Spanish (Spa). The numbers in the bracket are the baseline results using a Naive Bayes classifier.

| Conig | Ita(I) | Por(P) | Ron(R) | Spa(S) |
|---|---|---|---|---|
| **IPR** | 87.0(78.8) | 88.9(62.1) | 85.4(69.1) | **60.4**(43.9) |
| **IPS** | 88.6(77.2) | 88.9(62.9) | **37.4**(40.7) | 88.9(80.4) |
| **IRS** | 88.1(88.2) | **41.3**(30.0) | 86.6(73.2) | 88.7(80.0) |
| **PRS** | **50.3**(40.9) | 87.8(59.9) | 84.9(69.1) | 88.3(79.9) |

TABLE 13.8: Classification Accuracy for a multilingually trained model. The languages in bold are the languages that are not present in the train set. The numbers in the bracket are the baseline results using a Naive Bayes classifier.

The results for multilingual training can be seen in Table 13.6 and Table 13.8 when trained on $n = 3$ languages. The results in bold are for the language not in the training set. The numbers in the bracket show performance with the baseline (Naive Bayes) classifier. We see that our proposed neural network architecture improves on the baseline results significantly in almost all cases. The power of multilingual training becomes apparent when we look at the performance on a language not present in T. We find that a multilingual classifier always performs better on an unknown language $L_u \notin T$ when compared to zero-shot transfer by monolingual model without significant performance loss in individual languages. Its important to consider that we haven't augmented the data in any form, thus the multilingual model see far fewer example of a specific language than monolingual models.

The results in Table 13.6 and Table 13.8 show that there is larger amount of cross-lingual transfer when the model is trained on many languages from the same language family. In a practical scenario, this means that a deployed multilingual model is more likely to generalize better to an unknown language or variation in dialect than a monolingual model. This is especially useful for the case of low-resource languages for which it's hard to collect any training data. We illustrate this point by taking the example of Bengali in the Indic language family. There are four models across Tables 14.2 and 13.6 that do not have Bengali in the training set. These are the models not highlighted in the Bengali column in Table 14.2 and the highlight value in the Bengali column in Table 13.6. The multilingual model performs best out of these four models. This is also true for Spanish in Tables 13.7 and 13.8.

The performance for an unknown language $L_u \notin T$ can further be improved by injecting a very small amount of data for $L_u$ in the training set. We added training data for language $L_u$ in increments of a ratio of 0.05 of the training set as shown in Figure 13.3. We see that introducing even the slightest amount of training data for the unknown language increases its performance

significantly while not affecting the performance of the other languages. Figure 13.3 shows an increase in performance of about 9% for Marathi, 14% for Hindi and 17% for Gujarati only by an injection of data 5% the size of training dataset.

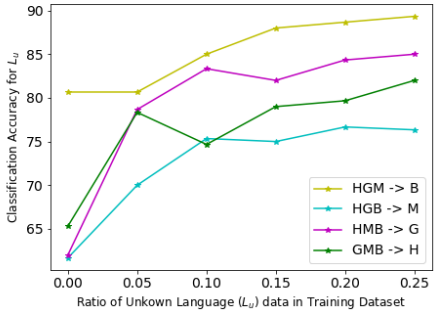



FIGURE 13.3: Plot showing performance of a multilingual intent classification model for when data for a language is injected into the training set in increments of ratio of 0.05 for Indic languages. For example, HGM -> B represents a model trained on Hindi, Gujarati, Marathi and we're checking the increase in perfromance on Bengali by injecting Bengali data into the training dataset.

## 13.5 DISCUSSION

We present a novel approach for intent recognition in low resourced languages with experiments on two different language families. It was shown that zero-shot performance for a language not in the training set of the model but still within the language family can be improved with multilingual training. This helps in maximal cross-lingual transfer between languages that are linguistically and geographically closer to each other. We also found that performance for such a language not in the training set can be improved significantly by introducing a minimal amount of training data.

Our present work was based on synthesized data due to the absence of enough natural speech datasets for intent recognition for low resource languages. Future work can include corroboration of our results with natural speech. The synthesized speech also had little speaker variation in terms of speaker style or prosody though we did include variation in speaker gender.

## 13.6 CONCLUSION

We present a novel acoustics based intent recognition system that classifies intents from phonetic transcripts generated using a (nearly-)universal phone recognizer, bypassing the need to build language specific ASR. We also show that multilingual training within same language families produce better zero shot transfer within same the language family when compared to monolingual models.

# 14

# EXPLAINABILITY - Identification of Intents and slots

Intent Recognition and Slot Identification are crucial components in spoken language understanding (SLU) systems. In this work, we present a novel approach towards both these tasks in the context of low-resourced and unwritten languages. We use an acoustic based SLU system that converts speech to its phonetic transcription using a universal phone recognition system. We build a word-free natural language understanding module that does intent recognition and slot identification from these phonetic transcription. Our proposed SLU system performs competitively for resource rich scenarios and significantly outperforms existing approaches as the amount of available data reduces. We train both recurrent and transformer based neural networks and test our system on five natural speech datasets in five different languages. We observe more than 10% improvement for intent classification in Tamil and more than 5% improvement for intent classification in Sinhala. Additionally, we present a novel approach towards unsupervised slot identification using normalized attention scores. This approach can be used for unsupervised slot labelling, data augmentation and to generate data for a new slot in a one-shot way with only one speech recording.

## 14.1 Introduction

Spoken dialog systems are slowly integrating themselves in everyday human lives, being used for various applications that include accessing information, doing transactions, tutoring and

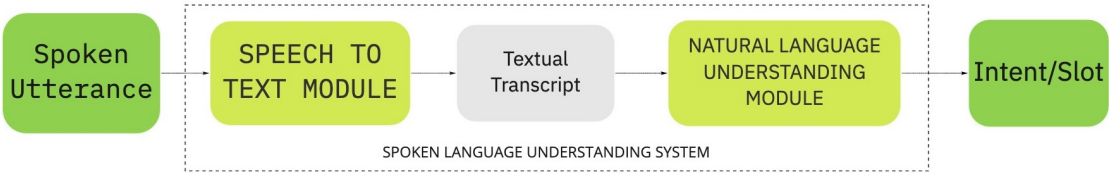



FIGURE 14.1: Block diagram of a typical spoken language understanding system

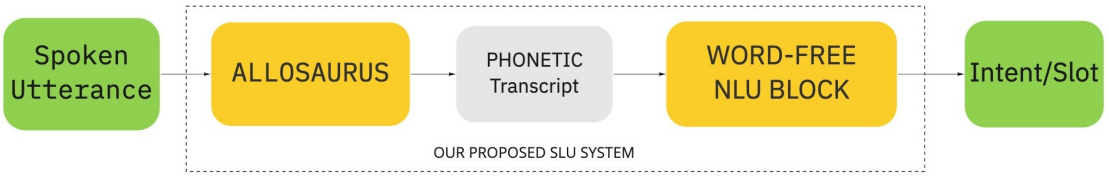



FIGURE 14.2: Block diagram representing our proposed SLU system.

FIGURE 14.3: Diagrammatic description of a typical SLU system and our proposed SLU system.

entertainment. Speech is presumably the most natural form of interaction for humans. Spoken dialog systems not only create a very natural interface for humans to interact with technology, but also overcome the barriers posed by a written interface. Thus, access to technology is not restricted by literacy and can also be done in unwritten languages. Currently, spoken dialog systems are available only in a limited number of languages. A major bottleneck in extending these systems to other low-resourced, local and unwritten languages is the lack of availability annotated data in these languages.

Spoken language understanding (SLU) systems are fundamental building blocks of spoken dialog systems. A typical SLU pipeline, shown in Figure 14.1, comprises of a Speech-to-Text (STT) module followed by a Natural Language Understanding (NLU) module. STT modules convert speech to textual transcriptions and NLU modules perform downstream tasks like intent recognition and slot filling from the transcripts obtained. Creating language specific automatic speech recognition (ASR) modules for each language requires a large amount labelled data, which is usually not available for most languages. Language specific ASR systems thus form a bottleneck for creating SLU systems for low-resourced languages.

In this work, we present a novel acoustics based SLU system where we bypass the need to create a language specific ASR system. A block diagram representing our proposed system is shown in Figure 14.2. We replace language specific STT modules with Allosaurus (Li et al., 2020), which is a universal phone recognition system that creates phonetic transcription of input speech. We then create language and task specific, word-free, natural language understanding modules that perform NLU tasks like intent recognition and slot filling from phonetic transcriptions.

In this work, we show that our proposed SLU system performs competitively with current state of the art SLU systems in high resource setting, and gets better as the amount of data available to the system reduces. We train both recurrent and transformer based neural networks and compare their performance for the task of intent classification. We work with natural speech

datasets in five languages - English, Belgian Dutch, Mandarin, Sinhala and Tamil. Our system improves on the state of the art intent classification accuracy by approximately 5% for Sinhala and 11% for Tamil in low resource settings. We also propose an unsupervised slot value identification algorithm based on the self-attention mechanism. This enables one-shot data generation for a new slot value, where only one speech utterance is needed to generate new data.

## 14.2 Related Work

Spoken language understanding (SLU) systems are a vital component of spoken dialog systems. These systems are responsible for understanding the meaning of a spoken utterance. Doing so requires identifying speaker intent, a task which sometimes requires slot filling. Current research in spoken language understanding is moving towards creating End-to-End (E2E) SLU systems (Qian et al., 2017) (Serdyuk et al., 2018) (Chen et al., 2018c) which have various advantages over conventional SLU systems (Lugosch et al., 2019). To aid development in SLU, various speech to user intent datasets have been created in different languages including English (Lugosch et al., 2019) (Wu et al., 2020) (Hemphill et al., 1990) (Saade et al., 2018), Sinhala (Buddhika et al., 2018) (Karunanayake et al., 2019c), Tamil (Karunanayake et al., 2019c), Belgian Dutch (Renkens et al., 2014) (Renkens et al., 2018), Mandarin (Zhu et al., 2019) and French (Saade et al., 2018). For our work, we choose English, Sinhala, Tamil, Belgian Dutch and Mandarin datasets.

In low-resourced scenarios, building language specific ASR systems is not viable. In previous work, NLU modules have been built on top of outputs of an English ASR system, for example, using the softmax outputs of DeepSpeech (Hannun et al., 2014) for Sinhala and Tamil. DeepSpeech is a character level model and the softmax outputs corresponding to the model vocabulary were used as inputs to the intent classification model (Karunanayake et al., 2019c). Softmax outputs of an English phoneme recognition system (Lugosch et al., 2019) have also been used to build intent recognition systems (Karunanayake et al., 2019b) for Sinhala and Tamil. MFCC features of input speech have also been used for intent classification in Sinhala (Buddhika et al., 2018).

In our work, we build a unique natural language understanding module for SLU systems based on phonetic transcriptions of audio. These phonetic transcriptions were obtained from Allosaurus (Li et al., 2020), a universal phone recognition system that gives language and speaker independent phonetic transcriptions of input audio. These transcriptions are finer grained when compared to a language specific phonemic transcription. This can be seen in the experiments sections where using only the top-1 prediction made by Allosaurus improves the performance on Sinhala and Tamil, which previously used the entire softmax vector of an English ASR system. The advantage of using Allosaurus to generate phonetic transcriptions are manifold. Allosaurus is trained to perform universal phone recognition, and is not a language specific model. This means the phonetic transcriptions encode finer grained information

| Language | Utts | Intents | Speakers | PT Size |
|---|---|---|---|---|
| English (Lugosch et al., 2019) | 30,043 | 31 | 97 | 100 (Panayotov et al., 2015) |
| Sinhala (Karunanayake et al., 2019d) | 7624 | 6 | 215 | 17 (Kjartansson et al., 2018) |
| Tamil (Karunanayake et al., 2019d) | 400 | 6 | 40 | 7(He et al., 2020) |
| Flemish (Renkens et al., 2014) | 5940 | 36 | 11 | 63 (Köhn et al., 2016) |
| Mandarin (Zhu et al., 2019) | 6925 | 4 | - | 15 (Wang and Zhang, 2015) |

TABLE 14.1: Dataset statistics for English, Sinhala and Tamil datasets. PT denotes pre-training

when compared to English phonemic representations. Also, these phonetic transcriptions incorporate language specific nuances and is expected to generalize better to novel languages, especially to languages that are phonetically different from English.

A prototypical intent classification system was built for banking domain in Hindi from these phonetic transcriptions (Gupta et al., 2020b). In this work, a small natural speech dataset was used with Naive Bayes classifier as the intent recognition model. (Gupta et al., 2020a) showed that such intent recognition systems built on top of phonetic transcriptions work for a large number of languages, including various Indic and Romance languages. They also showed that multilingual training helps in building more robust systems and improves performance on an unknown language within the same language family. The intent classification system described in (Gupta et al., 2020a) was built for a large dataset with multiple intents, but this system was built using synthetic speech. In this work, we perform intent classification and slot identification experiments on standard SLU datasets with natural speech. These are the first results for our proposed SLU system on natural speech.

Transformer (Vaswani et al., 2017) based architectures have achieved state of the art performance in various speech and natural language processing tasks. BERT (Devlin et al., 2018) is a transformer based contextualized word embedding model which pushed the boundaries on performance on various NLP tasks including classification, natural language inference and question answering. BERT consists of the encoder modules of the Transformer, trained on the Masked Language Modelling (MLM) and the Next Sentence Prediction (NSP) objectives. RoBERTa (Liu et al., 2019), makes various modifications to the original BERT model including removing the NSP objective. In our work, we train a RoBERTa based model with a vocabulary of phones.

## 14.3 Datasets

We work with five standard SLU datasets for five different languages. All of these are natural speech datasets. For English, we use the Fluent Speech Commands (FSC) dataset (Lugosch et al., 2019), which is the largest freely available speech to intent dataset. The dataset was collected using crowdsourcing and was also validated by a separate set of people by crowdsourcing. The dataset has 248 distinct sentences spoken by 97 different speakers. The FSC dataset was further divided into train, validation and test splits by the respective authors, where the validation

and test sets comprised exclusively of 10 speakers which were not included in the other splits (Lugosch et al., 2019). Detailed dataset statistics are shown in Table 14.1. Each utterance in the FSC dataset has three types of slot values for *action*, *object* and *location*. The dataset can be modelled as a multilabel classification problem or a standard classification problem that flattens out all the different intents and slot values (Lugosch et al., 2019) (Radfar et al., 2020). We have used the 31-class intent classification formulation of the problem in our work. An example utterance in the dataset is given below:

Utterance: *Switch the lights on in the kitchen*

(action: activate), (object: lights), (location: kitchen)

The Flemish dataset, Grabo, (Renkens et al., 2014) was collected by asking users to control a service robot. There are 36 commands spoken by 11 different speakers which typically look like "move to position $x$" or "grab object $y$". The dataset was divided into a ratio of 60-20-20 for training, validation and testing. For detailed dataset statistics, please refer to Table 14.1. Here, each speaker was asked to say the same utterance 15 times. Thus, each intent class contains the exact same utterance repeated many times.

For Mandarin, we modify the CATSLU dataset (Zhu et al., 2019) to make it suitable for intent classification. The original CATSLU dataset is ideal for dialog state tracking with conversations about 4 domains - Navigation, Music, Video and Weather. We convert the conversations into a 4-class intent classification dataset into the above four domains. To do this, we chose utterances corresponding to the semantic labels of the above domains as labels. The dataset statistics are shown in Table 14.1. This dataset contains longer and free-flowing sentences when compared to the other datasets. Examples of utterances for the class of *weather* are:

*what's the weather in Shanghai today*, (Intent: Weather)

*Is it sunny tomorrow*, (Intent: Weather)

We see that the above utterances are much more complex than utterances in other datasets and requires inferring that *'Is it sunny tomorrow?'* corresponds to the domain of weather even though the word 'weather' is not present in the utterance. This makes this dataset the most complex out of all the datasets used in our work. This is also shown by the fact that a BERT-based textual intent classification model achieves a classification accuracy of only 93% and F1 score of 91%.

We also work with speech to intent datasets in Sinhala (Buddhika et al., 2018) (Karunanayake et al., 2019c) and Tamil (Karunanayake et al., 2019c). The Sinhala and Tamil datasets contain user utterances for a banking domain. The dataset has 6 different intents to perform common banking tasks including money withdrawal, deposit, credit card payments etc. Both datasets were collected via crowdsourcing. The Sinhala and Tamil datasets were not divided into train

and test splits by the respective authors and previous work and results provided in literature on these datasets are based on 5-fold cross-validation (Buddhika et al., 2018) (Karunanayake et al., 2019b) (Karunanayake et al., 2019c). The detailed dataset statistics are also shown in Table 14.1. The utterances in the Tamil dataset are at times code-mixed with English.

We also pre-train our models using large speech corpus released for public use. The hours of data used for pre-training is shown in Table 14.1. We pass the speech utterances present in pre-training corpuses through Allosaurus to obtain their phonetic transcriptions. These transcriptions are used to pre-train our models.

## 14.4 Models

To the best of our knowledge, in this work we train the first BERT-based language models for a vocabulary of phones. The transformer model we use is based on RoBERTa (Liu et al., 2019). We use the CLS token for generating sentence level representation for the input utterance, which is used for classification. We do a grid search for hyper parameters like number of attention heads, hidden layer size for the feed forward layers and number of encoder layers. We refer the reader to the RoBERTa (Liu et al., 2019) and Transformer (Vaswani et al., 2017) papers for architectural details. For pre-training the transformer, we use the MLM objective where 15% of the tokens are randomly masked. Out of those, 80% tokens and randomly changed to the token MASK, 10% tokens are changed to a random token and the remaining 10% are kept the same.

We modify the language model architecture proposed in (Gupta et al., 2020a) for our work. The architecture proposed in (Gupta et al., 2020a) consists of a Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) based language model followed by a fully connected classification layer attached to the final time step output of the LSTM. The language model consists of CNN layers with varying filter sizes, capturing N-gram like features of word embeddings, very similar to the architecture shown in Figure 14.4. The CNN layers are followed by an LSTM layer. The CNN+LSTM layer forms the language model of our phonetic transcriptions. For intent classification, the LSTM output at the final time step is fed into a fully connected layer to perform intent classification.

In addition to intent classification, we also propose an algorithm for unsupervised slot value identification, leveraging the self-attention mechanism (Bahdanau et al., 2014) to do so. We use the standard key-query-value formulation of the self-attention mechanism. The keys and values are the outputs of the LSTM at each time step, and the final time step output of the LSTM is used as the query. We use dot-product attention between query and key to calculate the attention scores. A softmax is taken across the attention scores for normalization. The final output of the self-attention mechanism is a linear combination of *values* weighted by their normalized attention scores. The output of the self-attention layer is sent to the fully connected layer for the classification decision, as shown in Figure 14.4.

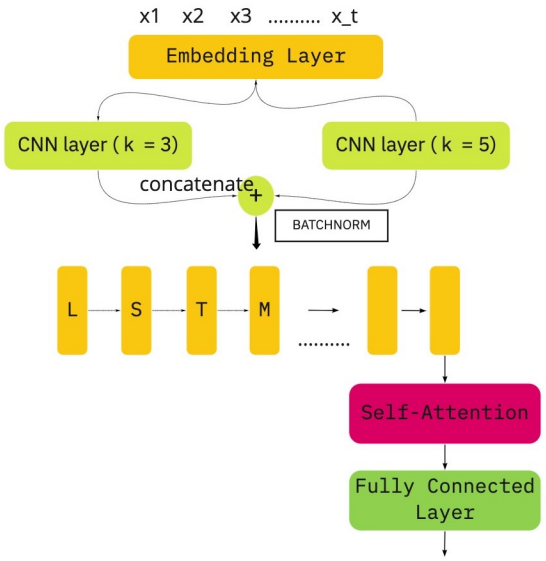



FIGURE 14.4: Model used for unsupervised slot identification.

| Dataset | Baseline* | LSTM | LSTM(PT) | Tranformer | Tranformer(PT) |
|---|---|---|---|---|---|
| English (FSC) | 98.8% | 92.67 % | 92.77 % | 90.77 % | 90.91 % |
| Flemish (Grabo) | 94.5 % | 78.82 % | 79.69 % | 85.41 % | 87.84 % |
| Mandarin | - | 65.59 % | 70.35 % | 64.29 % | 65.14 % |
| Sinhala | 97.31% | 95.68 % | 96.33 % | 95.60 % | 94.66 % |
| Tamil | 81.7% | 92.00 % | 91.50 % | 91.00 % | 92.50 % |

TABLE 14.2: Intent classification results in terms of accuracy for our proposed SLU system. *Note that baseline indicates the approach presented in the paper that introduced the dataset. The baselines we used are as follows: English (Lugosch et al., 2019), Flemish (Renkens et al., 2018), Sinhala (Karunanayake et al., 2019a) and Tamil (Karunanayake et al., 2019a). PT denotes pre-training.

.

The phonetic transcriptions of all intent classification datasets which were used as training data including the language specific data splits will be made available publicly along with the codes used in this work. The specific architectures used for achieving the results in section 14.5 will also be released along with the codes.

## 14.5 Experiments

### 14.5.1 Intent Classification

We compare the performance of our proposed system with previously reported results on five different languages - English, Belgian Dutch, Mandarin, Sinhala and Tamil. The results are shown in Table 14.2. We see that the results of our proposed method improves as the size of the dataset decreases. This makes our system an ideal candidate to be used for low-resourced

scenarios. We report state of the art results for Tamil, which improves on the previous best by approximately 11% and halves the error rate.

Our proposed phonetic transcription based intent classifier performs competitively on the relatively larger English dataset. The model trained on the FSC dataset in (Lugosch et al., 2019) is an E2E-SLU model, which has various advantages over a two-module split SLU system. An E2E-SLU model learns better representation of data as it directly optimizes for the metric of intent classification (Lugosch et al., 2019). Instead, our system consists of two blocks that are not optimized for the errors made by the other, causing errors to propagate through the system. Thus, an end-to-end system with enough data provides an upper limit for the intent classification results. Furthermore, results shown in Table 14.2 only use the top-1 predictions made by Allosaurus, which means we select the phone with the highest softmax score for generating phonetic transcriptions. When we use the top-5 predictions made by Allosaurus, thus giving more information about the spoken utterance to our models, we achieve an accuracy of 96.31%.

The Belgian Dutch dataset observes significant improvement with the use of Transformer models when compared to the other datasets. We attribute this improvement to the kinds of utterance present in the dataset. All utterances corresponding to the same intent in the Dutch dataset are spoken in the exact same word order. The positional embeddings of a transformer are responsible for encoding the grammatical structure of an utterance. When the structure is jumbled, as in the case of the other datasets which contain multiple ways of saying the same intent, the positional embeddings don't encode useful information and require larger amounts of data. Also, note that since multiple phones can correspond to the same spoken word, the token order in phonetic transcriptions are even more jumbled. Since the Dutch dataset does not have jumbled word order, there is minimal variability in the dataset and hence transformers are able to produce better results for Belgian Dutch when compared to other languages.

The Mandarin dataset is by far the toughest dataset. The other four datasets have simple commands with most sentence lengths of 2-5 words. The Mandarin dataset on the other hand contains free-flowing questions which are also longer in length. The utterances do not contain intent specific words as shown in section 14.3 which makes the task harder. This is why we see modest classification performance for the Mandarin dataset.

The baseline models for the Sinhala and Tamil datasets (Karunanayake et al., 2019b) are a two-module split SLU systems, where the intent classifier is built from phonemic transcriptions generated from an English ASR. The performance of our system is comparable for the Sinhala dataset while it significantly outperforms the phonemic transcription based model for the Tamil dataset. We again point out to the reader that in our systems, we've only used the top-1 softmax predictions made by Allosaurus, thus providing much less information about spoken utterances to our model. On the other hand, baseline models use the entire softmax layer vectors of the ASR systems as an embedding to encode spoken information.

The above results also show the effectiveness of using Allosaurus when compared to a language specific ASR for encoding spoken information. Phonemes are perceptual units of sounds and

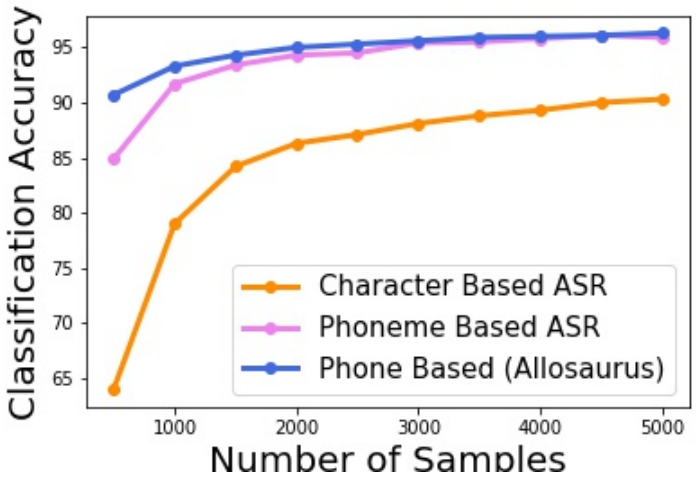



FIGURE 14.5: Comparing the performance of our proposed phonetic transcription based SLU system with previous characted and phone based systems.

changing phonemes ends up changing the spoken word. Phones on the other hand are language independent and correspond to the actual sound produced. Changing a phone does not necessarily change a word in a particular language. Usually multiple phones are mapped to a single phoneme, and this mapping is language specific. This makes using ASR systems built for a high-resourced language like English sub-optimal when cross-lingually encoding spoken utterances into a vector for chosen target languages. Allosaurus (Li et al., 2020) is a universal phone recognition model and is trained to recognize fine grained differences in spoken utterances at the level of phones. We hypothesize that the intent classification models are able to extract fine-grained phone level differences when larger amounts of data is available, but as the amount of data reduces, this becomes increasingly difficult. This is why we see improvement in performance as the dataset size decreases.

To illustrate the effectiveness of our proposed system over previous approaches, we randomly select subsets of the utterances in Sinhala in increments of 500, just as it was done in (Karunanayake et al., 2019c) (Karunanayake et al., 2019b). We compare the performance of our phonetic transcription based model to the English character based ASR model (Karunanayake et al., 2019c) and the English phoneme based ASR model (Karunanayake et al., 2019b). The authors very generously provided us with the exact numbers for the comparison. The results can be seen in Figure 14.5 . We can see that intent classifiers built from phonetic transcriptions produced by Allosaurus significantly outperforms intent recognition systems built on top of character or phoneme based transcriptions as the amount of data reduces. For a training set size of 500 samples for the Sinhala dataset, we improve on the English character based system by approximately 25% and on the English phoneme based system by more than 5%. The transformer and pre-trained models do not outperform the CNN+LSTM models, especially for low data scenarios.

### 14.5.2 Unsupervised Slot Value Identification

With the aim of creating entire NLU modules based on phonetic transcriptions of speech, we shift our focus on slots. The Sinhala and Tamil datasets used in this work do not have sequence level slot information either in speech or textual transcriptions. This is a very realistic low-resource scenario where we cannot expect utterances to have labelled slot values. Similar argument holds for unwritten languages. In this section, we propose an attention based **L**ow-Resource **U**nsupervised **S**lot value **Id**entification (LUSID) algorithm to identify slots values when no labels are present. We also aim to identify the span of existing slots in our training data (note that our training data is phonetic transcriptions of speech).

#### 14.5.2.1 Problem Definition

We pose the unsupervised slot value identification as a classification problem. We use an attention based classification model as described in section 14.4 to identify slot values in an unsupervised manner. A self-attention module is added before the final classification layer of the LSTM+CNN based model. To test our algorithm, we create a 2-class attention-based classification model. The two classes correspond to two slot values belonging to the same intent, thus the differentiating feature is the slot value between them.

We use an example from the FSC dataset for illustration. The intent of *activating lights* is used from the FSC dataset with two slot values - *bedroom* and *kitchen*. Figure 14.6 shows the normalized attention scores for a given phonetic input when the utterances are passed through the attention-based classifier. The title of the figure represents the textual transcription of the speech input. The x-axis labels correspond to the phonetic transcription of the input produced by Allosaurus. The y-axis represent the normalized attention scores for each token in the phonetic transcription. Figure 14.6 shows activated weights for phones corresponding to the word bedroom. We can see that the self-attention mechanism is able to identify the approximate location of the slot value for the slot *location*.

#### 14.5.2.2 Identifying Exact Slot Location

Next, we identify the location of slot values in an utterance. This would enable us to replace the slot value with a new slot value, thus generating synthetic data. For the purpose of this illustration, let's say our base slot value is *bedroom* and the target slot value is *kitchen*. *Base slot value* is the slot value in the utterance we're working with. This is the slot value we want to replace in the current utterance with the *target slot value*, keeping the remaining utterance the same. To do this, we identify the phone corresponding to the highest attention score for the base slot value, and remove all phones within a left window of size $l = 4$ phones and a right window of size $r = 3$ phones from the highest score phone. This gives us the location of the base slot value. $(l, r)$ are tuned as hyperparameters for each model. Once we have the

| Dataset | Left Window | Right Window | Classification Accuracy |
|---|---|---|---|
| English (FSC) | 4 | 3 | 99.24 % |
| Sinhala | 8 | 1 | 93.61 % |

TABLE 14.3: Classification accuracy of generated utterances using LUSID.

location of the base slot value, we replace it with the target slot value (corresponding to the phonetic transcription of *kitchen*).

#### 14.5.2.3 Verification of Generated Data

The above process gives us a synthetically generated utterance for the slot value *kitchen* from an utterance corresponding to the slot value of *bedroom*. Note that this new utterance is generated purely in the phonetic transcription domain, thus avoiding the need for textual transcriptions and supervised slot level labelling. Next, we need to test that this new generated utterance actually corresponds to the target slot value, *kitchen*. To do so, we feed the generated utterance back to the same classifier, with the expectation of now being classified into the target slot value class, *kitchen*. The accuracy for the new utterance generated from the base slot value, being classified as the target slot value, is shown in Table 14.3 with optimal $(l, r)$ values for both the English and the Sinhala dataset. For the English slots, the best classification accuracy achieved is 99.24 %, which means that the model classifies the generated data into the target class 99.24% time. This shows that the model is not able to differentiate between generated utterances and actual data. For the Sinhala dataset, we used the intents *bill Payments* and *credit card payments* and achieved an accuracy of 93.61 %.

#### 14.5.2.4 Discussion

It is important to highlight the non-triviality of these results. Firstly, the slot can be present anywhere in the sentence and we identify the span of the slot in an unsupervised way. This can be seen in Figure 14.6. Secondly, taking the example of the English dataset, the generated utterances are created from the training data for base slot value of the *bedroom* class. Thus, the model is trained with almost 100% accuracy to recognize the non-replaced part of the utterance as the *bedroom* class. Yet, it recognizes new utterance generated from the base class (*bedroom*) as belonging to the target class (*kitchen*), showing that we have successfully removed the slot value which was responsible for making the classification decision.

LUSID can be used to generate artificial data for a new slot value from a single spoken utterance. We can also use this algorithm to generate unsupervised slot labels in the phonetic transcription domain when slot labels are not present as well as for data augmentation, since it allows us to generate new data samples for a given slot value for an existing dataset.

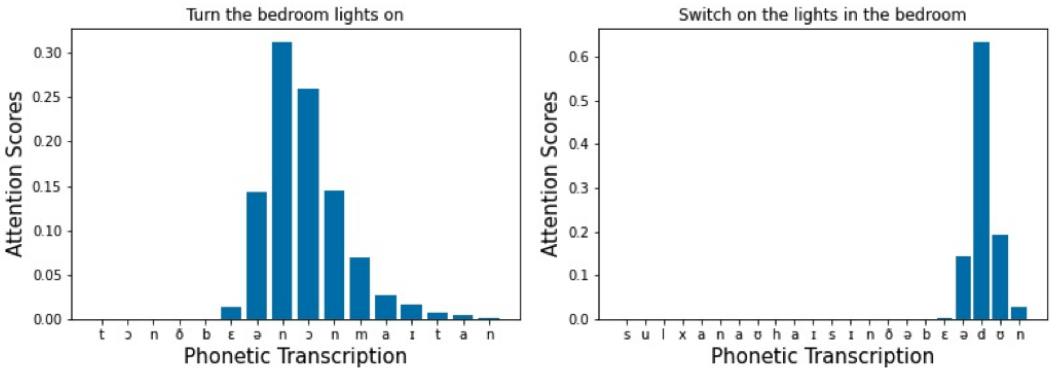



FIGURE 14.6: Attention scores for each phone in the phonetic transcription of an utterance.

## 14.6 Conclusions

In this work, we present a unique spoken language understanding system (SLU) for low-resourced and unwritten languages. The SLU system converts speech to its phonetic transcription using a universal speech to phone converter. We then build natural language understanding modules for utterances in the phonetic transcriptions domain, which perform competitively with current end-to-end SLU models and outperforms state of the art approaches for low-resourced languages. Moreover, we show that the performance of our system surpasses state of the art systems as the amount of labelled data decreases, which makes it an ideal candidate for low-resourced settings. We also propose an attention-based unsupervised slot value identification algorithm that identifies slots in the phonetic transcription domains when slot labels are not present. This technique, which we call LUSID, has various applications from one-shot data generation to data augmentation.

# 15

# EXPLAINABILITY - Justification by De-Entanglement: A Case Study with Language Identification

Explainability is a significant challenge to be addressed before a technology can be considered ubiquitous. While there have been many works(Lipton, 2018; Miller, 2019) already, this seems to be an evolving topic with multiple perspectives. In this dissertation, I limit myself to one particular topic - 'Justification' within the suite of seemingly many equally important topics that Explainability entails. In addition, I also limit myself to explore one particular line of investigation with respect to Justification - an architecture with multiple models where one of the models attempts to justify the predictions by the overall architecture. I understand that this approach has its limitations. For instance, the approach attempts to justify the predictions of one machine learning model using another. Having said that, I believe that it is possible to specify certain constraints on the participating models under which the architecture is employed to generate plausible justifications. In this chapter, I present a preliminary case study using an architecture with two tasks - a primary task of interest that requires Justification and an auxiliary task that provides Justification. Specifically, I employ Code Mixed detection - detection if an utterance is code mixed using acoustics - as the primary task and Acoustic Unit Discovery(from chapter 11) as the auxiliary task.

## 15.1 Introduction

Code Mixing is a conversational phenomenon where linguistic units such as phrases, words and morphemes of one language are embedded into the utterance of another language (Muysken, 2000; Gella et al., 2014). This is quite common in multilingual societies such as in India, Singapore where English has transitioned from the status of a foreign language to that of a second language. Today such mixing has manifested itself in various types of text ranging all the way from news articles through comments/posts on social media, leading to co-existence of multiple languages in the same sentence. In the context of devices such as Alexa/Siri, interfaces deployed in code mixed contexts should to be able to process mixed speech without ignoring the content from one of the languages. Identifying whether or not the utterance is mixed, and the participating languages in case of a mixed utterance can benefit the downstream speech and natural language models.

In this work, we present an approach aimed at identifying if a given speech utterance is code mixed or not. It has been shown that humans accomplish language detection by using two types of information from the speech utterance(Zhao et al., 2008): content information such as phonetic repertoire, phonotactics and style information such as rhythm and intonation. Infants have been shown to rely on content information to discriminate between languages even before having gained lexical knowledge(Zhao et al., 2008). Inspired by this, we propose an approach aimed at hypothesizing the discrete linguistic space of the given speech utterance and using this information to detect if the utterance is characterized by code mixing. To accomplish this, we employ multi task learning paradigm with detection of mixing as the primary task and speech reconstruction as the secondary task. Specifically, we employ sequence to sequence models with latent random variables for reconstructing speech, thereby mediating the task of estimating likelihood through stochastic latent variables. Since these models provide a mechanism to jointly optimize both latent representations as well as the likelihood corresponding to the downstream task, they are expected to discover causal factors of variation present in the distribution of original data.

A typical optimization challenge observed while training latent stochastic variable models is referred to as KL-collapse (Bowman et al., 2015), wherein the decoder network marginalizes the learnt latent representation. Approaches to dealing this issue involve annealing the KL divergence loss (Bowman et al., 2015; Zhou and Neubig, 2017b), weakening the generator (Zhao et al., 2017b) and ensuring the recall using bag of words loss. In (van den Oord et al., 2017a), authors propose a principled solution using vector quantization in the latent space, effectively making the loss due to KL Divergence a hyperparameter. In addition, they show that the resultant discrete latent representations correspond to linguistic units. Our approach is inspired by this observation and we hypothesize that the learnt discrete units must be different for monolingual and code mixed utterances. Building on (van den Oord et al., 2017a; Chorowski et al., 2019b), we add additional constraints in the prior space forcing the latent representations to follow articulatory dimensions: The encoded representation is hashed to a latent code based on a articulatory prior bank designed using a discrete codebook. This coded representation

is fed both to our decoder to detect mixing as well as to the generator for reconstructing the input. Our generator is a conditional WaveNet using speaker embedding as global embedding trained to regenerate input audio using the coded sequence as local information.

This work is organized as follows: In section 15.2, we present some background followed by earlier work on exploiting acoustic units. This is followed by an explanation of our proposed approach in section 16.2. We present our experiments in section 15.4 followed by an analysis of the proposed approach. This is followed by conclusion in section 15.5.

## 15.2 Background

### 15.2.1 Acoustic Unit Discovery

Let us consider a speech corpus X which consists of speakers $\{s_1, s_2..., s_n\}$. The goal of acoustic unit discovery is to come up with a set of units **U** that represent a speech utterance $x \subset$ X allowing robust resynthesis. The elements of such a set also might conform to desirable characteristics such as being injective, consistent and compact, i.e. that different inputs should have discriminant acoustic units, but expected variance such as speaker or dialect should produce the same acoustic units.

There have been numerous attempts to discover such acoustic units in an unsupervised fashion. In (Huijbregts et al., 2011), authors presented an approach to modify the speaker diarization system to detect speaker-dependent acoustic units. (Jansen et al., 2013) proposed a GMM-based approach to discover speaker-independent subword units. However, their system requires a separate Spoken Term Detector. Recently, due to the surge of deep generative model, using unsupervised method such as auto-encoder and variational auto-encoder (VAE). (Badino et al., 2014) designed a stacked AutoEncoder using backpropagation and then cluster the representations at the bottleneck layer. To avoid quick transitions leading to repeated units, they employed a smoothing function based on transition probabilities of the individual states. (Ebbers et al., 2017) extended the structured VAE to incorporate the Hidden Markov Models as latent model. (van den Oord et al., 2017a; Chorowski et al., 2019b) proposed VQ-VAE and argue that by vector quantization the "posterior collapse" problem could be circumvented.

### 15.2.2 Neural Generative Models for Speech

Artificial generation of speech based on neural approaches has soared in the recent past. There have been continuous and significant improvements in both the aspects of speech generation - fidelity and flexibility. Autoregressive models such as (van den Oord et al., 2016), flow based models such as (Prenger et al., 2018a) have shown to generate audio that rivals the quality of natural speech. Approaches such as (Taigman et al., 2017b,a) have shown ways to incorporate inductive biases into the generative process. (Watts, 2012) developed generic methods to

enable the usage of distributional analysis of text at phone, word, and character levels in an unsupervised fashion. These techniques have been utilized in building highly flexible systems capable of generating different styles of speech and ability to build voices from noisy or very minimal data.

### 15.2.3 Disentanglement

In (Chen et al., 2018b), authors decompose Evidence Lower Bound (ELBO) and show that there are terms measuring the total correlation between the latent variables. In (Burgess et al., 2018a), authors propose incorporating a channel capacity term to promote disentanglement of causal factors of variation in the data. Our work is similar to these in that we analyze ELBO to show that it is possible to control what gets disentangled. In (Esmaeili et al., 2018a), authors present a generalization of ELBO by factorizing the latent representation into a hierarchy. In (Ansari and Soh, 2018), authors present an approach to accomplish disentanglement by modifying the co-variance matrix of the latent representations. In (Kim and Mnih, 2018) authors augment ELBO using the density ratio trick to accomplish disentanglement. In (Hoffman and Johnson, 2016), authors posit that to improve ELBO we must also improve the marginal KL, meaning we must have good priors. In (Banijamali et al., 2017) authors show that actively trying to disentangle the causal factors of variation is better than trying to pressurize the model to forget the invariant representations. We take inspiration from these approaches that manipulate the prior distribution and impose domain specific constraints - based on intuitions from articulatory features - on the prior space. Manipulating prior space has other benefits such as interpreting the intermediate stage outputs of the model. However, such analysis is beyond scope of the current study.

## 15.3 Proposed Approach

Let us consider a speech corpus $X$ consisting of languages $\{l_1, ..., l_n\}$, where each $l_i$ might comprise of multiple speakers. Let $x_1$,...,$x_n$ denote acoustic frames $X$. Note that $x_i$ might be either monolingual or a code mixed utterance and let $Y$ denote this information. Our model learns the joint distribution between $\{\mathbf{x}, \mathbf{y}\}$. We mediate this process using latent discrete random variables represented by $Z$.To ensure that the latent representations correspond to the corresponding speech utterance, we also add a generative decoder that generates raw audio samples. Thus our model performs multi task learning with language detection as the primary task and reconstruction as the auxiliary task. Our model can be summarized by the following set of equations:

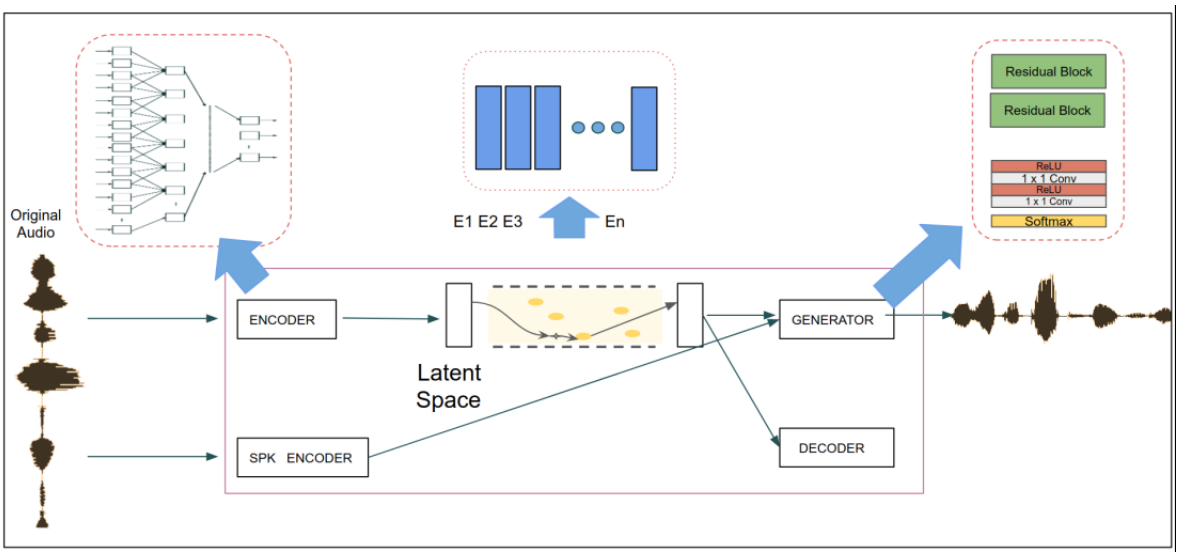



FIGURE 15.1: *Architecture of proposed approach. Both our Encoders perform downsampling of the input sequence. Our generator is a WaveNet. Our decoder predicts logits denoting the language information in the utterance. (Best viewed in color)*

$$
\begin{aligned}
encoded_{1:T2} &= \mathbf{H}^{Encoder}(x_{1:T1}) \\
spk_{encoded} &= \mathbf{H}^{SpeakerEncoder}(x_{1:T1}) \\
z_{1:T2} &= \mathbf{VQ}(encoded_{1:T2}) \\
y &= \mathbf{H}^{Decoder}(z_{1:T2}) \\
y_{audio} &= \mathbf{H}^{Generator}(z_{1:T2}, spk_{encoded})
\end{aligned}
\tag{15.1}
$$

At training time, parameters are learnt using Variational Inference. Specifically, we first draw latent code sequence(dropping the subscript for brevity) $z$ from the current posterior represented by $\mathbf{H}^{Encoder}$. We then feed $z$ into the decoder to optimize the likelihood and feed $z$ along with $spk_{encoded}$ into the generator to reconstruct $x$. During inference, we pass the speech utterance through encoders and obtain speaker encoding as well as the latent units. The latent units are then fed to the decoder to obtain logits. The architecture of our proposed approach can be found in figure 15.1.

### 15.3.1 Model Components

#### 15.3.1.1 Encoders

We have two acoustic encoders, one for obtaining speaker encoding and the other for obtaining temporal representation to be fed to the variational layer. Both these encoders are implemented as downsampling encoders following the same architecture as in (van den Oord et al., 2017a). For speaker encoder, we have two additional components: We employ an LSTM layer on top

of the downsampling encoder to summarize the output and use then final hidden state output from LSTM as the embedding. This embedding is passed through attention layer similar to the global style token attention from (Wang et al., 2018c). The only difference in the tokens learn via attention is that in our model, the vectors correspond to speakers.

#### 15.3.1.2 Decoder

Our decoder consists of an LSTM followed by an Attention block and a linear layer. The latent units $z$ are first passed through decoder LSTM and then Attention block acts on the temporal sequence output by LSTM. We employ soft attention and implement it using dot product.

#### 15.3.1.3 Latent Vector Quantization and Articulatory Priors

Speech presents a characteristic advantage in that a speech utterance has both continuous and discrete priors. The generative process of speech assumes a Gaussian prior distribution which is continuous in nature. However, speech utterance also encompasses information about language which can be approximated to be sampled from a discrete prior distribution. Exact manifestation of this in the can be at different linguistic levels: phonemes, words, syllables, subword units, etc. Inspired by this, we engineer our prior space to account for the phonetic information in the utterance by representing the prior as a discrete latent variable bank. Each discrete latent variable has a different set of states reflecting one of the articulatory dimensions. The specific design of our latent space is highlighted in Table 15.1.

#### 15.3.1.4 Generator

We employ WaveNet (van den Oord et al., 2016) as our generator. The joint probability of a waveform **X** can be written as:

$$P(Y_{audio}|\theta) = \prod_{t=1}^{T} P(z_t|z_1, z_2..z_{t-1}, \theta) \tag{15.2}$$

given model parameters $\theta$. During implementation of WaveNet, the autoregressive process is realized by a stack of dilated convolutions. The final output $y_t$ at time step $t$ can be expressed mathematically as:

$$\hat{y_t} \sim \sum_{d=0}^{D} h_d * r_d(x) \tag{15.3}$$

where $D$ is the number of different dilation used and $d$ is the dilation factor; $h_d$ is the convolution weights. This stack of convolutions is repeated multiple times in the original WaveNet. Optimization in WaveNet is performed based on the error between predicted sample and the ground truth sample conditioned on previous samples in the receptive field alongside the local conditioning. We define the divergence similar to the (Salimans et al., 2017).

### 15.3.2 Optimization and Model Interpretation

The loss we optimize is given below:

$$\begin{aligned} L = \lambda_{reconstruction} MOL(y_{audio}, \hat{y}_{audio}) \\ +\lambda_{LID} CE(y, \hat{y}) \\ +\lambda_{encoder}(EncoderPenalty) + \lambda_{VQ}(VQPenalty) \end{aligned} \tag{15.4}$$

To optimize generator, the contribution from the individual convolution layers towards this global error function must be nullified. Let us consider the expression for intermediate output for a single filter in Eqn 15.3:

$$y_{audio}(t) = \sum_{\tau=0}^{t} h(\tau) z(t-\tau) \tag{15.5}$$

where $\tau$ is the receptive field covered by the model and $h(\tau)$ represents the discrete state representation at time $t$. This can be interpreted as follows: generator model acts as the transfer function and the input is discretized by convolving with the filters controlled by dilation rate. It has to be noted that this is similar to the formulation of source filter model of speech.

The advantage of formulating generative model in this fashion is the presence of random variables that might capture the causal factors of variation in input based on prior information about the distributional characteristics of data. Techniques aimed at this (Higgins et al., 2016) have shown that it is possible to effectively disentangle the factors of variation using stochastic variables. Hence, we postulate that augmenting our generator with appropriate prior distribution helps obtain the acoustic phonetic units. We employ articulatory priors in this work. Since the latent representations are shared between reconstruction and detection of mixing, we hypothesize that the extracted units will be useful in detecting if a given utterance exhibits code mixing.

Table 15.1: Articulatory Phonetic Features

| Feature name | Value | Details |
|---|---|---|
| vc | + - 0 | vowel or consonant |
| vlng | s l d a 0 | vowel length |
| ctype | s f a n l r 0 | consonant type |
| cplace | l a p b d v g 0 | place of articulation |

### 15.3.3 Hyperparameters

The architecture of our model is built on top of VQ-VAE. As our encoders, we use dilated convolution stack of layers which downsample the input audio by 64. The speech signal was power normalized and squashed to the range (-1,1) before feeding to the downsampling encoders. For speaker encoder, we add attention block(Wang et al., 2018c) after the downsampling encoder and it had 100 classes. To make the training faster, we have used chunks of 2000 time steps. This implies we get 31 timesteps at the output of the encoder. The quantizer acts as a bottleneck and performs vector quantization. Quantization is implemented using minimum distance in the embedding space. The number of classes was chosen to be 64, approximating 64 universal phonemes. We use a linear mapping to first project the 128 dimensional vector to 160 dimensions. We then perform comparisons with respect to individual articulatory dimensions. Assuming $z_e(x)$ denotes the encoder output in the latent space, then the input of decoder $z_d(x)$ will be obtained by argmin of $d(e_j, z_e(x))$, where $d$ is a similarity function of two vectors. In this work, we consider Euclidean distance as the similarity metric. Our generator is an iterated dilated convolution-based WaveNet that uses a 256-level quantized raw signal as the input and the output from vector quantization module as the conditioning. The decoder takes the output from the quantizer and is trained using cross entropy to predict if the utterance is mixed or not.

## 15.4 Experimental Setup

### 15.4.1 Data

We have used data released as part of shared task on detection of code mixing from speech (India, 2020). Data included speech utterances for three Indian languages Gujarati, Tamil and Telugu. The labels for each utterance indicated if the utterance exhibits code mixing. No transcriptions or speaker labels have been provided as part of the official data release. The release included train-dev-test splits and therefore we have followed the same splits.

### 15.4.2 Baselines

An official baseline was released along with data (India, 2020). Official baseline employed acoustic model similar to DeepSpeech and used CTC loss to optimize the network. We have built the following systems as baselines:

#### 15.4.2.1 CBHG Baselines

This class of baselines were built using CBHG from Tacotron(Wang et al., 2017b) as the acoustic model and a linear layer as the decoder. This baseline was trained using cepstral representation. We have experimented with 80 dimensional Mel features used typically in speech synthesis as well as 39 dimensional MFCCs used in speech recognition.

#### 15.4.2.2 Downsampling Baselines

This class of baselines used downsampling encoder from (van den Oord et al., 2017a) as the acoustic model and a linear layer as the decoder. We have experimented with different rates of downsampling from 2 through 64. We have also experimented with different input representations of audio: cepstral and raw audio. For systems using raw audio, we have also performed $\mu$ law quantization using 256 levels.

#### 15.4.2.3 Experts Baselines

We built Mixture of Experts baselines extending the approach proposed in (Zhao et al., 2019). Specifically, we train a number of expert models with soft parameter sharing implemented by gating mechanism. We hypothesized that each expert could capture information relevant to one language, or the individual linguistic units. We experimented with 2 through 8 experts where each expert was hypothesized to track the characteristics of a language and/or mixture of languages.

TABLE 15.2: Accuracy and Equal Error Rates on Dev Set for Various Systems in all the three languages: Gujarati, Tamil and Telugu. BL - Baseline

| System | Gujarati | Tamil | Telugu |
|---|---|---|---|
| Official BL | 76.8 / 11.6 | 71.2 / 14.4 | 74.0 / 13.0 |
| CBHG BL | 75.7 / 11.9 | 70.4 / 14.9 | 68.8 / 14.3 |
| Downsampling BL | 76.2 / 11.7 | 70.2 / 15.0 | 67.4 / 14.5 |
| Experts BL | 77.4 / 11.2 | 71.7 / 14.3 | 74.8 / 13.0 |
| Proposed Approach | **78.2 / 10.8** | **73.1 / 13.2** | **77.1 / 12.8** |

## 15.5 Conclusion

In this work, we investigate approaches towards building systems capable of detecting code mixing from a speech utterance. For this we employ multi task learning by using generative model of speech with discrete latent stochastic variables as auxiliary task. We compare our approach with a several baselines and show that our approach outperforms them.

# Part V

# Extensions to other Modalities

# 16

# Extensions: Image Captioning

Multimodal tasks require learning joint representation across modalities. In this work, we present an approach to employ latent stochastic models for a multimodal task - image captioning. Encoder Decoder models with stochastic latent variables are often faced with optimization issues such as latent collapse preventing them from realizing their full potential of rich representation learning and disentanglement. We present an approach to train such models by incorporating joint continuous and discrete representation in the prior distribution. We evaluate the performance of proposed approach on a multitude of metrics against vanilla latent stochastic models. We also perform a qualitative assessment and observe that the proposed approach indeed has the potential to learn composite information and explain novel combinations not seen in the training data.

## 16.1 Introduction

Tasks involving multiple modalities such as Audio Visual Speech Recognition (Afouras et al., 2018), Visual Question Answering (Antol et al., 2015a), Video Transcription (Chen et al., 2017c), Translation (Su et al., 2018), etc are AI complete in some capacity and therefore need to deal with the challenges of Representation Learning, Translation, Alignment, Fusion and Co-learning (Baltrušaitis et al., 2018) of the modalities present. Such tasks are also deceptively non trivial - they tend to give a false illusion of having learnt visually grounded representations (Shekhar et al., 2017). Traditional encoder-decoder architectures for such tasks have shown to learn biases present in the data (Goyal et al., 2017; Agrawal et al., 2018). Such models fail to learn robust

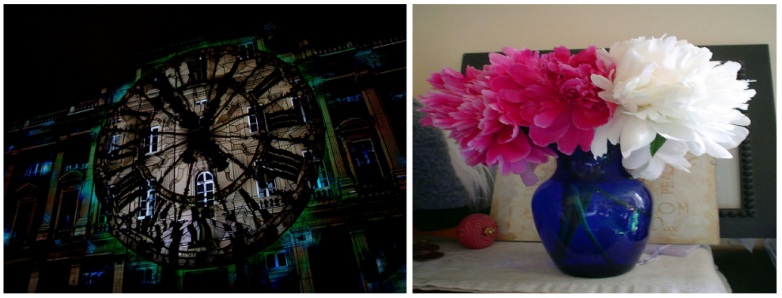


FIGURE 16.1: (a) **Ground Truth**: A Gigantic clock is displayed on the side of a building. **Proposed Model**: a very tall clock with roman numerals on a wall. (b) **Ground Truth**: A small blue glass vase on a table. **Proposed Model**: A vase filled with pink roses on top of a table.

representations, and do not generalize to unseen compositions of the seen objects (Agrawal et al., 2017b). In addition, such models are easily prone to adversarial attacks (Chen et al., 2017b; Chakraborty et al., 2018; Zhao et al., 2018b; Yuan et al., 2017). In this work, we present an initial approach to incorporate and learn latent stochastic random variables in encoder decoder models (Chung et al., 2015; Kingma et al., 2016; Chen et al., 2016) for such multimodal tasks using image captioning as a case study.

Specifically, we investigate the ability of latent stochastic encoder decoder models to learn disentangled representations. Disentangled representations are defined as ones where a change in a single unit of the representation corresponds to a change in single factor of variation of the data while being invariant to others (Bengio et al., 2012). Such representations are attractive from the perspective of generalizabilty across tasks (Esmaeili et al., 2018b), zero shot learning(Higgins et al., 2017), transfer learning and low resource scenarios. Moreover, disentangled representations are usually aligned with the attributes of original data and are conditionally dependent on variance in the original data, hence are more interpretible (Lipton, 2016).

For image captioning, the deployed models are first expected to summarize both global information like objects and their positions in an image and local information like attributes and relation with other objects. Further, the models are required to generate factual and grammatically meaningful text descriptions. We hypothesize that latent stochastic models provide a flexible framework for address the challenges involved in such generative tasks. These models provide a mechanism to jointly train both the latent representations as well as the downstream inference network. They are expected to both discover and disentangle causal factors of variation present in the distribution of original data, so as to generalize at inference time. We believe that disentanglement is an important property for such tasks as it can improve the ability of models to generate new concepts by combining different global and local properties (see Figure 16.1). Due to the nature of challenges involved and the flexible framework of deep-latent models, we employ image captioning using latent stochastic models as the testbed for our experiments in this study.

While training latent stochastic models, optimizing the exact log likelihood can be intractable. To address this, a recognition network is employed to approximate the posterior probability

using reparameterization (Kingma and Welling, 2013). When deployed in encoder decoder models, this approach is often subject to an optimization challenge referred to as KL-collapse (Bowman et al., 2015) - wherein the generator (usually an RNN) marginalizes the learnt latent representation. Typical approaches to dealing this issue involve annealing the KL divergence loss(Bowman et al., 2015; Zhou and Neubig, 2017b), weakening the generator (Zhao et al., 2017b) and ensuring the recall using bag of words loss.

In this work, we present a method to incorporate inductive bias into latent stochastic models by forcing the prior distribution to be slightly more complex compared to the univariate Gaussian distribution typically employed. Specifically, we propose to split the latent prior space used for approximating the posterior distribution into continuous and discrete counterparts. This is motivated by the observation that tasks involving multiple modalities usually inherently contain both continuous and discrete factors that are responsible for the generation of observed data. In the context of caption generation both the involved modalities - textual even though primarily symbolic and visual even though primarily spatial - are characterized by distinct discrete and continuous factors of variation. For instance, distinct objects or entities would intuitively perhaps be better represented by discrete variables, while their spatial location and relationships between them might be represented by continuous variables. Based on this hypothesis, we constrain the latent prior space to include both continuous as well as discrete variables, thus forcing the model to encode important information into the latent representation, and subsequently forcing the generator to use this information during inference. Our contributions are as follows: (1) We propose a simple yet effective architecture that splits the latent space into continuous and discrete factors that better capture the relations between entities. (2) We perform quantitative and qualitative analysis on MSCOCO dataset and observe that the model is able to not only generate diverse captions but also makes less mistakes in terms of entity attributes.

## 16.2 Proposed Approach

### 16.2.1 Analysis of optimization and disentanglement

Latent stochastic models have shown promising results in unsupervised, unimodal settings and are the preferred models for representation learning. However, when we apply these models in an encoder decoder framework, optimization becomes harder due to KL-vanishing (Bowman et al., 2015). This is mainly because the latent variable distributions are usually approximated by simpler networks compared to the powerful RNNs used in the encoders and decoder (Chen et al., 2016).

The problem becomes apparent by looking at the Variational Lowerbound (ELBO) such models try to optimize. For instance, consider the ELBO being optimized by Beta CVAE:

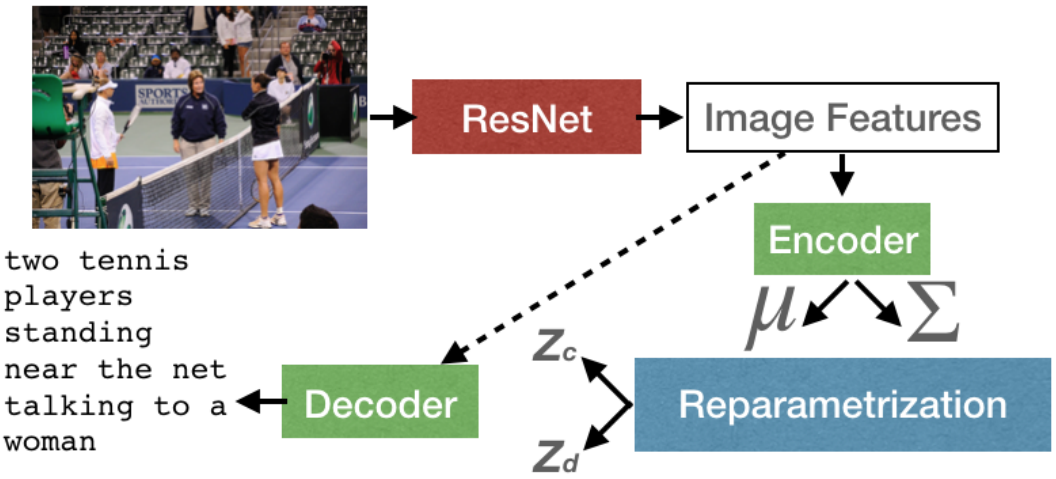



FIGURE 16.2: The latent representation space in the proposed model is split into continuous ($z_c$) and discrete ($z_d$) prior space.

$$E_{q_\phi(z|x,c)}[logp_\theta(x|c,z)] - \beta|D_{KL}(q_\phi(z|x,c)||p_\theta(z|c)) - C_z| \tag{16.1}$$

where $C_z$ is the channel capacity term (Burgess et al., 2018b). The first term in ELBO is the reconstruction error while the second is the divergence between approximate and true posteriors. Rewriting the first term as

$$logp_\theta(x|c,z) = logp_\theta(x|z,c) + logp_\theta(z|c) \tag{16.2}$$

It can be seen that the optimal value of this likelihood estimate can be conditionally independent of the latent representation (z) if the recognition network is complex enough (Shen et al., 2018). In other words, if the decoder network employs powerful universal approximators, the model is incentivized to ignore the latent representation. The second term in the expression acts as a regularizer to penalize such behavior. However, a trivial solution for the model is to force this posterior distribution to closely follow the Gaussian prior distribution (Chen et al., 2016).

The second term, KL divergence between the true and approximate posterior distributions obtains the global minimum 0 only when both the distributions match each other. From Bayes rules,

$$p_\theta(z|x,c) = \frac{p_\theta(x|z,c)p_\theta(z|c)}{p_\theta(x|c)} \tag{16.3}$$

It can be seen that a trivial solution to reach global minimum again is by ignoring the latent variable. Models such as $\beta$ VAE and the subsequently proposed channel capacity based

approaches (Burgess et al., 2018b) address this issue by gradually increasing the channel capacity. This would effectively result in pressurizing the posterior distribution to match the prior closely. However, following such an approach translates to an unrealistic constraint in scenarios that have categorical distribution as their output (tasks such as language modeling, machine translation, image captioning among others). In addition, it is unintuitive to assume that the true prior that generates latent distribution is a Gaussian when the likelihood is based on discrete sequential data. In such cases the decoder is implicitly weakened and the model is forced to encode information into the latent dimensions. At this stage, any local information is encoded within the hidden states of the decoder while remaining information is encoded in the latent space (Shen et al., 2018). Thus, we obtain disentanglement of independent factors of variation in the original data.

However, it has to be noted that during training optimization is performed in expectation over minibatches. The KL term can then be written as

$$E_{p(x)}[D_{KL}(q_\phi(Z|x)||p(z))] = I(x;z) + D_{KL}(q(z)||p(z)) \tag{16.4}$$

In other words, the KL term is the upperbound on the mutual information that can be encoded into the latent dimensions (Makhzani and Frey, 2017a). Penalizing this mutual information results in an increased reconstruction loss. Therefore, optimization in latent stochastic models follows a compromise between the capability for reconstruction and the potential for disentanglement.

### 16.2.2 Models

**Base System (CNN + RNN)**: As the base for our latent stochastic models we used a simple but powerful Encoder Decoder architecture. In our encoder framework we have used pretrained ResNet features, that inturn have CNN feature extractors. The decoder RNN is trained in a teacher forcing fashion by stacking together encoder output and caption embedding.

**Latent Stochastic Baseline Model (VED):** This system is a modification of our base system to include variational inference. Specifically, we designed our encoder model to output the mean ($\mu$) and log variance ($\Sigma$) of the latent distribution. We then sample a latent representation ($z$) using reparameterization trick (Kingma and Welling, 2013). Decoder is same as our baseline. The input to decoder is a stacked vector of latent vector and caption embedding. For training this model, we use scheduled annealing using logistic function for KL divergence as pointed out in (Bowman et al., 2015; Zhou and Neubig, 2017b). The step size for logistic function was fixed at 2500.

**Multi Space Latent Stochastic Model Config A**: In this system we have incorporated joint continuous ($z_c$) and discrete ($z_d$) latent representation as the prior distribution being modeled by the latent stochastic model. Since there are around 80 unique objects in MS COCO dataset,

| System | BLEU 4 | METEOR | CIDER | ROUGE L |
|---|---|---|---|---|
| RNN Baseline | 12 | 0.15 | 0.32 | 0.38 |
| VED Baseline | 13 | 0.15 | 0.33 | 0.40 |
| CCVED (Ours) | **16** | **0.18** | **0.49** | **0.43** |

TABLE 16.1: Performance comparison across models

it might be intuitive to allow atleast so many dimensions in the discrete space. Following this intuition, we have used 128 dimensions each for the discrete and continuous components.

**Multi Space Latent Stochastic Model Config B**: In this system we have incorporated joint continuous and discrete latent representation as the prior distribution being modeled by the latent stochastic model. Since there are around 80 unique objects in MS COCO dataset, it might be intuitive to allow atleast so many dimensions in the discrete space. Following this intuition, we have used 128 dimensions each for the discrete and continuous components.

**Multi Space Latent Stochastic Model Config C**: In this system we have incorporated joint continuous and discrete latent representation as the prior distribution being modeled by the latent stochastic model.

## 16.3 Experimental Setup

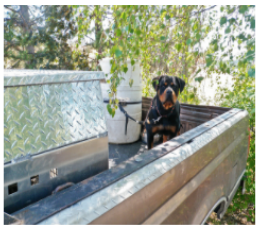
**Dataset**: We conduct our experiments using the challenging MS COCO (2014) dataset (Chen et al., 2015b), which has 82,783 images and was generated using human subjects on the Amazon Mechanical Turk (AMT). We used the NLTK tokenizer for the captions and limit the vocabulary to include words that occur at-least 10 times. The final vocabulary size was 8855. We do not repartition the training and validation sets for MS COCO to increase the training data since we wanted to test the ability of the models to generalize to novel combinations.
**Evaluation Metrics**: We report the performance of our proposed approach as well as the baseline models using BLEU, a measure that loosely corresponds to precision of word n-grams between hypothesis and reference sentences. Additionally we also report the results based on METEOR, ROUGE and CIDEr.
**Hyperparameters**: Hyperparameters across all our experiments were kept constant. $z$, $z_d$ and $z_c$ were fixed to 128 dimensions. 512 dimensions were used for hidden. Adam was used for optimization with a learning rate of 0.001. Epsilon value of 1e-12 was used for Gumbel argmax. We use minimal KL-annealing with logistic function between 300 and 2500 steps. All our models use greedy decoding (beam size=1).

## 16.4 Analysis

We observe that our proposed approach of using both continuous and discrete variables for representing latent space has consistent gains across different metrics, as compared to the baseline

Gold: a traffic light under a cloudy blue sky
RNN: **a black and white photo of a man on a skateboard**
VED: **a man standing on top of a lush green field**
CCVED: a **partially yellow** and grey **traffic** light near a house

Gold: a man sitting on a bench with a tall building behind him
RNN: **a man sitting on a bench** with a bird on it
VED: **a cat sitting on top of a bicycle near a traffic light**
CCVED: a **person sitting down near a stone structure**



Gold: a black and white dog sitting on the edge of a patio
RNN: a **woman sitting on a bench with** a **dog**
VED: a **dog** is **sitting** on a **bench in the woods**
CCVED: a **dog** that is **leaning over a piece of wood**

Gold: a couple of buses that are on a empty street
RNN: **a man and woman are walking down the street with umbrellas**
VED: **a train station** has several people riding bikes and cars
CCVED: **the empty bench along a roadway and pillars near the building**

FIGURE 16.3: Examples of generated captions across models (Blue words represent generated concepts that are factual, but not in the gold caption. Green words represent generated concepts that are present in the gold. Red words represent non-factual concepts.)

| N-grams | Gold | RNN Baseline | VED Baseline | CCVED (Ours) |
|---|---|---|---|---|
| a man sitting | 23 | 3716 | 40 | **19** |
| a dog | 400 | **694** | 1010 | 714 |
| a woman sitting | 11 | 2508 | 51 | **21** |

TABLE 16.2: Count of n-grams that appear at start of caption

models (see Table 16.1). RNN based models optimize the likelihood objective via cross entropy loss. This biases the decoder to over-generate n-gram patterns that occur more frequency in training data, leading to non-factual captions. On the contrary, our proposed approach optimizes KL-divergence that outperforms the baseline models in estimating the prior n-gram distribution (see Table 16.2).

Captions generated by our proposed model capture more details than the baseline models (see Figure 16.3). The model is evidently able to disentangle the learnt properties and create new abstract concepts at inference time. As a result, our proposed model generates more diverse and relevant captions compared to the baseline models. For example, the model generates

FIGURE 16.4: Counting errors in generated captions (a) a plate with a sandwich and three sandwiches (b) a number of horses on a beach near the water (c) four guys relaxing on a narrow sofa

FIGURE 16.5: Common sense errors in generated captions. (a) a man in a giraffe has a branch pinned between his ear (b) a black man unk a fish under a framed view of the unk

*stone structure* for describing the *building* in the image. The model is also able to map similar properties to each other. For example, the model learns *leaning* and *sitting* fall into the same semantic space. However, our proposed model is also prone to errors. We observed that our model is weak at counting (see Figure 16.4(a) and (c)). Sometimes, it produces factually correct, but more general words like *many* and *number of* to denote multiple objects in the image (see Figure 16.4(b)). Unfortunately, this is penalized by the evaluation metric. Another shortcoming of our proposed model is its lack of common sense knowledge. This leads to generation of bizarre captions. For example in Figure 16.5(a), the branch in background is visible from in between the giraffes ear, and is not pinned between his ear. In Figure 16.5(b), the model assumes the reflection of a man in black-suit on the window is a black-man standing. Nevertheless the model is able to create novel concepts like *pinned in between something*.

## 16.5 Conclusion

Multimodal problems like caption generation require learning representation across modalities. In this work, we proposed an approach to incorporate joint continuous and discrete representation in the prior distribution. Our model learns better representations, and generalizes well on unseen data. It outperforms baseline models on a multitude of metrics, and is able to generate more detailed, relevant and diverse captions. In future, we would like study this module in other zero-shot learning tasks.

# 17

# Extensions - Visual Question Answering

*Visual Question Answering encompasses multiple modalities viz image and text. Hence, it is challenging to learn rich feature representations that aid this task. Providing textual descriptions of the input image have improved this task, showing the models lack reasoning across modalities. Further, existing SOTA models fail to generalize on unseen samples, revealing they lack compositionality. In this work, we propose three approaches - **JUPITER, VENUS**, and **MARS**, all aimed at compositionality, and in turn improve VQA. Our numbers improve on the baseline scores established by the module networks - 23.31% in number category, 63.93% binary questions and 26.65% accuracy in other kinds of questions. We provide a detailed analysis of our approaches and analyze how they address compositionality better than existing models.*

## 17.1 Introduction

Visual Question Answering (VQA) involves answering a natural language query about an image. Questions can be arbitrary and they encompass many sub-problems in computer vision: (1) Object recognition (2) Object detection (3) Attribute classification (4) Scene classification (5) Counting. VQA is characterized by wide ranging applications from helping visually impaired people through human machine interaction. It has the potential to serve as an effective media content retrieval framework. A primary form of implementing a VQA system would be to use a bucketing approach: by learning image and text features and fusing them to get an answer. In recent years, there have been several extensions to the trivial approach mentioned above (Fukui et al., 2016); (Lu et al., 2016); (Yang et al., 2016); (Lu et al., 2015) claim to learn good

representations of abstract concepts needed to answer questions. However, it has been shown (Agrawal et al., 2017b) that most of the approaches capture surface level correlations and fail to handle unseen novel combinations during test time.

In this work, we investigate approaches to improve compositionality in VQA, where we explicitly focus on learning compositionality between concepts and objects. Language and vision are inherently composite in nature. For example different questions share substructure viz *Where is the dog?* and *Where is the cat?* Similarly images share abstract concepts and attributes viz *green pillow* and *green light*. Hence it is vital not only to focus on understanding the information present across both these modalities, but also to model the abstract relationships so as to capture the unseen compositions of seen concepts at test time. Achieving this would then allow the model to generalize better by learning an inference procedure, resulting in true success on this task.

In this work, we propose three approaches. The first approach ***JUPITER*** - **JU**stification via **P**ointwise combination of **I**mage and **T**ext based on **E**xpected **R**ewards, is built on top of the Neural Module Networks (Hu et al., 2017). This is motivated from our hypothesis that generating captions can provide additional information to improve VQA. Additionally, JUPITER uses Reward Augmented Maximum Likelihood (Norouzi et al., 2016), which is improves caption generation. Our second approach is ***VENUS*** - **V**ariational **E**ntanglement **NU**llification **S**ystem. VENUS is motivated from our hypothesis that latent space of images can be split into discrete objects and continuous relationships between them, and hence can provide better generalizability. We exploit VENUS from an architecture viewpoint. Our third approach is ***MARS*** - **M**ultimodal **A**uto**R**egressive **S**quasher. MARS learns feature representations from different modalities using mutlilevel fusions. We present our results and qualitative analysis on the VQA v1.0 dataset (Antol et al., 2015b).

## 17.2 Related Work

**Visual Question Answering**: (Kazemi and Elqursh, 2017) provided a strong baseline for VQA using a simple CNN-LSTM architecture, and achieved 64.6% on the VQA 1.0 Openended QA challenge. This further proved that the dataset is biased. (Agrawal et al., 2017a) introduced grounding to prevent the model from memorizing this bias. Similarly, (Li et al., 2018) used a zero-shot training approach to improve the generalizabilty of the model, and prevent the model to learn the bias. However, recently (Agrawal et al., 2017d) showed that most models degrade in performance when tested on unseen samples. In this work, we aim to tackle this lack of generalizability.

**Neural Module Networks**: To the best of our knowledge, the work by authors in (Hu et al., 2017) and (Andreas et al., 2015) is the only work so far that explicitly uses a divide and conquer approach for compositionality. Natural language questions are best answered when broken

down into their subparts. The authors use a similar intution and propose a modular architecture. This approach first parses the natural language question into linguistic components. Second, each component is assigned to a sub-module that solves a single task. Lastly, these modules are then composed into an appropriate layout that predicts an answer for each training example. Such a dynamic network not only helps learning object-object relationships well via compositionally, but also improves the reasoning abilities of the model.

**Multitask Learning**: There have been number of works that explore multitask learning as an approach to joint learning of vision and language tasks. In one such work (Johnson et al., 2018), authors learn related regions of the image by simultaneously training three different semantic tasks - scene graph generation, object detection, and image captioning. A multi-task learning architecture was also proposed by (Zhao et al., 2018a) for image captioning where they enable sharing of a CNN encoder and an LSTM decoder between object classification task and the syntax generation tasks. (Ruder, 2017; Lin et al., 2018) show mutlitask learning reduces overfitting in limited-resource settings, and can learn representations to improve downstream (part-of-speech tagging and name-entity recognition) tasks. Our purpose of joint training in multitask learning is to provide regularization on the learned features for VQA, with an added benefit of achieving better performance on the auxiliary task (of generating captions).

**Incorporating additional knowledge**: In (Chandu et al., 2018) authors show that incorporating captions helps resolve some ambiguities in visual question answering. In (Aditya et al., 2018) authors first obtain captions and then use them for improving VQA via the framework of predicate logic. In (Wu et al., 2016) authors learn attributes from an image using an image labeling and then query using an external knowledge base.

## 17.3 Proposed Approaches

In this section, we first present an overview of each of our proposed approaches, and then describe the optimization being performed in detail. Based on our understanding of the task, we believe that adding captions helps VQA since there is usually a mapping from the distribution of captions to that of answers. Specifically, we hypothesize that captions have the potential to act as horizontal enabling module within the framework of NMN. To test this hypothesis, we propose three approaches, each of which incorporate captions at different levels in the typical pipeline for VQA. Specifically, we present

- *JUPITER(Justification by Pointwise combination of Image and Text based on Expected Rewards)* - an approach that involves manipulation of the loss function.
- *VENUS(Variational Entanglement Nullification System)* - an approach that exploits models that explicitly allow us to encode compositionality.
- *MARS(Multimodal AutoRegressive Squasher)* - an approach that attempts to learn a better multimodal joint representation.

### 17.3.1 JUPITER - Justification by Pointwise combination of Image and Text based on Expected Rewards

The key motivation of this approach [depicted in Figure: 17.1] was to manipulating the loss function to account for captions. We hyothesize that explicitly accounting for captions in the loss function will affect the downstream VQA predictions. Figure 17.1 shows the framework architecture and functioning.

#### 17.3.1.1 Model Description

Our model uses Neural Module Networks (NMN), along with multiple proposed extensions. More specifically, we use the following extensions:

- *Multitask Learning*: We modify the decoder to perform multiple tasks namely, caption generation and VQA. We use the attention grid generated by *'Find'* module in the NMN, the encoded question layout, and the input image to generate captions in an auto regressive. Our hypothesis is that using this conditioning, we can force the model to generate attention grid that is suitable to both downstream tasks, in turn improving VQA performance.

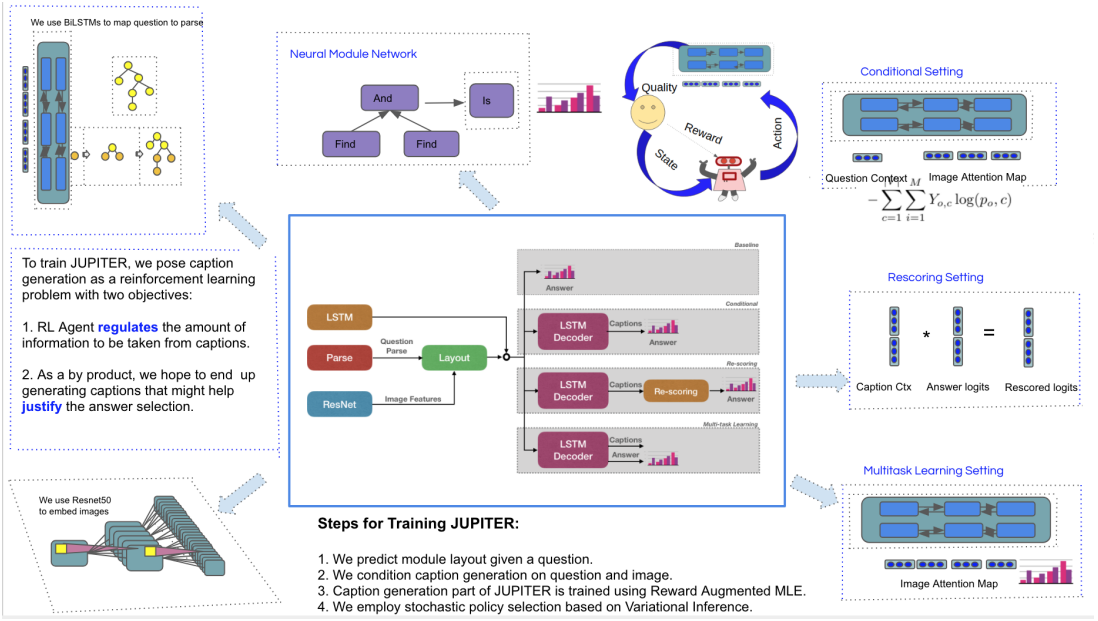



FIGURE 17.1: Justification by Pointwise combination of Image and Text based on Expected Rewards

- *Conditional Generation*: As opposed to multitask learning approach, in this extension we explicitly provide the generated captions as input to VQA decoder. More specifically, we train the model to first generate a relevant image caption using previously defined setup. Next we condition the answer decoder on the generated caption. The intuition is that

providing the model with information more explicitly will help to predict answers based on this information.

- *Re-weighting*: In this extension, we re-weight the answer hypothesis using the generated caption. We hypothesize that this will help the model to disambiguate between answer logits that have maximum entropy.

- *M-Hybrid and C-Hybrid*: In order to harvest complimentary benefits from our primary extensions, we also implemented two hybrid systems. M-Hybrid extension combined multitask learning and re-weighting approach, and the C-Hybrid extension combined conditional generation and re-weighting approach.

- *Reinforcement Learning*: This extension uses Reward Augmented Maximum Likelihood (RAML) as opposed to Maximum Likelihood (MLE) for generating captions. The intuition for this extension was to enable the agent to generate captions that will help the model to answer the given question. More specifically, the agent at each caption generation step can perform one of the two tasks: (1) Generate next word for the captions or (2) Answer the question based on caption generated so far. The agent is rewarded based on VQA accuracy. Since training with REINFORCE is known to be unstable, we use a baseline wherein we generate answers based on the final hidden state of a deocder trained using MLE.

#### 17.3.1.2 Learning

We denote input question as $Q$ and input image as $I$. $L^*$ denotes the gold layout for $Q$ and $C^*$ is gold caption for $I$. We denote $L$ as the layout generated by NMN for $Q$. $C$ is caption generated from JUPITER. We denote answer classes by $y$ and the correct answer class by $y^*$. $T$ is the training data samples of type ($I$, $Q$, $y^*$). Next, we describe the objective function for each extension in detail.

- *Multitask Learning*: We use a two-part objective function for multitask learning. The first part is generating captions from the input and the second is generating answer logits from the input and the generated NMN layout.

$$L(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L) + logP_\theta(C|I,Q) \tag{17.1}$$

- *Conditional Generation*: This extension uses a similar objective function. However, we generate answer logits from the input, generated NMN layout as well as the generated captions.

$$L(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L,C) + logP_\theta(C|I,Q) \tag{17.2}$$

- *Re-weighting*: This extension uses a similar objective as conditioned generation. Further, for re-weighting we define new answer logits $y$'.

$$y' = C_T y \tag{17.3}$$

where, $C_T$ is the final hidden state of generated caption, and y is the previous answer logits. The updated objective function is:

$$L(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y'|I,Q,L,C) + logP_\theta(C|I,Q) \tag{17.4}$$

- *Reinforcement Learning*: The agent transitions between generating next word in the caption and generating final answer. The agent receives minibatch VQA accuracy as its reward. The Baseline we use to stabilize the training and the expected reward of our agent respectively are expressed as

$$L_{baseline}(\theta) = \sum_{(I,Q,y*)\in T} logP_\theta(y|I,Q,L,C) \tag{17.5}$$

We use cross-entropy loss to train the model. We jointly train our captions module in JUPITER alongside NMN, which learns a question layout $L$. The details of the layout loss have been described in Section 4.2.

### 17.3.2 VENUS - Variational Inference based Entanglement Neutralization System

In this approach [depicted in Figure 17.2], we present an architecture modification for VQA. We investigate a latent stochastic model that explicitly provides a framework to achieve compositionality - the ability to decompose an entangled multimodal representation into its constituent disentangled representations.

#### 17.3.2.1 Model Description

We model VQA as a Bayesian formulation, where image and question is the observed variable (**X**), and answer is output variable (**Y**). We hypothesize that inferring the answer requires understanding the latent relationship between different entities in the image. This relationship can be formulated as latent variables (**Z**). Further, the relation between entities are continuous in space and the entities themselves are discrete. To account for this, we factorize the latent representation into discrete and continuous components ($Z_c$, $Z_d$). This approach is inspired by (Zhou and Neubig, 2017c). We first extract visual and textual representations and concatenate them. We then infer a joint latent representation which is reparameterized and decoded into

a softmax over the answer categories. We split the latent prior space into continuous and discrete components. To generate captions in this framework, we employ $\beta$ VAE and augment the channel capacity term. We also split the channel capacity into continuous and discrete terms.

#### 17.3.2.2 Learning

We use a latent stochastic model, more specifically a variational encoder decoder network. The encoder approximates a posterior in the form of second order statistics of the input data. A latent representation is then sampled from continuous and discrete prior. We concatenate both the samples and pass it to the decoder, which is trained to maximize the likelihood of the input data. Our model optimizes the following variational lower bound:

$$E_{q_\phi(z|x,c)}[logp_\theta(x|c,z)] - \beta|D_{KL}(q_\phi(z|x,c)||p_\theta(z|c)) - C_z| \tag{17.6}$$

where $C_z$ is the channel capacity term (Burgess et al., 2018b). The first term in ELBO is the reconstruction error while the second is the divergence between approximate and true posteriors.

Optimizing the latent space is a trade-off between the ability to reconstruct the input and the capacity to disentangle the latent factors of variation. This loss is a sum of KL-divergence that tries to disentangle the latent causal factors of data, and cross entropy loss which is a substitute of reconstruction loss. The parameters are optimized using gradient descent and re-parameterization suggested by (Kingma and Welling, 2013).

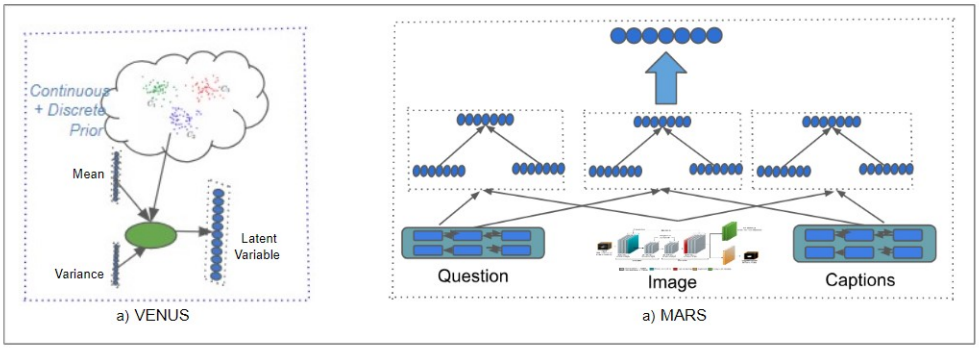



FIGURE 17.2: Proposed approaches - VENUS (left) and MARS (right)

### 17.3.3 MARS - Multimodal Autoregressive Squasher

In this approach [depicted in Figure 17.2] we hypothesize that learning stronger joint multimodal representation implicitly helps accomplish compositionality. VQA has two modalities viz. vision and text. Textual modality has two input streams viz. captions that describe the image and question. The caption can be considered as another view of the information present

in the images. This analogy allows us to follow the multimodal polynomial fusion (MPF) architecture (Du et al., 2018).

#### 17.3.3.1 Model Description

We use non linear activation to map features from multiple data streams to create a weighted sum of the intermodal interactions, and generate a fused representation. More specifically, given vector representations $h_c$, $h_q$, and $h_i$ that belong to captions, question and image features respectively, we learn first-order relations $h_{cq}$, $h_{qi}$ and $h_{ic}$ where each of them are formed by selecting two different streams. Next, $h_{fusion}$ is obtained by interaction between the first order relations. Since images do not have time-step information, we interpolate their representation to match the desired length and randomly perturb each interpolation. The intuition behind this is that perturbation might act as a regularizer. In this context, desired length is *max (caption_length, question_length)*

#### 17.3.3.2 Learning

We transform the feature vectors to perform element-wise product (the Hadamard product) across different streams. Each interaction is derived by an element-wise product of features. The three equations below show interactions between different modalities along with learnable parameters that control the flow of information. They capture the first order relations. The fourth equation capture the interactions between first order relations.

$$h_{cq} = (\gamma_{00} * h_c) \odot (\gamma_{10} * h_q) \tag{17.7}$$

$$h_{qi} = (\gamma_{11} * h_q) \odot (\gamma_{20} * h_i) \tag{17.8}$$

$$h_{ic} = (\gamma_{21} * h_i) \odot (\gamma_{01} * h_q) \tag{17.9}$$

$$h_{fusion} = \alpha_0 * ((\beta_0 * h_{cq}) \odot (\beta_1 * h_{qi}) \odot (\beta_2 * h_{ic})) \tag{17.10}$$

## 17.4 Experimental Setup

### 17.4.1 Dataset and Input Modalities

VQA dataset by (Antol et al., 2015b) has 265016 images, 614163 questions. The dataset consists of 82,783 training, 40,504 validation, and 40,775 test images. Each image has 3 questions on average and 10 ground truth answers. Questions as well as answers are open ended, accounting for a more real-world scenario. The questions are rich in a way, as the require the model to have complex reasoning and understanding abilities.

### 17.4.2 Multimodal Baselines

We built three baseline models to compare the performance of our proposed approaches. It is important to note that these baselines are strong models by themselves. These are outlined below:

**Neural Module Network**: The NMN (Hu et al., 2017) formulates VQA as a classification task, where the model uses the input image and question to learn a predictive distribution over the answers. The authors define a set of modules that can be used to solve primitive tasks. For a given input question, these modules are combined to generate a layout. This layout is learned by an RL agent trained on groud truth parses. The layout is used to predict an answer from the image. The model is trained end-to-end by minimizing two losses: log likelihood of the generated layout given a question's parse, and the cross entropy loss from the predicted answer. In our baseline, we use the same set of modules and training strategy as the original paper by (Hu et al., 2017; Andreas et al., 2016a). We use this baseline to compare the performance our JUPITER approach.

**Fusion based Baseline Models**: We built two baselines for VENUS and MARS approach. They are based on concatenation of the input representations. More specificially, *RNN* uses an LSTM to encode the image and question, and then generate an affine transformation to predict the answers. *VED* uses a similar encoding strategy, with reparameterization before the decoder predictions.

### 17.4.3 Experimental methodology

This section describes the configurations of our models. The hyperparameters we use for our experiments are consistent across baseline and proposed models. For our experiments, we use the original VQA v1.0 train-val-test split. We use embedding dimension of 128, hidden dimension of 256, and latent dimension of 256 for VQA and caption generation experiments. The vocabulary size is caped at 8853. We use Adam with learning rate 0.001. We clip the gradients at 25. To handle KL collapse in latent stochastic models, we anneal the KL term using a sigmoid function with step size at peak of 2500 and the exponent factor of 0.0025. We use $\beta$ value of 150.

## 17.5 Results and Discussion

In this section, we discuss the results from our proposed approaches viz. JUPITER, VENUS and MARS, and compare them against our baselines. Table 17.1 consolidates the results of our experiments. To better understand the performance of these models, we report the performance across different answer categories namely, Number, Yes/No and Other. The overall best baseline model for VQA is NMN by (Hu et al., 2017).

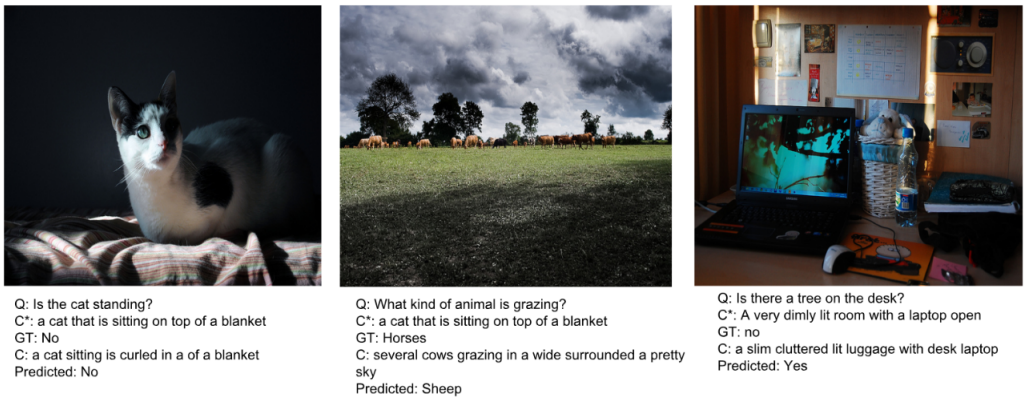



FIGURE 17.3: Qualitative Analysis from JUPITER: left image depicts a scenario where generating caption helped the model in selection of the right answer. Image in the center depicts a scenario where captions end up confusing the model. Image in the right most highlights an interesting scenario where the generated caption seems irrelevant.

### 17.5.1 Results: Baseline Models

The input to our baseline models is the image and the question. We do not use any external knowledge. Our results show that the baseline models have highest accuracy on the Yes/No questions. However, the Number type questions often require deeper understanding of the image, and so our baselines have lowest performance on them. Humans tend to have low agreement for Yes/No questions. We attribute this to question ambiguity or missing information in the image.It has to be noted that our implementation of the NMN baseline achieves better scores compared to the open source original implementation. This can be attributed to the presence of additional modules in our implementation, specifically OR, COUNT, FILTER, and EXIST modules.

| Model | System | Input | Number | Yes/No | Other |
|---|---|---|---|---|---|
| Human | Best | Image + Question | 83.39 | 95.77 | 72.67 |
| Human | Worst | Image + Question | 65.28 | 46.52 | 78.02 |
| NMN | Baseline (Replicated) | Image + Question | 23.31 | 63.93 | 26.65 |
| NMN | Baseline (Our implementation) | Image + Question | 26.35 | 64.49 | 31.55 |
| RNN | Baseline | Image + Question | 19.34 | 57.82 | 17.77 |
| VED | Baseline | Image + Question | 17.76 | 58.00 | 10.43 |
| RNN | MARS | Image + Question + Caption* | 23.09 | 57.88 | 18.22 |
| RNN | MARS | Question + Caption* | 21.72 | 57.95 | 21.59 |
| VED | VENUS | Image + Question + Caption* | 19.25 | 57.83 | 10.10 |
| VED | VENUS | Question + Caption* | 18.13 | 58.10 | 10.33 |
| NMN | M-Hybrid | Image + Question + Caption | 26.31 | 64.27 | 30.43 |
| NMN | C-Hybrid | Image + Question + Caption | 27.48 | 65.8 | 32.2 |
| NMN | JUPITER | Image + Question + Caption | 32.82 | 67.95 | 33.15 |

TABLE 17.1: Results from human, baselines and proposed approaches. * denotes systems that employ Gold captions

### 17.5.2 Results: Proposed Models

Looking at the objective evaluation results from table 17.1, it is clear that incorporating captions leads to improvements across the approaches. This result empirically validates our hypothesis related to captions: Captions help VQA. To understand the extent of this, we have also performed ablation analysis wherein we have used just captions to answer the question ignoring the input image. Surprisingly, systems built in this fashion seem to perform better than our baselines. This leads to an interesting observation: *Captions seem to contain supplementary and in some cases complementary information to the images themselves.* However, we acknowledge that proving such hypothesis would require additional experimentation. For instance, it would be interesting to perform similar ablation analyses employing computationally more powerful frameworks such as attention as baselines or adding more visual information such as ground truth bounding boxes. It is also interesting to note that the proposed approaches achieve better scores compared against the *worst* human performance in Yes/No category.

Our approach JUPITER outperforms all other approaches across all the categories. In addition, within the models employing module networks, the system employing reinforcement learning outperforms other approaches. This is in line with our hypothesis related to Reward Augmented Maximum Likelihood and raises interesting questions related to *comparison between supervised approaches such as Maximum Likelihood and their reward based reinforcement counterparts.* It would be interesting to perform a much larger scale evaluation comprehensively comparing the effectiveness of these approaches in the context of downstream tasks. In figure 17.3, we present some scenarios that highlight the way captions get utilized for answering question about the corresponding images.

## 17.6 Conclusion and Future Direction

VQA is challenging as it requires deep understanding of the image and question, as well as learn rich representations to reason across them. In this work, we have primarily focused on incorporating external knowledge in the form of captions to aid VQA. We established three baselines for this task - each focused on improving the loss function, the model architecture and the feature representation respectively. With a similar motivation, we also proposed three approaches JUPITER, VENUS and MARS, that also incorporate captions as a horizontal enabling module. All three approaches outperformed their corresponding baselines. Our strongest approach was the C-Hybrid strategy in JUPITER, alongside the RAML strategy. This formulation, we believe, also has the potential generate justifications to the generated answers. We would like to pursue this direction and perform experiments that help us manipulate the reward mechanism. For instance, one promising approach is to introduce a delay penalty factor that assigns a negative reward for every word in caption that the agent chooses to generate without generating the

answer. This constraint might force the agent to generate interpretible and relevant image descriptions that help the model answer the question as opposed to generic descriptions that are not helpful.

Moving ahead, we would also like to further explore disentanglement. Disentanglement transcends to tasks beyond VQA, and has the potential to improve generalizability for them. Motivated from (Higgins et al., 2018), we would like to include disentanglement to solve other tasks that involve multiple modalities.

# 18

# Conclusions

## 18.1 Summary of Contributions

This dissertation is organized into 4 parts.

- **Part 1**: I introduce the framework and motivations behind it. I then provide a detailed explanation of De-Entanglement. In chapter 3, I describe 'FALCON', a toolkit I am developing as part of my PhD and used to perform experiments in this dissertation.
- **Part 2**: I address the challenge of 'Scalability'. To demonstrate the effectiveness of De-Entanglement, I present experiments first on De-Entanglement of content using Blind source separation as task(chapter 5). I then present experiments from De-Entanglement of style using code switching(chapter 4).

  I highlight two scenarios that the challenge of scalability poses: conversational speech and multilingual conditions. In my dissertation I am specifically limiting my self to code switching which is related to both these scenarios. I present experiments that show joint De-Entanglement of content and style helps build systems capable of effectively handling code switched inputs(chapters 6 and 7).
- **Part 3**: I address the challenge of Flexibility. I first present De-Entanglement of style by using detection of paralinguistic information from speech. Specifically, I present experiments that use utterance level representations in chapter 9 and approach that learns representation based on divergence in chapter 10. In chapter 11 I will present

De-Entanglement of content using priors with Acoustic Unit Discovery as the target application. I employ the extracted units to accomplish Voice Conversion to prove that the learnt units address Flexibility.

- **Part 4**: In this part I address the challenge of Explainability. I posit that explainable speech technologies should be characterized by two properties: (a) Reasonable Understanding of internal mechanisms in the model and (b) Demonstrable Utility of the model for downstream applications. Using acoustic unit discovery (chapter 15), I present experiments to show that we can inject reasonable priors into the model architecture. Using acoustic intent recognition, I show that such a model that we have reasonable understanding can be reliably employed for a downstream application under a variety of scenarios.

## 18.2 Other Contributions

- **Speech Recognition**: I have shown (Rallabandi et al., 2018b) that different code switched data is manifested in different styles. I divide the data based on style information based on two categories - count based and span based. I show that such an approach helps in improving the perplexity of code switched language models. I have also worked on building speech recognition for low resource languages (Duan et al., 2019)
- **Source Separation**: Using Variational AutoEncoder as the model, we have shown an approach (chapter 5) that provides a knob to control source separation.
- **Voice Conversion**: I have demonstrated that Voice Conversion can be realized employing the model VACONDA(chapter 11). Since the discrete latent units from the model resemble articulatory features, they can be utilized to realize cross gender and cross lingual conversion of voice characteristics.
- **Unsupervised Slot Identification**: We have presented a case study to identify slots in an unsupervised fashion(chapter 14). Such an approach is extremely helpful in the context of zero and very low resource scenarios.
- **Sentiment Analysis**: We have shown that code mixing ratio can be employed as the style information to design a self training based sentiment analysis system (Gupta et al., 2021a) in the context of code switching. We have also demonstrated transfer learning based approach to improve code switched sentiment analysis systems (Gupta et al., 2021b).
- **Parts of Speech Tagging and Named Entity Recognition:** using switch point location as the style information, we have shown that large language models can be re purposed to accomplish a number of downstream tasks by employing Parts of Speech Tagging and Named Entity Recognition in low resource scenarios as the target applications(Chopra et al., 2021a).

- **Image Captioning and Visual Question Answering**: To show that the framework can be extended to other modalities, I present experiments showing De-Entanglement from Image captioning(chapter 16. We employ a variant of Variational Auto Encoder that splits the latent space into both continuous and discrete variables. We use the discrete variables to map the individual objects in the image and continuous variables to map the relationships between the objects. Using Visual Question answering as the task and image captioning developed in chapter 16 as the auxiliary task, I present a case study to show that this mix of tasks has the potential to address 'Justification'.

## 18.3 Future Directions and Extensions

### 18.3.1 Identification of Restricted speakers

Applications related to speech technology have grown rapidly over the last decade and span both human-human as well as human-machine interactions. Given this impact, there has been a rise in interest in the research community towards building 'failsafes' against possible abuse of such advances (Wu et al., 2015a, 2017). Multitarget speaker detection and identification challenge (Shon et al.) is another evaluation in this vein. The aim of this challenge is two fold: (1) Given a test utterance, detect if the utterance was possibly spoken by a speaker from 'restricted cohort'. (2) If indeed the utterance is from the cohort, identify the speaker.

The challenge(Shon et al.) aims to evaluate how well current machine learning algorithms are able to detect a restricted set of speakers. This task has semblance to a widely deployed module in any sophisticated email program: spam detection. Having said that, spam detection is based on the 'content' of the message while the objective of current challenge is to detect the 'transmitter' of message: speaker. The data used in this version of challenge is from telephone conversations. This means there is a domain match with deployment conditions and hence, provides us relatively more reliable way to interpret (final)model behavior.

### 18.3.2 Transfer Learning and Self Training for Code Switched Data

Style or content information as proposed in this dissertation could be employed to control self training in under resource scenarios. I posit that such bias based augmentation helps models in the context of under resourced scenarios. To demonstrate this, we have performed experiments across different tasks within NLP. Specifically, we employed code switching token ratio to self train code switched sentiment analysis models in zero shot and few shot settings (Gupta et al., 2021a,b) and switch point information to improve Named Entity Recognition and Parts of Speech Tagging(Chopra et al., 2021b) models.

### 18.3.3 Confidence Measures as measure of reliability of the Model

Style or content information could be employed to estimate the confidence of a model in addition to the prediction being made. This is especially useful in the context of architectures that employ an auxiliary task, as presented in the chapters . Employing a model internal representation as a proxy for confidence could help design stronger self training models and prevent the bias effect in extreme low resource scenarios.

## 18.4 Extensions - Development of JUDITH

JUDITH is being built as an in house personal assistant within FALCON. The overarching goal of JUDITH is to facilitate building an intelligent agent like Jarvis aimed at accompanying and facilitating high quality research. JUDITH has support for basic functionality like logging the experiment status. In addition to the basic functionality, JUDITH currently has the following advanced capabilities:

### 18.4.1 Monitoring Experiment Status

I will employ Text to Speech(TTS) system as an example application to demonstrate this capability. Most of the TTS systems today are built in an end to end fashion and involve attention in the acoustic model. It is an informal understanding among researchers building acoustic models that the objective value of the divergence, implemented as L1 distance between the predicted and original spectrogram does not provide useful information to judge the model convergence. Model convergence is typically inspected by subjective evaluation and visualizing attention plots. It has to be noted that attention provides a faster way among the two approaches since alignment typically appears within 6 to 7 percent within model training. Therefore, while training multiple models, inspecting attention helps one quickly evaluate and comment about model convergence instead of waiting for subjective evaluation after model is fully trained. JUDITH exploits this observation in the *Monitoring* capability. Specifically, JUDITH inspects the attention variable early in the training and alerts the user in case a particular experiment appears to be challenging in terms of convergence.

This capability is implemented using a pretrained ResNet. The attention plot during model training is first passed through pretrained model to obtain (2048 X 14 X 14) dimensional feature vector. JUDITH then performs mean reduction of the additional dimensions to obtain a 2048 dimensional feature vector. This representation is then classified to one of the three classes indicating the model convergence. Since model training within FALCON follows graph based modular architecture, JUDITH has access to all the architectures that use blocks similar to the current model. This information can be additionally used as conditional information to the classifier. An example depicting this capability is shown in figure 18.1.

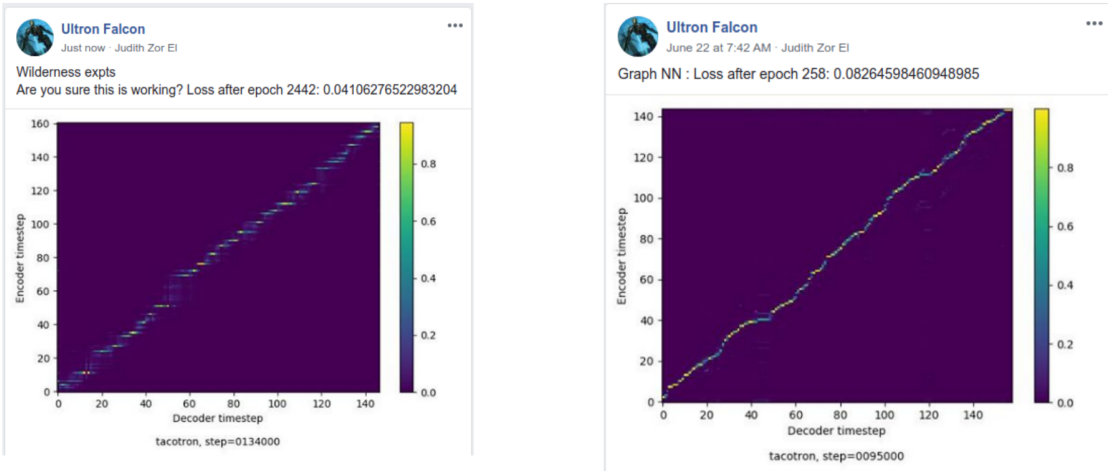



FIGURE 18.1: Example posts illustrating Monitoring capability of JUDITH
(a) Attention plot not a clear indication of model convergence and (b) Clear and sharp attention indicating model training on a successful trajectory. The user is alerted with a message to inspect the training more closely when JUDITH is not confident (can be seen in (a))

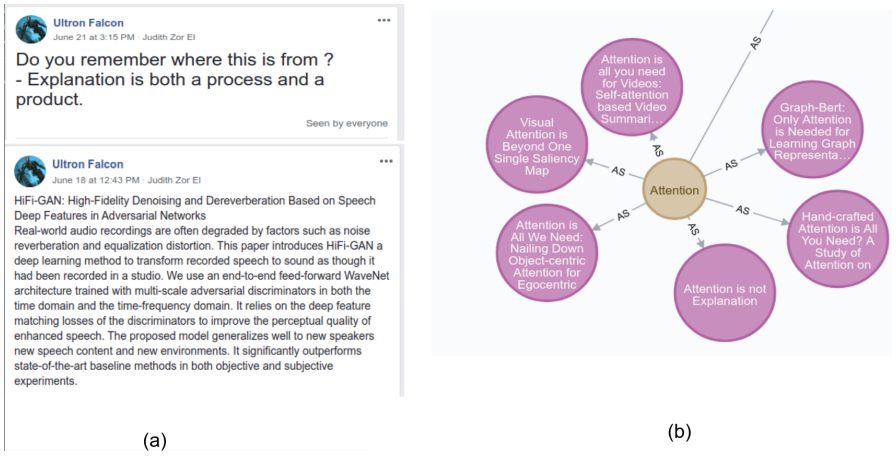



FIGURE 18.2: Figure illustrating literature review and recommendations capability of JUDITH. Each paper is encoded as a node and is related to a super topic. (a) Review: JUDITH extracts quotes from the papers and periodically sends them to a predetermined location(above). If unable to extract quotes from the paper within a pre set confidence level, JUDITH sends the abstract for review and manual annotation of quotes from the paper(below). (b) Recommendation: In this context, Attention(in the image) and Explanation(not in the image) are the super nodes. Each paper related to these super nodes is encoded as a node.

### 18.4.2 Literature Review and Recommendations

I will employ a segment of literature review related to 'Explainability' within this Dissertation to demonstrate this capability. Each block within FALCON is annotated with its source material in two forms: (a) code source or (b) literature source. These sources are encoded as nodes and are matched to *concepts* as super nodes. For instance, the query 'Attention is not not an Explanation' is encoded as a node while 'Attention' is encoded as a super node. JUDITH periodically populates nodes crawling literature from various sources including ArXiv and Google scholar to help research topic visualization.

This capability is implemented by using pretrained BERT model from (Wolf et al., 2019). The abstract from each literature entry is classified into supernodes using a multi label classifier implemented using linear layers stacked on top of the BERT encoder. JUDITH uses 8 attention heads and 12 layers of self attention before the linear layers. The text is fed to the model at the word level. A visualization of this capability can be seen in figure 18.2.

## 18.5 Broader Impact beyond Natural Language Processing

Thus far I have described the aplication of De-Entanglement towards NLP. Now I would like to mention a couple of applications of the ideas proposed in this dissertation outside the speech research community.

### 18.5.1 Marine Acoustics

The orca whales belong to oceanic dolphin family. Due to the threat associated with them, it is important to detect and monitor their activity. The orcas have vocal behavior and therefore their detection is studied as a topic of underwater acoustics. Using the framework of De-Entanglement, style information from acoustics can be employed for Orca detection.

### 18.5.2 Baby Sounds Detection

Classification of infant and parent vocalizations helps in understanding how infants learn to regulate emotions while growing up. Seemingly trivial infant outbursts such as cry, fuss, laugh, babble, and screech have the potential to convey meaningful information to parents; for instance, a loud and harsh cry signals hunger while burbling laugh signals satisfaction. Using the framework of De-Entanglement, style information from acoustics can be employed for detection of infant vocalizations.

# Bibliography


https://awb.pc.cs.cmu.edu/.

Jaime C Acosta and Nigel G Ward. 2011. Achieving rapport with turn-by-turn, user-responsive emotional coloring. *Speech Communication*, 53(9-10):1137–1148.

Somak Aditya, Yezhou Yang, and Chitta Baral. 2018. Explicit reasoning over end-to-end neural architectures for visual question answering. *arXiv preprint arXiv:1803.08896*.

Triantafyllos Afouras, Joon Son Chung, Andrew W. Senior, Oriol Vinyals, and Andrew Zisserman. 2018. Deep audio-visual speech recognition. *CoRR*, abs/1809.02108.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2017a. Don't just assume; look and answer: Overcoming priors for visual question answering. *CoRR*, abs/1712.00377.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4971–4980.

Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017b. C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset. *arXiv preprint arXiv:1704.08243*.

Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017c. C-vqa: A compositional split of the visual question answering (vqa) v1.0 dataset. *CoRR*, abs/1704.08243.

Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017d. C-VQA: A compositional split of the visual question answering (VQA) v1.0 dataset. *CoRR*, abs/1704.08243.

Abdellah Agrima, Laila Elmazouzi, Ilham Mounir, and Abdelmajid Farchi. 2017. Detection of negative emotion using acoustic cues and machine learning algorithms in moroccan dialect. In *International Conference on Soft Computing and Pattern Recognition*, pages 100–110. Springer.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Dario Amodei et al. 2016. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA. PMLR.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2015. Deep compositional question answering with neural module networks. arxiv preprint. *arXiv preprint arXiv:1511.02799*, 2.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

Abdul Fatir Ansari and Harold Soh. 2018. Hyperprior induced unsupervised disentanglement of latent representations. *arXiv preprint arXiv:1809.04497*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015a. VQA: visual question answering. *CoRR*, abs/1505.00468.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015b. VQA: visual question answering. *CoRR*, abs/1505.00468.

Gopala Krishna Anumanchipalli, Kishore Prahallad, and Alan W Black. 2011. Festvox: Tools for creation and analyses of large speech corpora. In *Workshop on Very Large Scale Phonetics Research, UPenn, Philadelphia*, page 70.

Yusuf Aytar, Carl Vondrick, and Antonio Torralba. 2016. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900.

Arun Baby. 2006. Resources for Indian languages. In *CBBLR workshop, International Conference on Text, Speech and Dialogue*.

Arun Baby, Nishanti NL, Anju Leela Thomas, and Heam A Myrthy. 2016. Resources for Indian Languages. In *Proceedings of Text, Speech and Dialogue*.

Leonardo Badino, Claudia Barolo, and Silvia Quazza. 2004. Language independent phoneme mapping for foreign tts. In *Fifth ISCA Workshop on Speech Synthesis*.

Leonardo Badino, Claudia Canevari, Luciano Fadiga, and Giorgio Metta. 2014. An auto-encoder based approach to unsupervised learning of subword units. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7634–7638. IEEE.

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

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kalika Bali, Monojit Choudhury, Sunayana Sitaram, and Vivek Seshadri. Ellora: Enabling low resource languages with technology.

Pallavi Baljiker, Sai Krishna Rallabandi, and Alan Black. 2018. An investigation of convolution attention based models for multilingual speech synthesis of indian languages. In *Proceedings of Interspeech*.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ershad Banijamali, Amir-Hossein Karimi, Alexander Wong, and Ali Ghodsi. 2017. Jade: Joint autoencoders for dis-entanglement. *arXiv preprint arXiv:1711.09163*.

Jacob D Bekenstein. 1973. Black holes and entropy. *Physical Review D*, 7(8):2333.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2012. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538.

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2014. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 48–53. ACM.

K Bhuvanagiri and Sunil Kopparapu. 2010. An approach to mixed language automatic speech recognition. *Oriental COCOSDA, Kathmandu, Nepal*.

Kiran Bhuvanagirir and Sunil Kumar Kopparapu. 2012. Mixed language speech recognition without explicit identification of language. *American Journal of Signal Processing*, 2(5):92–97.

Fadi Biadsy, Ron J Weiss, Pedro J Moreno, Dimitri Kanvesky, and Ye Jia. 2019a. Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation. *arXiv preprint arXiv:1904.04169*.

Fadi Biadsy, Ron J Weiss, Pedro J Moreno, Dimitri Kanvesky, and Ye Jia. 2019b. Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation. *arXiv preprint arXiv:1904.04169*.

Alan Black. 2019. Cmu Wilderness Multilingual Speech Dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Alan Black, Paul Taylor, Richard Caley, and Rob Clark. 1998a. The festival speech synthesis system.

Alan Black, Paul Taylor, Richard Caley, and Rob Clark. 1998b. The festival speech synthesis system.

Alan W Black. 2006a. Clustergen: A statistical parametric synthesizer using trajectory modeling. In *Ninth International Conference on Spoken Language Processing*.

Alan W Black. 2006b. Clustergen: a statistical parametric synthesizer using trajectory modeling. In *Proceedings of Interspeech*.

Alan W Black and John Kominek. 2009. Optimizing segment label boundaries for statistical speech synthesis. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 3785–3788. IEEE.

Alan W Black, Kevin Lenzo, and Vincent Pagel. 1998c. Issues in building general letter to sound rules.

Alan W Black and Prasanna Kumar Muthukumar. 2015. Random forests for statistical speech synthesis. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Bajibabu Bollepalli, Lauri Juvela, and Paavo Alku. 2018. Speaking style adaptation in text-to-speech synthesis using sequence-to-sequence models with attention. *arXiv preprint arXiv:1810.12051*.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Darshana Buddhika, Ranula Liyadipita, Sudeepa Nadeeshan, Hasini Witharana, Sanath Javasena, and Uthayasanker Thayasivam. 2018. Domain specific intent classification of sinhala speech data. In *2018 International Conference on Asian Language Processing (IALP)*, pages 197–202. IEEE.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018a. Understanding disentangling in Beta VAE. *arXiv preprint arXiv:1804.03599*.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018b. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. *arXiv preprint arXiv:1909.05358*.

Nick Campbell. 2001. Talking foreign. In *Proc. Eurospeech*, pages 337–340.

Alex de Carvalho, Angela Xiaoxue He, Jeffrey Lidz, and Anne Christophe. 2019. Prosody and function words cue the acquisition of word meanings in 18-month-old infants. *Psychological science*, 30(3):319–332.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *CoRR*, abs/1810.00069.

Joyce YC Chan, PC Ching, Tan Lee, and Helen M Meng. 2004. Detection of language boundary in code-switching utterances by bi-phone probabilities. In *Chinese Spoken Language Processing, 2004 International Symposium on*, pages 293–296. IEEE.

Khyathi Raghavi Chandu, Mary Arpita Pyreddy, Matthieu Felix, and Narendra Nath Joshi. 2018. Textually enriched neural module networks for visual question answering. *arXiv preprint arXiv:1809.08697*.

Khyathi Raghavi Chandu, Sai Krishna Rallabandi, Sunayana Sitaram, and Alan W Black. 2017. Speech synthesis for mixed-language navigation instructions. *Proc. Interspeech 2017*, pages 57–61.

Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. 2017a. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. *arXiv preprint arXiv:1712.02051*.

Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. 2017b. Show-and-fool: Crafting adversarial examples for neural image captioning. *CoRR*, abs/1712.02051.

Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. 2018a. Attacking visual language grounding with adversarial examples: A case study on neural image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2587–2597, Melbourne, Australia. Association for Computational Linguistics.

Ling-Hui Chen, Tuomo Raitio, Cassia Valentini-Botinhao, Zhen-Hua Ling, and Junichi Yamagishi. 2015a. A deep generative architecture for postfiltering in statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(11):2003–2014.

Shizhe Chen, Jia Chen, and Qin Jin. 2017c. Generating video descriptions with topic guidance. *CoRR*, abs/1708.09666.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018b. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2615–2625.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015b. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yi-Chen Chen, Zhaojun Yang, Ching-Feng Yeh, Mahaveer Jain, and Michael L Seltzer. 2020. Aipnet: Generative adversarial pre-training of accent-invariant networks for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983. IEEE.

Yuan-Ping Chen, Ryan Price, and Srinivas Bangalore. 2018c. Spoken language understanding without speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6189–6193. IEEE.

Yutian Chen et al. 2018d. Sample efficient adaptive text-to-speech. *arXiv preprint*.

Hyeong-Seok Choi, Changdae Park, and Kyogu Lee. 2020. From inference to generation: End-to-end fully self-supervised generation of human face from speech. *arXiv preprint arXiv:2004.05830*.

Yeunju Choi, Youngmoon Jung, Younggwan Kim, Youngjoo Suh, Hoirin Kim, et al. 2018. An end-to-end synthesis method for korean text-to-speech systems. *Phonetics and Speech Sciences*, 10(1):39–48.

Parul Chopra, Sai Krishna Rallabandi, Alan W Black, and Khyathi Raghavi Chandu. 2021a. Switch point biased self-training: Re-purposing pretrained models for code-switching. *arXiv preprint arXiv:2111.01231*.

Parul Chopra, Sai Krishna Rallabandi, Alan W Black, and Khyathi Raghavi Chandu. 2021b. Switch point biased self-training: Re-purposing pretrained models for code-switching. *arXiv preprint arXiv:2111.01231*.

Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord. 2019a. Unsupervised speech representation learning using wavenet autoencoders. *arXiv preprint arXiv:1901.08810*.

Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord. 2019b. Unsupervised speech representation learning using wavenet autoencoders. *arXiv preprint arXiv:1901.08810*.

Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee. 2018. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. *arXiv preprint arXiv:1804.02812*.

M Choudhury, G Chittaranjan, P Gupta, and A Das. 2014. Overview and datasets of fire 2014 track on transliterated search. In *Pre-proceedings 6th workshop FIRE-2014*.

Min Chu, Hu Peng, Yong Zhao, Zhengyu Niu, and Eric Chang. 2003. Microsoft Mulan-a bilingual TTS system. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Wei Chu and Zoubin Ghahramani. 2005. Gaussian processes for ordinal regression. *J. Mach. Learn. Res.*, 6:1019–1041.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988.

Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. 2018. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *Journal of Machine Learning Research*, 19(41):1–42.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. 2018. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems*.

Margaret Deuchar, Peredur Davies, Jon Herring, M Carmen Parafita Couto, and Diana Carter. 2014. Building bilingual corpora. *Advances in the Study of Bilingualism*, pages 93–111.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. 2020. Guided variational autoencoder for disentanglement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7920–7929.

Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Yulun Du, Alan W Black, Louis-Philippe Morency, and Maxine Eskenazi. 2018. Multimodal polynomial fusion for detecting driver distraction. *Interspeech 2018*.

Mingjun Duan, Carlos Fasola, Sai Krishna Rallabandi, Rodolfo M Vega, Antonios Anastasopoulos, Lori Levin, and Alan W Black. 2019. A resource for computational experiments on mapudungun. *arXiv preprint arXiv:1912.01772*.

Janek Ebbers, Jahn Heymann, Lukas Drude, Thomas Glarner, Reinhold Haeb-Umbach, and Bhiksha Raj. 2017. Hidden markov model variational autoencoder for acoustic unit discovery. In *INTERSPEECH*, pages 488–492.

Arthur Eddington. 2012. *The nature of the physical world: Gifford lectures (1927)*. Cambridge University Press.

Albert Einstein, Boris Podolsky, and Nathan Rosen. 1935. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777.

Albert Einstein and Nathan Rosen. 1935. The particle problem in the general theory of relativity. *Physical Review*, 48(1):73.

Vanessa Elias, Sean McKinnon, and Ángel Milla-Muñoz. 2017. The effects of code-switching and lexical stress on vowel quality and duration of heritage speakers of spanish. *Languages*, 2(4):29.

Naresh Kumar Elluru, Anandaswarup Vadapalli, Raghavendra Elluru, Hema Murthy, and Kishore Prahallad. 2013. Is word-to-phone mapping better than phone-phone mapping for handling english words? In *ACL (2)*, pages 196–200.

Roberto Emparan, Alessandro Fabbri, and Nemanja Kaloper. 2002. Quantum black holes as holograms in ads braneworlds. *Journal of High Energy Physics*, 2002(08):043.

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem van de Meent. 2018a. Structured disentangled representations. *stat*, 1050:12.

Babak Esmaeili, Hao Wu, Sarthak Jain, Siddharth Narayanaswamy, Brooks Paige, and Jan-Willem Van de Meent. 2018b. Hierarchical disentangled representations. *arXiv preprint arXiv:1804.02086*.

Dunbar Ewan et al. 2019. Zerospeech 2019: Tts without t. In *Interspeech 2019*, pages 3442–3446.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM.

Zhe-Cheng Fan, Yen-Lin Lai, and Jyh-Shing Roger Jang. 2017. SVSGAN: singing voice separation via generative adversarial network. *CoRR*, abs/1710.11428.

Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba. 2018. High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 60–68. Association for Computational Linguistics.

Richard P Feynman, Robert B Leighton, and Matthew Sands. 2011. *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*, volume 1. Basic books.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*.

Matheus Gadelha, Subhransu Maji, and Rui Wang. 2017. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pages 402–411. IEEE.

Matheus Gadelha, Aartika Rai, Subhransu Maji, and Rui Wang. 2019. Inferring 3d shapes from image collections using adversarial networks. *arXiv preprint arXiv:1906.04910*.

B. Gamback and A Das. 2014. On measuring the complexity of code-mixing. In *Proc. of the 1st Workshop on Language Technologies for Indian Social Media (Social-India)*.

Björn Gambäck and Amitava Das. 2014. On measuring the complexity of code-mixing. In *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*, pages 1–7.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.

Daniel Garcia-Romero and Carol Y Espy-Wilson. 2011. Analysis of i-vector length normalization in speaker recognition systems. In *Twelfth Annual Conference of the International Speech Communication Association*.

Spandana Gella, Kalika Bali, and Monojit Choudhury. 2014. "ye word kis lang ka hai bhai?" testing the limits of word level language identification.

Arnab Ghosh, Viveka Kulharia, Vinay P Namboodiri, Philip HS Torr, and Puneet K Dokania. 2018. Multi-agent diverse generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8513–8521.

Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. 2017. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in neural information processing systems*, pages 2962–2970.

Matthew Gibson and William Byrne. 2011. Unsupervised intralingual and cross-lingual speaker adaptation for hmm-based speech synthesis using two-pass decision tree construction. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):895–904.

Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2019. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *International Journal of Computer Vision*, 127(4):398–414.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Emad M. Grais, Gerard Roma, Andrew J. R. Simpson, and Mark D. Plumbley. 2016. Discriminative enhancement for single channel audio source separation using deep neural networks. *CoRR*, abs/1609.01678.

Brian Greene. 2020. *Until the End of Time: Mind, Matter, and Our Search for Meaning in an Evolving Universe*. Knopf.

SPTK Working Group et al. 2009. Speech signal processing toolkit (sptk). *h ttp://sp-tk. sourceforge. net.*

Akshat Gupta, Xinjian Li, Sai Krishna Rallabandi, and Alan W Black. 2020a. Acoustics based intent recognition using discovered phonetic units for low resource languages. *arXiv preprint arXiv:2011.03646*.

Akshat Gupta, Sargam Menghani, Sai Krishna Rallabandi, and Alan W Black. 2021a. Unsupervised self-training for sentiment analysis of code-switched data. *arXiv preprint arXiv:2103.14797*.

Akshat Gupta, Sai Krishna Rallabandi, and Alan W Black. 2020b. Mere account mein kitna balance hai?–on building voice enabled banking services for multilingual communities. *arXiv preprint arXiv:2010.16411*.

Akshat Gupta, Sai Krishna Rallabandi, and Alan W Black. 2021b. Task-specific pre-training and cross lingual transfer for sentiment analysis in dravidian code-switched languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 73–79.

Gualberto A Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *Proceedings of INTERSPEECH*.

Bruce Hanington and Bella Martin. 2012. *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Fei He, Shan-Hui Cathy Chu, Oddur Kjartansson, Clara Rivera, Anna Katanova, Alexander Gutkin, Isin Demirsahin, Cibu Johny, Martin Jansche, Supheakmungkol Sarin, et al. 2020. Open-source multi-speaker speech corpora for building gujarati, kannada, malayalam, marathi, tamil and telugu speech synthesis systems. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6494–6503.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2010. A sequential model for discourse segmentation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 315–326. Springer.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. 2017. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1480–1490. JMLR. org.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew D Hoffman and Matthew J Johnson. 2016. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*.

Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. 2018. A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation. *arXiv preprint arXiv:1804.00522*.

Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. 2017. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Yu-An Chung, Yuxuan Wang, Yonghui Wu, and James Glass. Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Yu-An Chung, Yuxuan Wang, Yonghui Wu, and James Glass. 2018a. Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization.

Wei-Ning Hsu, Yu Zhang, Ron J. Weiss, Yu-An Chung, Yuxuan Wang, Yonghui Wu, and James Glass. 2018b. Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. 2018c. Hierarchical generative modeling for controllable speech synthesis. *arXiv preprint arXiv:1810.07217*.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. *CoRR*, abs/1704.05526.

P. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. 2012. Singing-voice separation from monaural recordings using robust principal component analysis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60.

Marijn Huijbregts, Mitchell McLaren, and David Van Leeuwen. 2011. Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection. In *2011 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4436–4439. IEEE.

David Imseng, Hervé Bourlard, Mathew Magimai Doss, and John Dines. 2011. Language dependent universal phoneme posterior estimation for mixed language speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5012–5015. IEEE.

Microsoft Research India. 2020. First workshop on speech technologies for code-switching in multilingual communities.

Keith Ito et al. 2017. The LJSpeech dataset.

Aren Jansen, Samuel Thomas, and Hynek Hermansky. 2013. Weak top-down constraints for unsupervised acoustic model training. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8091–8095. IEEE.

Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. 2018. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *European Conference on Computer Vision*, pages 829–845. Springer.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 13–24.

Bing Jiang, Yan Song, Si Wei, Jun-Hua Liu, Ian Vince McLoughlin, and Li-Rong Dai. 2014. Deep bottleneck features for spoken language identification. *PloS one*, 9(7):e100795.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. *CoRR*, abs/1804.01622.

John Jonides. 1981. Voluntary versus automatic control over the mind's eye's movement. *Attention and performance IX*, 9:187–203.

Pratik Joshi, Christain Barnes, Sebastin Santy, Simran Khanuja, Sanket Shah, Anirudh Srinivasan, Satwik Bhattamishra, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali. 2019. Unsung challenges of building and deploying language technologies for low resource language communities. *arXiv preprint arXiv:1912.03457*.

Kushal Kafle and Christopher Kanan. 2017. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3 – 20. Language in Vision.

Alexander Kain and Mike Macon. 1998. Personalizing a speech synthesizer by voice adaptation. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. 2018. Stargan-vc: Non-parallel many-to-many voice conversion with star generative adversarial networks. *arXiv preprint arXiv:1806.02169*.

Takuhiro Kaneko and Hirokazu Kameoka. 2017. Parallel-data-free voice conversion using cycle-consistent adversarial networks. *arXiv preprint arXiv:1711.11293*.

Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. 2019. Label-noise robust generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2476.

Yohan Karunanayake, Uthayasanker Thayasivam, and Surangika Ranathunga. 2019a. Sinhala and tamil speech intent identification from english phoneme based asr. In *2019 International Conference on Asian Language Processing (IALP)*, pages 234–239. IEEE.

Yohan Karunanayake, Uthayasanker Thayasivam, and Surangika Ranathunga. 2019b. Sinhala and tamil speech intent identification from english phoneme based asr. In *2019 International Conference on Asian Language Processing (IALP)*, pages 234–239. IEEE.

Yohan Karunanayake, Uthayasanker Thayasivam, and Surangika Ranathunga. 2019c. Transfer learning based free-form speech command classification for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 288–294.

Yohan Karunanayake, Uthayasanker Thayasivam, and Surangika Ranathunga. 2019d. Transfer learning based free-form speech command classification for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 288–294.

Hideki Kawahara. 2006. Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds (¡ special issue¿ introduction to the amazing world of sounds with demonstrations). *Acoustical science and technology*, 27(6):349–353.

Vahid Kazemi and Ali Elqursh. 2017. Show, ask, attend, and answer: A strong baseline for visual question answering. *CoRR*, abs/1704.03162.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*.

Jiseob Kim, Seungjae Jung, Hyundo Lee, and Byoung-Tak Zhang. 2019. Encoder-powered generative adversarial networks. *arXiv preprint arXiv:1906.00541*.

Simon King. 2016. The blizzard challenge 2016.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-sourced speech corpora for javanese, sundanese, sinhala, nepali, and bangladeshi bengali.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Arne Köhn, Florian Stegen, and Timo Baumann. 2016. Mining the spoken wikipedia for speech data and beyond. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4644–4647.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Alexander Kuhnle, Huiyuan Xie, and Ann A. Copestake. 2018. How clever is the film model, and how clever can it be? In *ECCV Workshops*.

Anoop Kunchukuttan, Ratish Puduppully, and Pushpak Bhattacharyya. 2015. Brahmi-Net: A transliteration and script conversion system for languages of the Indian subcontinent. In *HLT-NAACL*, pages 81–85.

Ratish Puduppully Kunchukuttan, Anoop and Pushpak Bhattacharyya. 2015. Brahmi-net: A transliteration and script conversion system for languages of the indian subcontinent. In *HLT-NAACL. 2015*.

Mikko Kurimo, William Byrne, John Dines, Philip N Garner, Matthew Gibson, Yong Guan, Teemu Hirsimäki, Reima Karhila, Simon King, Hui Liang, et al. 2010. Personalising speech-to-speech translation in the emime project. In *Proceedings of the ACL 2010 System Demonstrations*, pages 48–53. Association for Computational Linguistics.

Javier Latorre, Koji Iwano, and Sadaoki Furui. 2005. Polyglot synthesis using a mixture of monolingual corpora. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 1, pages I–1. IEEE.

Javier Latorre, Koji Iwano, and Sadaoki Furui. 2006. New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer. *Speech Communication*, 48(10):1227–1242.

Younggun Lee and Taesu Kim. 2019a. Robust and fine-grained prosody control of end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5911–5915. IEEE.

Younggun Lee and Taesu Kim. 2019b. Robust and fine-grained prosody control of end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5911–5915. IEEE.

Xinjian Li et al. 2020. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8249–8253. IEEE.

Ying Li, Pascale Fung, Ping Xu, and Yi Liu. 2011. Asymmetric acoustic modeling of mixed language speech. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5004–5007. IEEE.

Yuanpeng Li, Yi Yang, Jianyu Wang, and Wei Xu. 2018. Zero-shot transfer vqa dataset. *arXiv preprint arXiv:1811.00692*.

Hui Liang, Yao Qian, and Frank K Soong. 2007. An HMM-based Bilingual (Mandarin-English) TTS. *Proceedings of SSW6*.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 799–809.

Zinan Lin, Kiran Koshy Thekumparampil, Giulia Fanti, and Sewoong Oh. 2019. Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers. *arXiv preprint arXiv:1906.06034*.

Zachary C Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

Zachary C Lipton. 2018. The mythos of model interpretability. *Queue*, 16(3):31–57.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2018. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*.

Ignacio Lopez-Moreno, Javier Gonzalez-Dominguez, Oldrich Plchot, David Martinez, Joaquin Gonzalez-Rodriguez, and Pedro Moreno. 2014. Automatic language identification using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5337–5341. IEEE.

Jiasen Lu, Xiao Lin, Dhruv Batra, and Devi Parikh. 2015. Deeper lstm and normalized cnn visual question answering model.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670*.

Jing Luo, Xinyu Yang, Shulei Ji, and Juan Li. 2020. Mg-vae: Deep chinese folk songs generation with specific regional styles. In *Proceedings of the 7th Conference on Sound and Music Technology (CSMT)*, pages 93–106. Springer.

Dau-Cheng Lyu and Ren-Yuan Lyu. 2008. Language identification on code-switching utterances using multiple cues. In *Ninth Annual Conference of the International Speech Communication Association*.

Dau-Cheng Lyu, Ren-Yuan Lyu, Yuang-chin Chiang, and Chun-Nan Hsu. 2006. Speech recognition on code-switching among the chinese dialects. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE.

Alireza Makhzani and Brendan J Frey. 2017a. Pixelgan autoencoders. In *Advances in Neural Information Processing Systems*.

Alireza Makhzani and Brendan J Frey. 2017b. Pixelgan autoencoders. In *Advances in Neural Information Processing Systems*, pages 1975–1985.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Hamid Reza Marateb, Marjan Mansourian, Payman Adibi, and Dario Farina. 2014. Manipulating measurement scales in medical statistical analysis and data mining: A review of methodologies. In *Journal of research in medical sciences : the official journal of Isfahan University of Medical Sciences*.

Daniel Marcu. 1999. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 365–372. Association for Computational Linguistics.

Mikiko Mashimo, Tomoki Toda, Kiyohiro Shikano, and Nick Campbell. 2001. Evaluation of cross-language voice conversion based on gmm and straight.

Pavel Matejka, Le Zhang, Tim Ng, Harish Sri Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang. 2014. Neural network bottleneck features for language identification. *Proceedings of IEEE Odyssey*, pages 299–304.

Prem Melville and Vikas Sindhwani. 2017. *Recommender Systems*. Springer US, Boston, MA.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Seyed Hamidreza Mohammadi and Alexander Kain. 2014. Voice conversion using deep neural networks with speaker-independent pre-training. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 19–23. IEEE.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. 2016. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884.

Michele Morningstar, Dainelys Garcia, Melanie A Dirks, and Daniel M Bagner. 2019. Changes in parental prosody mediate effect of parent-training intervention on infant language production. *Journal of consulting and clinical psychology*, 87(3):313.

K Sri Rama Murty and Bayya Yegnanarayana. 2008. Epoch extraction from speech signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(8):1602–1613.

Prasanna Kumar Muthukumar and Alan W Black. 2016. Recurrent neural network postfilters for statistical parametric speech synthesis. *arXiv preprint arXiv:1601.07215*.

Pieter Muysken. 2000. *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. In *Advances In Neural Information Processing Systems*, pages 1723–1731.

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

Aaron van den Oord, Oriol Vinyals, et al. 2017a. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Aaron van den Oord, Oriol Vinyals, et al. 2017b. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

Aaron van den Oord et al. 2018. Parallel WaveNet: Fast high-fidelity speech synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926, Stockholmsmässan, Stockholm Sweden. PMLR.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Keiichiro Oura, Keiichi Tokuda, Junichi Yamagishi, Simon King, and Mirjam Wester. 2010. Unsupervised cross-lingual speaker adaptation for hmm-based speech synthesis. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010*, pages 4594–4597. IEEE.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Alok Parlikar. 2012a. TestVox: web-based framework for subjective evaluation of speech synthesis. *Opensource Software*.

Alok Parlikar. 2012b. TestVox: web-based framework for subjective evaluation of speech synthesis. *Opensource Software*.

Alok Parlikar, Sunayana Sitaram, Andrew Wilkinson, and Alan W Black. 2016. The festvox indic frontend for grapheme to phoneme conversion. In *WILDRE: Workshop on Indian Language Data-Resources and Evaluation*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

P J Phillips, Amanda C Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. 2020. Four principles of explainable artificial intelligence (draft).

Kenneth L Pike. 1945. The intonation of american english.

Wei Ping, Kainan Peng, and Jitong Chen. 2018. Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv preprint arXiv:1807.07281*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, EPFL-CONF-192584. IEEE Signal Processing Society.

Kishore Prahallad, A Vadapalli, S Kesiraju, H Murthy, S Lata, T Nagarajan, M Prasanna, H Patil, A Sao, S King, et al. 2014. The blizzard challenge 2014. In *Proceedings of Blizzard Challenge workshop*.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2018a. Waveglow: A flow-based generative network for speech synthesis. *arXiv preprint arXiv:1811.00002*.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2018b. Waveglow: A flow-based generative network for speech synthesis. *arXiv preprint arXiv:1811.00002*.

Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, and Gautham J Mysore. 2020. F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6284–6288. IEEE.

Yao Qian, Rutuja Ubale, Vikram Ramanaryanan, Patrick Lange, David Suendermann-Oeft, Keelan Evanini, and Eugene Tsuprun. 2017. Exploring asr-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 569–576. IEEE.

Yao Qian, Ji Xu, and Frank K Soong. 2011. A frame mapping based hmm approach to cross-lingual voice transformation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011*, pages 5120–5123. IEEE.

Martin Radfar, Athanasios Mouchtaris, and Siegfried Kunzmann. 2020. End-to-end neural transformer based spoken language understanding. *arXiv preprint arXiv:2008.10984*.

Z. Rafii and B. Pardo. 2013. Repeating pattern extraction technique (repet): A simple method for music/voice separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1):73–84.

Sai Krishna Rallabandi and Alan Black. 2019. Variational Attention using Articulatory priors for generating code mixed speech using monolingual corpora. In *in proceedings of Interspeech*.

Sai Krishna Rallabandi, Bhavya Karki, Carla Viegas, Eric Nyberg, and Alan W Black. 2018a. Investigating utterance level representations for detecting intent from acoustics. In *INTERSPEECH*, pages 516–520.

Sai Krishna Rallabandi, Anandaswarup Vadapalli, Sivanand Achanta, and S Gangashetty. Iiit hyderabad's submission to the blizzard challenge. In *Proc. of Blizzard Challenge 2015*.

SaiKrishna Rallabandi and Alan W Black. 2017. On building mixed lingual speech synthesis systems. *Proceedings of Interspeech 2017*, pages 52–56.

SaiKrishna Rallabandi, Sunayana Sitaram, and Alan W Black. 2018b. Automatic detection of code-switching style from acoustics. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*.

Mirco Ravanelli and Yoshua Bengio. 2018. Interpretable convolutional filters with sincnet. *arXiv preprint arXiv:1811.09725*.

Toni Reid. 2018. Everything alexa learned in 2018. *Preuzeto*, 18:2019.

Heisenberg Uncertainty Relation and Wave-Particle Duality. Spacetime is built by quantum entanglement.

Vincent Renkens, Steven Janssens, Bart Ons, Jort F Gemmeke, et al. 2014. Acquisition of ordinal words using weakly supervised nmf. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 30–35. IEEE.

Vincent Renkens et al. 2018. Capsule networks for low resource spoken language understanding. *arXiv preprint arXiv:1805.02922*.

Jason Rennie and Nathan Srebro. 2005. Loss functions for preference levels: Regression with discrete ordered labels. *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*.

Fred Richardson, Douglas Reynolds, and Najim Dehak. 2015a. Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, 22(10):1671–1675.

Fred Richardson, Douglas Reynolds, and Najim Dehak. 2015b. A unified deep neural network for speaker and language recognition. *arXiv preprint arXiv:1504.00923*.

Zatorre RJ and Baum SR. 2012. Musical melody and speech intonation: Singing a different tune. In *PLoS Biol 10(7): e1001372*.

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

Andrew Rosenberg. 2018. Speech, prosody, and machines: Nine challenges for prosody research. In *Proc. Speech Prosody*, pages 784–793.

David Rousseau and Sotirios Tsaftaris. 2019. Data augmentation techniques for deep learning. In *Tutorial Session, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

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

Shannon RV. 2005. Speech and music have different requirements for spectral resolution. In *Int Rev Neurobiol*, pages 121–34.

Alaa Saade, Alice Coucke, Alexandre Caulier, Joseph Dureau, Adrien Ball, Théodore Bluche, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, et al. 2018. Spoken language understanding on the edge. *arXiv preprint arXiv:1810.12735*.

Daniel Sáez Trigueros, Li Meng, and Margaret Hartnett. 2018. Generating photo-realistic training data to improve face recognition accuracy. *arXiv*, pages arXiv–1811.

Sakriani Sakti, Eka Kelana, Hammam Riza, Shinsuke Sakai, Konstantin Markov, and Satoshi Nakamura. 2008a. Development of indonesian large vocabulary continuous speech recognition system within a-star project. In *Proceedings of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*.

Sakriani Sakti, R Maia, S Sakai, T Shimizu, and S Nakamura. 2008b. Development of hmm-based indonesian speech synthesis. In *Proc. Oriental COCOSDA*, pages 215–219.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. 2017. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. 2019. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. pages 815–823.

Björn Schuller, Stefan Steidl, Anton Batliner, Elika Bergelson, Jarek Krajewski, Christoph Janott, Andrei Amatuni, Marisa Casillas, Amdanda Seidl, Melanie Soderstrom, et al. 2017. The interspeech 2017 computational paralinguistics challenge: Addressee, cold & snoring. In *Computational Paralinguistics Challenge (ComParE), Interspeech 2017*, pages 3442–3446.

Björn Schuller, Stefan Steidl, Anton Batliner, Elika Bergelson, Jarek Krajewski, Christoph Janott, Andrei Amatuni, Marisa Casillas, Amdanda Seidl, Melanie Soderstrom, et al. 2018a. The interspeech 2018 computational paralinguistics challenge atypical and self-assessed affect, crying and heart beats. In *Computational Paralinguistics Challenge (ComParE), Interspeech 2017*, pages 3442–3446.

Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian Müller, and Shrikanth Narayanan. 2010. The interspeech 2010 paralinguistic challenge. In *Proc. INTERSPEECH 2010, Makuhari, Japan*, pages 2794–2797.

Björn Schuller, Stefan Steidl, Anton Batliner, Simone Hantke, Florian Hönig, Juan Rafael Orozco-Arroyave, Elmar Nöth, Yue Zhang, and Felix Weninger. 2015. The interspeech 2015 computational paralinguistics challenge: nativeness, parkinson's & eating condition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Björn Schuller, Stefan Steidl, Anton Batliner, Peter Marschik, Harald Baumeister, Fengquan Dong, Simone Hantke, Florian Pokorny, Eva-Maria Rathner, Katrin Bartl-Pokorny, Christa Einspieler, Dajie Zhang, Alice Baird, Shahin Amiriparian, Kun Qian, Zhao Ren, Maximilian Schmitt, Panagiotis Tzirakis, and Stefanos Zafeiriou. 2018b. The interspeech 2018 computational paralinguistics challenge: Atypical & self-assessed affect, crying & heart beats. In

*INTERSPEECH 2018 – 18th Annual Conference of the International Speech Communication Association, September 2-6, Hyderabad, India, Proceedings*.

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

Björn W Schuller and Anton M Batliner. 1988. Emotion, affect and personality in speech and language processing.

Björn W. Schuller et al. 2019. The interspeech 2019 computational paralinguistics challenge: Styrian dialects, continuous sleepiness, baby sounds orca activity. In *Proceedings INTERSPEECH 2019, Graz, Austria*.

Benjamin Schumacher. 1995. Quantum coding. *Physical Review A*, 51(4):2738.

Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE.

Azmeh Shahid, Kate Wilkinson, Shai Marcu, and Colin M. Shapiro. 2012. *Karolinska Sleepiness Scale (KSS)*. Springer New York, New York, NY.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017. FOIL it! find one mismatch between image and language caption. *CoRR*, abs/1705.01359.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al. 2017. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving variational encoder-decoders in dialogue generation. *arXiv preprint arXiv:1802.02032*.

Weiyan Shi and Zhou Yu. 2018. Sentiment adaptive end-to-end dialog systems. *arXiv preprint arXiv:1804.10731*.

Chi-Jiun Shia, Yu-Hsien Chiu, Jia-Hsin Hsieh, and Chung-Hsien Wu. 2004. Language boundary detection and identification of mixed-language speech based on map estimation. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, pages I–381. IEEE.

Suwon Shon, Najim Dehak, Douglas Reynolds, and James Glass. Mce 2018: The 1st multi-target speaker detection and.

Sunayana Sitaram and Alan W Black. 2016a. Speech synthesis of code-mixed text. In *LREC*.

Sunayana Sitaram and Alan W Black. 2016b. Speech Synthesis of Code-Mixed Text. In *LREC*.

Sunayana Sitaram, Sai Krishna Rallabandi, and Shruti Rijhwani Alan W Black. 2015a. Experiments with cross-lingual systems for synthesis of code-mixed text. In *9th ISCA Speech Synthesis Workshop*, pages 76–81.

Sunayana Sitaram, Sai Krishna Rallabandi, and Shruti Rijhwani1 Alan W Black. Experiments with Cross-lingual Systems for Synthesis of Code-Mixed Text. In *9th ISCA Speech Synthesis Workshop*, pages 76–81.

Sunayana Sitaram, Sai Krishna Rallabandi, Shruti Rijhwani, and Alan Black. 2015b. Experiments with cross-lingual systems for synthesis of code-mixed text. In *9th ISCA Speech Synthesis Workshop*, pages 76–81.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv preprint arXiv:1803.09047*.

Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 464–472. IEEE.

Yan Song, Bing Jiang, YeBo Bao, Si Wei, and Li-Rong Dai. 2013. I-vector representation based on bottleneck features for language identification. *Electronics Letters*, 49(24):1569–1570.

Zengjie Song, Oluwasanmi Koyejo, and Jiangshe Zhang. 2020. Towards a controllable disentanglement network. *arXiv preprint arXiv:2001.08572*.

Meet Soni, Neil Shah, and Hemant Patil. 2018. Time-frequency masking-based speech enhancement using generative adversarial network.

Frank Soong and B Juang. 1984. Line spectrum pair (lsp) and speech data compression. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, volume 9, pages 37–40. IEEE.

Kenneth N Stevens. 2000. *Acoustic phonetics*, volume 30. MIT press.

Emma Strubell et al. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *ACL*.

Yuanhang Su, Kai Fan, Nguyen Bach, C.-C. Jay Kuo, and Fei Huang. 2018. Unsupervised multimodal neural machine translation. *CoRR*, abs/1811.11365.

David Sundermann, Harald Hoge, Antonio Bonafonte, Hermann Ney, Alan Black, and Shri Narayanan. 2006. Text-independent voice conversion based on unit selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing,*, volume 1, pages I–I. IEEE.

Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. Voice synthesis for in-the-wild speakers via a phonological loop.

Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. 2017a. Voice synthesis for in-the-wild speakers via a phonological loop. *arXiv preprint arXiv:1707.06588*, pages 1–11.

Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. 2017b. Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*.

Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. 2017c. Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*.

Shinnosuke Takamichi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura. 2014. A postfilter to modify the modulation spectrum in hmm-based speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 290–294. IEEE.

Paul Taylor, Alan W Black, and Richard Caley. 1998. The architecture of the festival speech synthesis system. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*.

J. Tee and D. P. Taylor. 2018. Is Information in the Brain Represented in Continuous or Discrete Form? *arXiv e-prints*.

Mariët Theune, Sander Faas, Anton Nijholt, and Dirk Heylen. 2002. The virtual storyteller. *ACM SigGroup Bulletin*, 23(2):20–21.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2017. Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 301–308. IEEE.

Laura Mayfield Tomokiyo, Alan W Black, and Kevin A Lenzo. 2005. Foreign accents in synthetic speech: development and evaluation. In *Interspeech*, pages 1469–1472.

Christof Traber, Karl Huber, Karim Nedir, Beat Pfister, Eric Keller, and Brigitte Zellner. 1999. From multilingual to polyglot speech synthesis. In *Sixth European Conference on Speech Communication and Technology*.

George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. 2016. Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5200–5204. IEEE.

Satoshi Tsujioka, Sakriani Sakti, Koichiro Yoshino, Graham Neubig, and Satoshi Nakamura. 2016. Unsupervised joint estimation of grapheme-to-phoneme conversion systems and acoustic model adaptation for non-native speech recognition. *Interspeech 2016*, pages 3091–3095.

Cassia Valentini Botinhao, Xin Wang, Shinji Takaki, and Junichi Yamagishi. 2016. Speech enhancement for a noise-robust text-to-speech synthesis system using deep recurrent neural networks. In *Proceedings of Interspeech 2016*, pages 352–356.

Aäron Van Den Oord et al. 2016. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*.

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

Werner Verhelst and Marc Roelands. 1993. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 554–557. IEEE.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A first speech recognition system for mandarin-english code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892. IEEE.

Vincent Wan, Chun-an Chan, Tom Kenter, Jakub Vit, and Rob Clark. 2019. Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. *arXiv preprint arXiv:1905.07195*.

Dong Wang and Xuewei Zhang. 2015. Thchs-30: A free chinese speech corpus. *arXiv preprint arXiv:1512.01882*.

Ye Wang, Toshiaki Koike-Akino, and Deniz Erdogmus. 2018a. Invariant representations from adversarially censored autoencoders. *arXiv preprint arXiv:1805.08097*.

Yun Wang and Florian Metze. 2017. A transfer learning based feature extractor for polyphonic sound event detection using connectionist temporal classification. *Proceedings of Interspeech, ISCA*, pages 3097–3101.

Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Robert Clark, and Rif A. Saurous. 2017a. Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017b. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017c. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Yuxuan Wang, RJ Skerry-Ryan, Ying Xiao, Daisy Stanton, Joel Shor, Eric Battenberg, Rob Clark, and Rif A Saurous. 2017d. Uncovering latent style factors for expressive speech synthesis. *arXiv preprint arXiv:1711.00520*.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous. 2018b. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous. 2018c. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*.

Yuxuan Wang et al. 2018d. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*.

Nigel G Ward. 2019. *Prosodic patterns in English conversation*. Cambridge University Press.

Oliver Watts. 2012. *Unsupervised Learning for Text-to-Speech Synthesis*. Ph.D. thesis, University of Edinburgh.

Andrew Wilkinson, Alok Parlikar, Sunayana Sitaram, Tim White, Alan W Black, and Suresh Bazaj. Open-source consumer-grade indic text to speech. In *9th ISCA Speech Synthesis Workshop*, pages 190–195.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Sarah G Wood, Jerad H Moxley, Elizabeth L Tighe, and Richard K Wagner. 2018. Does use of text-to-speech and related read-aloud tools improve reading comprehension for students with reading disabilities? a meta-analysis. *Journal of learning disabilities*, 51(1):73–84.

Mike Wu, Jonathan Nafziger, Anthony Scodary, and Andrew Maas. 2020. Harpervalleybank: A domain-specific spoken dialog corpus. *arXiv preprint arXiv:2010.13929*.

Peter Wu, Sai Krishna Rallabandi, Alan W Black, and Eric Nyberg. 2019. Ordinal triplet loss: Investigating sleepiness detection from speech. In *INTERSPEECH*, pages 2403–2407.

Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4622–4630.

Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov. 2015a. Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King. 2015b. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4460–4464. IEEE.

Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, and Massimiliano Todisco. 2017. Asvspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604.

Sitao Xiang, Yuming Gu, Pengda Xiang, Mingming He, Koki Nagano, Haiwei Chen, and Hao Li. 2020. One-shot identity-preserving portrait reenactment. *arXiv preprint arXiv:2004.12452*.

Junichi Yamagishi, Takashi Nose, Heiga Zen, Zhen-Hua Ling, Tomoki Toda, Keiichi Tokuda, Simon King, and Steve Renals. 2009. Robust speaker-adaptive hmm-based text-to-speech synthesis. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1208–1230.

Junlin Yang, Nicha C Dvornek, Fan Zhang, Juntang Zhuang, Julius Chapiro, MingDe Lin, and James S Duncan. 2019a. Domain-agnostic learning with anatomy-consistent embedding for cross-modality liver segmentation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0.

Shuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. 2019b. Tet-gan: Text effects transfer via stylization and destylization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1238–1245.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29.

David Yarowsky. 1997. Homograph disambiguation in text-to-speech synthesis. In *Progress in speech synthesis*, pages 157–172. Springer.

Ching Feng Yeh, Chao Yu Huang, Liang Che Sun, and Lin Shan Lee. 2010. An integrated framework for transcribing mandarin-english code-mixed lectures with improved acoustic and language modeling. In *Chinese Spoken Language Processing (ISCSLP), 2010 7th International Symposium on*, pages 214–219. IEEE.

Ching-Feng Yeh and Lin-Shan Lee. 2015. An improved framework for recognizing highly imbalanced bilingual code-switched lectures with cross-language acoustic modeling and frame-level language identification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(7):1144–1159.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 2001. Mixed excitation for hmm-based speech synthesis. In *Seventh European Conference on Speech Communication and Technology*.

Xiaoyong Yuan, Pan He, Qile Zhu, Rajendra Rana Bhat, and Xiaolin Li. 2017. Adversarial examples: Attacks and defenses for deep learning. *CoRR*, abs/1712.07107.

Biqiao Zhang, Yuqing Kong, Georg Essl, and Emily Mower Provost. f-similarity preservation loss for soft labels: A demonstration on cross-corpus speech emotion recognition.

Jason Y Zhang, Alan W Black, and Richard Sproat. 2003. Identifying speakers in children's stories for speech synthesis. In *Eighth European Conference on Speech Communication and Technology*.

Mingyang Zhang, Xin Wang, Fuming Fang, Haizhou Li, and Junichi Yamagishi. 2019a. Joint training framework for text-to-speech and voice conversion using multi-source tacotron and wavenet. In *arXiV*.

Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling. 2019b. Learning latent representations for style control and transfer in end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6945–6949. IEEE.

Yi Zhang and Jianhua Tao. 2008. Prosody modification on mixed-language speech synthesis. In *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*, pages 1–4. IEEE.

Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, RJ Skerry-Ryan, Ye Jia, Andrew Rosenberg, and Bhuvana Ramabhadran. 2019c. Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning. *arXiv preprint arXiv:1907.04448*.

Jingjing Zhao, Hua Shu, Linjun Zhang, Xiaoyi Wang, Qiyong Gong, and Ping Li. 2008. Cortical competition during language discrimination. *NeuroImage*, 43(3):624–633.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2017a. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017b. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.

Wei Zhao, Benyou Wang, Jianbo Ye, Min Yang, Zhou Zhao, Ruotian Luo, and Yu Qiao. 2018a. A multi-task learning approach for image captioning. In *IJCAI*, pages 1205–1211.

Yue Zhao, Hong Zhu, Qintao Shen, Ruigang Liang, Kai Chen, and Shengzhi Zhang. 2018b. Practical adversarial attack against object detector. *CoRR*, abs/1812.10217.

Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 43–51.

Chunting Zhou and Graham Neubig. 2017a. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. *arXiv preprint arXiv:1704.01691*.

Chunting Zhou and Graham Neubig. 2017b. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. *arXiv preprint arXiv:1704.01691*.

Chunting Zhou and Graham Neubig. 2017c. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. *CoRR*, abs/1704.01691.

Su Zhu, Zijian Zhao, Tiejun Zhao, Chengqing Zong, and Kai Yu. 2019. Catslu: The 1st chinese audio-textual spoken language understanding challenge. In *2019 International Conference on Multimodal Interaction*, pages 521–525.