Neural Turn-Taking Models for Spoken Dialogue Systems

Matthew Roddy

Department of Electronic and Electrical Engineering

Trinity College Dublin

2021

A dissertation submitted to the University of Dublin

for the degree of Doctor of Philosophy

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work. I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Matthew Roddy

Abstract

In order to simulate naturalistic turn-taking behaviours, such as fast-turn switches, intentional overlap, backchanneling, and barge-in, spoken dialogue systems (SDSs) will need to have computational models of turn-taking that are both *predictive* and *incremental*. They will need to be predictive in the sense that they predict future user turn-taking behaviours rather than respond to behaviours that have already occurred, as is typically done in traditional endpointing-based systems. In the *projection theory* of Sacks et al. (1974) they proposed that humans are capable of anticipating turn endings before they occur. We argue that SDSs which aim to converse in a human-like manner should be capable of anticipating user behaviours as well. To make decisions based on these predictions, the system must process information incrementally, while the user is still speaking.

In this thesis we develop recurrent neural network (RNN) based models of turn-taking that are both predictive and incremental. Continuous turn-taking (CTT) models as proposed by Skantze (2017b) were taken as a starting point. We investigated these models and proposed a number of improvements and extensions. First, we performed an analysis of input features for CTT models, gained insights into the utility of different varieties of features, and proposed optimal sets. We then proposed architectural improvements to the original CTT model in the form of a multiscale RNN architecture that allows features to be processed at an independent rate. We then designed a control process based on partially observable Markov decision processes (POMDPs) that is able to employ the predictive nature of our RNN models to make responsive turn-taking decisions.

Our investigations led to the development of a different variety of model that can be used for generating naturalistic response timings using features from both the user's turn and the system turn. Our response timing networks (RTNets) are motivated by the observation that response timings carry communicative importance, and that listeners associate different timings with different types of responses. RTNets are still both predictive and incremental, but they differ from CTT models in many other aspects, such as their objective functions and architectures. We propose that these models address an overlooked aspect of SDS response generation that can increase the realism of SDS interactions.

Acknowledgements

Being able to work on this PhD has been a privilege. I'm truly grateful to my supervisor Naomi Harte, for giving me the opportunity to do it. Over the last four years Naomi has been an endless source of wisdom and encouragement. You really couldn't ask for a better supervisor. I'd also like to thank a bunch of my fellow Sigmedians and Adapters who have helped me out over the years, in particular (but in no particular order): Joao Cabral, Ilaria Torre, Francois Pitié, Anil Kokaram, Rozenn Dahyot, Christian Saam, Ali Karaali, George Sterpu, Marco Forte, Sébastian Le Maguer, Ailbhe Cullen, Daniel J. Ringis, and Colm O'Reilly. I'd like to express my gratitude to my examiners David Schlangen and Carl Vogel for their thoughtful insights. I'd also like to thank Gabriel Skantze for having me over to KTH for an internship. Finally, I'd like to thank my mom, dad, and my partner Kate for all their support and encouragement.

Contents

Abbreviations

ASR	Automatic Speech Recognition
BCE	Binary Cross-Entropy
CA	Conversation Analysis
CPS	Conversation-turns Per Session
CTT	Continuous Turn-Taking
DA	Dialogue Act
DM	Dialogue Manager
DST	Dialogue State Tracker
ECA	Embodied Conversational Agent
EOT	End-Of-Turn
FCI	False Cut-In
FSM	Finite State Machine
FSTTM	Finite State Turn-Taking Machine
IPU	Inter-Pausal Unit
IR	Information Retrieval
IVA	Intelligent Virtual Assistant
KL	kullback-leibler
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MFCC	Mel-Frequency Cepstral Coefficient
MMD	Mahnob Mimicry Database
MTC	HCRC MapTask Corpus
MTP	Mid-turn pause
NLG	Natural Language Generation
OOD	Out-Of-Distribution
OOV	Out-Of-Vocabulary

POMDP Partially Observable Markov Decision Process

POS Part-Of-Speech

RL Reinforcement Learning

RNN Recurrent Neural Network

RTNet Response Timing Network

SDS Spoken Dialogue System

SLU Spoken Language Understanding

SWBD Switchboard Corpus

TTS Text To Speech

VAD Voice Activity Detector

VAE Variational Autoencoder

Chapter 1

Introduction

1.1 Motivation

1.1.1 Modern Spoken Dialogue Systems

The predominant way for humans to interact with computers is through some form of graphical user interface (GUI), which can be controlled using hardware such as keyboards, mice, and touch screens. We are able to use these interfaces to input information to the computer, which then typically outputs information in visual form to a screen. The way we use these devices to interact with computers contrasts with the way humans most commonly use spoken dialogue to interact with one another. When compared with other forms of human language (e.g. written prose, oration, written dialogue), spoken dialogue is unique in that it is the first form of language that we learn as children. Dialogue, as a form of language, exists in spoken languages where there is no written form. It can be used to communicate in areas with high rates of illiteracy. It is also the form of language that most of us use most commonly in our day-to-day lives.

The use of spoken dialogue to interact with computers entails *spoken dialogue systems* (SDSs), which are computer systems where users are able to input information to the computer using speech, while the computer is able to output information using synthesized speech. SDSs have long been featured in science fiction films, and while some applications of SDSs have established use-cases going back several decades (most notably applications that involve telephony e.g. travel information (Raux et al., 2005) and banking (Melin et al., 2001)) it is only in the past decade that SDSs have achieved widespread adoption as a method for interfacing with *intelligent virtual assistants* (IVAs) such as Apple Siri, Microsoft Cortana, Amazon Alexa, and

Google Assistant.

IVAs are task-oriented dialogue systems that are designed to perform multiple useful functions such as: answer queries (e.g. retrieve weather forecasts or bus times), play music, set reminders, shopping, and news updates. Most of these systems are designed as a collection of skills, within which one or more speaking turns by the user and the system can be used to complete a task. For example, Alexa offers pizza-ordering skills where the user interacts with the IVA over multiple turns (Wong, 2017). Alexa is able to ask the user for information such as desired pizza type and size, while it is also able to supply the user with information on the menu details. These multi-turn interactions require that the SDS maintains a record of the *dialogue state* (what has already been said) while eliciting the necessary information from the user required to complete the task.

As the adoption of SDSs has surged over the past decade, thanks largely to the increasing popularity of IVAs (BusinessWire, 2018), two noteworthy commercial SDSs have recently emerged that embody two different (but not opposing) technological aspirations for SDSs: Google's Duplex, and Microsoft's XiaoIce. Duplex is a component of Google Assistant that can be used to make restaurant, cinema, and hair salon bookings where the SDS places a telephone call to the business to make the booking on behalf of the user. It represents a commercial effort to make a task-based interaction with a machine as "natural as possible, allowing people to speak normally, like they would to another person, without having to adapt to a machine" (Leviathan and Matias, 2018). This necessitates reproducing human-like behaviours such as backchannels, fillers, interruptions, and low-latency responses. Due to the complexities involved in human conversational dialogue, Duplex is specifically designed to only operate in a highly constrained number of task-based domains.

Microsoft's XiaoIce, on the other hand, represents a commercial effort to design a social chatbot that optimizes for long-term user engagement as measured in expected *conversation-turns per session* (CPS) (Zhou et al., 2020). The objective is to keep the user engaged with the dialogue system for as long as possible, rather than to efficiently complete a task. It keeps the user engaged, in part, by establishing a social and emotional connection with the user which is maintained over multiple conversations in a similar way that a human-human social relationship would be maintained. For example, it can carry out discussions about pop culture, music, and current affairs, while maintaining its persona and remembering information about the user's personal views and opinions that have been stated previously. Notably, it is able to imitate human

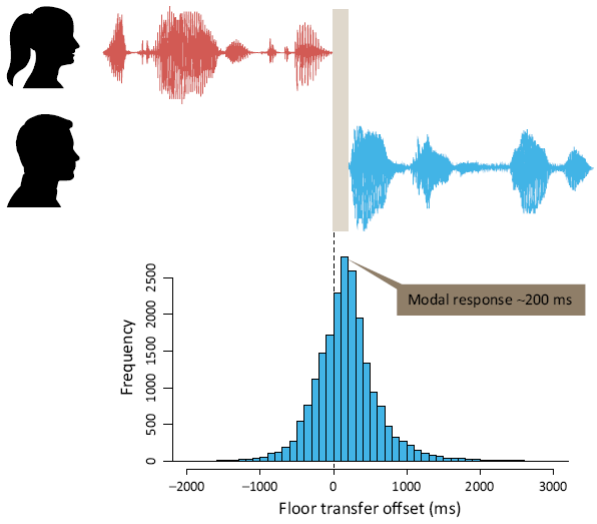


Figure 1.1: Histogram of floor transfer offset timings. From (Levinson, 2016)

empathetic behaviours in that it is designed to comfort users when they are feeling unhappy. It can do this while also having the ability to perform standard operations associated with IVAs (e.g. weather forecasts, playing music, setting alarms, etc.). XiaoIce has been reported to be very popular in China and Japan, with over 660 million users, making it the most used social chatbot in the world (Zhou et al., 2020).

While it is uncertain how long either Duplex or XiaoIce will remain active as commercial products, they have been introduced in this thesis as illustrative examples of modern commercial applications of SDSs that diverge from the traditional format of IVAs. They both attempt to simulate aspects of human conversational interactions that don't usually exist in IVAs. In XiaoIce it is the higher-level semantic *content* of the dialogue that aspires to be similar to human social interactions. While with Duplex, the focus is on making the lower-level *interaction* with the content more natural and human-like.

1.1.2 Turn-Taking in Conversation

Turn-taking is a fundamental consideration for simulating naturalistic interactions through spoken dialogue. It is the process by which dialogue participants organize the starts and ends of their speaking turns. In their influential paper, Sacks et al. (1974) proposed that participants in a conversation attempt to minimize gaps and overlaps. When considered objectively, humans are very good at this organization process. Although overlap occurs frequently during conversations,

less than 5% of speech occurs in overlap (Levinson, 1983). Also, the modal turn switch offset time (the time from the end of the first speaker's turn to the start of the second speaker's turn) has been observed to be around 200 ms, which is close to the limit of human reaction time for any stimuli (Levinson and Torreira, 2015). Figure 1.1 shows a histogram of floor transfer offset timings in a dataset of casual telephone conversations (Switchboard). These fast turn switches have been observed across cultures, independent of the language (Stivers et al., 2009).

Given the slow nature of the human language production mechanism (Levinson and Torreira, 2015), in order to achieve these fast turn-switch times it is necessary for listeners to have planned what they are going to say, and to predict the end of a speaker's turn before the speaker has finished speaking. In order to predict the end of a speaker's turn, listeners are able to exploit turn-taking cues that exist in multiple levels of the dialogue including: acoustics, prosody, syntax, lexicon, semantics, discourse structure, and pragmatics. In face-to-face interactions, predictions may also be made using visual turn-taking cues such as eye-gaze behaviours.

1.1.3 Turn-Taking Modelling

While humans are very good at this prediction process, reproducing the same turn-taking behaviours in SDSs is far from straightforward (Ward et al., 2005; Raux et al., 2005). The acoustic and prosodic cues that human listeners use are often subtle, while the linguistic cues are often rooted in syntactic and semantic elements of language that are not always simple to detect. The process of determining when the user has finished speaking is known as *endpointing* or *end-of-turn* detection. Endpointing models function by detecting silence from the user and then using features from the dialogue to make inferences as to whether the silence is an end-of-turn (EOT) silence or a mid-turn-pause (MTP). Endpointing mistakes can lead to either interruptions or undesirable long silences.

IVAs, such as the ones discussed above in Section 1.1.1, rely on endpointing to determine when the system can take a turn. Current cloud-based IVAs typically respond to questions with average offsets in the approximate range of 1.6 to 2.3 seconds[1] (Dgit, 2020). These offsets are much slower than those typically observed in natural conversation. While recent advances in on-device ASR (He et al., 2019) can improve these offsets, endpointing cannot simulate the natural distribution of offsets observed in Fig. 1.1 since it relies on reacting to the detection of silence. This precludes the generation of fast turn-switches that occur in partial overlap with the

[1]Exact offsets are difficult to measure since they also depend on external factors such as internet connection speed. The reported range is only a rough indicator.

user's turn.

In order to simulate naturalistic turn-taking behaviours with fast turn switches, it is necessary to predict whether the user is *about* to finish speaking, as humans do, rather than waiting until they are already finished. This requires making predictions about the future turn-taking behaviour of the user in an incremental manner. A continuous turn-taking (CTT) model was proposed by Skantze (2017b) that makes predictions about the future speech activity of a dialogue participant at regular frame-based intervals using a recurrent neural network (RNN). In theory, these models enable the types of responsive turn-taking behaviours that are absent from IVAs, since CTT models make predictions about the future, whereas endpointing models react to events that have already occurred.

1.2 Thesis Statement

This thesis argues that, in order to achieve naturalistic turn-taking behaviours, the turn-taking models will need to have two properties: they should be *predictive* and *incremental*. The models will need to be *predictive* in the sense that the behaviour of the user should be anticipated by the system, as opposed to reacted to. Endpointing as a reactive process is unable to model the fast offsets that occur in partial overlap with the end of a user's utterance. Making explicit decisions before the user has finished speaking also requires that the system be (at least partially) *incremental*. Without being incremental, the system will not be able to model overlaps or other naturalistic behaviours such as barge-in.

Therefore, in the work done for this thesis, efforts to achieve these natural turn-taking behaviours in SDSs are made through investigations in the domain of predictive incremental models. CTT models were taken as a starting point. We investigated these models and proposed improvements. Our investigations led to the development of a different variety of model that can be used for generating naturalistic response timings using features from both the user's turn and the system turn. The response timing prediction models are still both predictive and incremental, but they differ from CTT models in many other aspects, such as their objective functions and architectures.

1.3 Contributions

In the rest of this section we list and summarize the main contributions of the thesis. The main contributions also correspond to the main chapters of the thesis:

1. **An Analysis of Features in CTT Models**

   CTT models are by their nature different from non-continuous endpointing models since they capture more general information about turn-taking behaviours in the training data. CTT models can be applied to a variety of different prediction tasks rather than simply making endpointing decisions at the end of a user's turn. This presents questions about CTT models concerning which features are most useful for the variety of predictions that CTT models can make.

   In Chapter 3 we present an analysis of acoustic and linguistic features when used in CTT models. We use evaluation metrics that capture the performance of models on a variety of endpointing and non-standard turn-taking tasks. The results provide insight into how feature choices could affect performance in SDS turn-taking behaviours. Using our feature sets and a different loss function, we were able to improve upon previously reported benchmarks by Skantze (2017b).

2. **A multiscale CTT architecture to fuse temporally independent modalities**

   In the original formulation of CTT models proposed by Skantze (2017b), POS features and acoustic features are concatenated in a vector that is input to the RNN at each regular time step. A limitation of this approach is that all of the features across different modalities must be input to the RNN at a fixed temporal rate. Since the relevant information from acoustic and lexical features occurs at different temporal granularities, using this fixed temporal granularity makes it difficult for the RNN to model long-term dependencies that exist between lexical features.

   Building on the work in the previous contribution, in Chapter 4 we present a multiscale CTT architecture that allows features from different modalities to be modelled independently in subnetwork RNNs. We show that modelling features at independent temporal granularities that are more suited to the modality improves the overall performance (relative to the benchmarks achieved in the previous contribution). We also investigate the use of our multiscale architecture with visual features such as eye-gaze vectors.

3. **An application of decision theory to CTT**

   One of the overall objectives of this thesis is to be able to simulate the fast turn-switch offsets that exist in human conversations using turn-taking models. In theory the continuous nature of CTT models should allow for decisions to be made before a silence has been detected by the user, enabling fast turn-switches that occur in overlap with the end of the user's turn. However, the previous proposed models all still rely on the detection of silence by the user to trigger turn-switch decisions. By relying on silence detection we are still constrained to making reactive endpointing decisions, rather than using predictions about when the user will stop.

   In Chapter 5 we propose a control process that enables turn switch decisions to be made prior to the end of the user's utterance, anticipating the end of the turn. The control process is based on partially observable Markov decision processes (POMDPs) and uses probabilistic output predictions from a modified CTT model. We show that the POMDP control process is able to outperform several other baselines. It also enables flexible tuning of the trade-off between latency and false-cut-ins (FCIs).

4. **A way of generating natural response timing offsets for SDSs**

   In natural spoken conversation, response offset timings depend on the context of both the first speaker's turn and the second speaker's turn. In cases where the first speaker asks a question and the second speaker gives a dispreferred response, it may be more natural for the second speaker to delay their response. In cases where the first speaker is uttering a backchannel, it may be natural for the second speaker to start their turn in overlap with the backchannel. While our previous contribution enables fast turn-switch offsets with small amounts of overlap, it does not take into account the context of the system response.

   In Chapter 6 we propose neural models that take into account the context of both the user's turn as well as the system's response to generate the distribution of turn-switch offsets. These models use an encoding of the system turn as well as acoustic and linguistic features extracted from the user's speech signal to make the binary decision to start speaking or not. These response timing networks (RTNets) represent a class of continuous model that is distinct from CTT models in its objective function and its architecture. We also introduce a variant of the RTNet model, RTNet-VAE, that uses a variational autoencoder (VAE) to train an interpretable representation of the response encoding, which allows

easier integration with SDS pipelines. We present the results of human listening tests which showed that listeners found that some response timings were more natural than others. We show that in instances where listeners are sensitive to response timings it is likely that our system will generate response timings that are more realistic than a system that generates the modal offset.

1.4 Thesis Structure

The rest of this thesis has four chapters corresponding to the four contributions outlined above, as well as a background chapter and a conclusion chapter. It is structured as follows: In Chapter 2 we discuss relevant background material about the social sciences, turn-taking, turn-taking models, spoken dialogue systems, and machine learning. In Chapter 3 we present our analysis of features as applied to CTT modelling. In Chapter 4 we present our multiscale approach to CTT. In Chapter 5 we present ways of applying decision theory to making HOLD/SHIFT decisions using CTT models. In Chapter 6 we present neural models to generate the response offset distributions for SDSs.

Chapter 2

Background and State of the Art

2.1 Human Conversations

In Section 1.1.2 some of basics of turn-taking were introduced. In this section we discuss the details of turn-taking and conversation analysis (CA) in more depth.

2.1.1 Turn-Taking Dynamics

As briefly mentioned in Section 1.1.2, the main theoretical underpinnings to CA theory on turn-taking were introduced by Sacks et al. (1974). The authors make the distinction between two different types of silences that occur during dialogue: *gaps* which are silences that occur at points where there is a speaker change, and *pauses* which are silences that occur within the speech of one speaker. In their paper the authors identify four ways that speaker changes can occur: in lapses (long gaps), in gaps, in overlaps, or in no-gaps-no-overlaps. From analysing conversational data, they found that the most common way for turn switches to occur was in no-gap-no-overlap or with a slight gap or slight overlap. This led them to propose their *projection theory*, that participants in a conversation tend to minimize gaps and overlaps in conversations by anticipating the end of the the current speaker's turn. The term 'projections' in this sense refers to predictions about future conversational events, that are formulated on the basis of contextual information in the dialogue.[1]

[1] This is distinguished from the concept of 'projection' as used in phrase structure analysis.

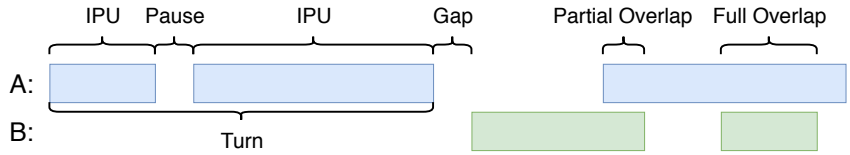


Figure 2.1: Diagram of spoken dialogue segmentation scheme. Rectangular boxes represent speech by a speaker.

There is ambiguity in what exactly a turn consists of (Ten Have, 1990). In the turn-taking paper of Sacks et al. (1974) the authors do not formally define what constitutes a turn but only state that turns consist of linguistic structures that "allow a projection of the unit-type under way, and what, roughly, it will take for an instance of that unit-type to be completed." (Sacks et al., 1974, p.702) As such, they propose that turns are mainly linguistic constructs that allow listeners to project when the unit is finished. However, they also note that other factors such as prosody may play a part in the projection process. This definition is difficult to apply methodically to corpus-based CA and therefore schemes have been proposed that can be used to objectively segment conversations.

This thesis uses the segmentation scheme described by Beňuš et al. (2011), which is based on segmenting dialogue into *interpausal units* (IPUs) and *turns*. IPUs are defined as a maximal sequence of speech by one speaker which is surrounded by silence of no greater than some threshold length (e.g. 200 ms). A turn is then defined as "a maximal sequence of IPUs from one speaker, such that between any two adjacent IPUs there is no speech from the interlocutor." This IPU-based approach to segmentation (shown in Fig. 2.1) is useful for formal analyses of corpora. However, Wlodarczak and Wagner (2013) show that arbitrarily selecting threshold IPU parameters has a considerable effect on the reported frequencies of silences and overlaps. Analyses are sensitive to this chosen threshold and a wide variety of different values have been used (for example: 50 ms (Gravano and Hirschberg, 2011), 100 ms (Koiso et al., 1998), 200 ms (Meena et al., 2014), and 500 ms (Razavi et al., 2019)).

Sacks et al. (1974) identify possible turn completion points as *transition relevance places* (TRPs). These are points where there is an opportunity for the conversational floor to be taken by another participant. However, other participants are not obliged to take the floor, and the current speaker could potentially self-select and continue to hold the floor. They note that "if a developing silence occurs at a transition-place, and is thus a (potential) gap, it may be ended by talk of the same party who was talking before it; so the 'gap' is transformed into a 'pause' (being

now intra-turn)." (Sacks et al., 1974, p.715) The authors do not fully define what constitutes a TRP but propose that they occur at points of possible syntactic completion, and that prosody plays an important role. They propose a set of rules that govern turn allocation at each TRP which were summarized by Gravano and Hirschberg (2011, p.603):

1. if the current speaker (CS) selects a conversational partner as the next speaker, then such partner must speak next;
2. if CS does not select the next speaker, then anyone may take the next turn;
3. if no one else takes the next turn, then CS may take the next turn.

The selection process referred to in rule 1 could be a question or statement directed to a conversational partner, or a cue directed towards a specific partner (e.g. gaze or a gesture). In this "one-at-a-time" model of conversation, interruptions and false starts are considered by Sacks et al. as repair devices to fix an error in the communication process. They observe that interruptions are not necessarily rude; they break down the normal flow of turn-taking but they can also serve a useful purpose in exchanges where information needs to be revised.

Much of the CA theory proposed by Sacks et al. has since been supported in large-scale corpus studies. Heldner and Edlund (2010) observed that gaps and overlaps are common in human spoken conversations but humans are good at keeping them to a minimum. In a corpus study of casual telephone conversations in English, Levinson and Torreira (2015) found that 30% of offsets occurred with some overlap but only 3.8% of the overall corpus consisted of overlap, while 77% consisted of speech by one speaker, and 19.2% consisted of silence. It has been observed that the overlaps that do occur tend to be at the end of a speaker's turn where there is a TRP (Levinson, 1983).

The distribution of the timings of turn-switch offsets has since been analysed more accurately and been shown to have a modal value of approximately 200 ms (Heldner and Edlund, 2010) (shown in Fig. 1.1), a value which has been observed to be consistent across cultures (Stivers et al., 2009). This 200 ms value can seem surprisingly fast when considered within the context of the delays caused by the human language production mechanism. Indefrey and Levelt (2004) investigated the cognitive and physiological delays from seeing a picture to naming the object in it. They estimated that the total average delay involved in conceptual preparation, lemma retrieval, and form encoding amounted to 600 ms. The analysis of the constraints of the human language production mechanism presents evidence supporting the projection theory of Sacks et

al., as well as presents evidence that planning plays a large role in turn-switch offsets. In order to achieve these fast offsets, it is likely that listeners need to have responses formulated before the end of a speaker's turn. Bögels et al. (2015) used experiments with EEGs to investigate when listener's responses are formulated. They showed that production processes for an upcoming turn start around 500 ms after the information necessary to formulate the response is presented to the upcoming speaker.

While the overall mode of turn switch offsets has been found to be consistent across cultures (Stivers et al., 2009), the distribution of these offsets is reported to be influenced by a number of factors. Cognitive load is often associated with larger response times. Bull and Aylett (1998) showed that increased task complexity resulted in longer response times. Roberts et al. (2015) found that the timing of responses was influenced by the syntactic and lexical complexity of both the first speaker's turn and the second speaker's turn. Strömbergsson et al. (2013) found that the response offsets to questions vary based on the type of question (e.g. yes/no questions, "wh" questions, open questions, etc.). This is to be expected due to the varying amount of cognitive load associated with each type. Heeman and Lunsford (2017) showed that the upcoming dialogue act, the previous dialogue act, and the nature of the dialogue all have an impact on turn-taking offsets.

There has been substantial research in CA on investigating the cues that are used by listeners to anticipate the end of a speaker's turn. Yngve (1970) found that humans rely on cues from syntax, acoustics, and prosody. Duncan (1972) reported that dialogue participants continuously monitor cues that can be found in syntax, semantics, prosody, and gesture. These cues can be categorized as either *turn-yielding* or *turn-holding* (Duncan, 1972; Koiso et al., 1998; Gravano and Hirschberg, 2011) and exist across multiple modalities (i.e. acoustics, language, or vision). Cues can also be entrained or dynamically accommodated by conversational participants (Garrod and Pickering, 2004). Hjalmarsson (2011) presents evidence that these cues are additive. In other words, that the presence of more turn-yielding cues there are, the more likely there will be a turn switch. She found that this also applied to synthesized speech.

A number of papers have pointed out that it is reasonable to expect different languages, cultures, dialects, task-settings, and genres to have different turn-taking cues (Clancy et al., 1996; Furo, 2013; Stivers et al., 2009; Brusco et al., 2017; Heeman and Lunsford, 2017; Ward et al., 2018; Lala et al., 2018). For this thesis, all of the datasets used are in English dialects (these datasets are reviewed in Section 2.4). It cannot be assumed that the same modelling

techniques and features would yield similar results in other languages. In the review of turn-taking cues that is presented in the next subsection, we primarily focus on English turn-taking cues. Publications that use conversational data from languages other than English are indicated as such.

2.1.2 Turn-Taking Cues

An understanding of turn-taking cues that are used in human interactions is central to the design of responsive turn-taking models. In this sub-section we present a survey of some of the main documented turn-taking cues from prosody, acoustics, timing, lexicon, syntax, semantics, discourse structure, breathing, and vision.

Syntax Sacks et al. (1974) considered the units that make up turns to be primarily syntactic in nature. They observed that the units that make up turns do not necessarily comprise of full sentences, but can also consist of syntactic units that are smaller than clauses, such as phrases and lexical items. *Syntactic completion points* are points in an utterance which can be considered syntactically complete "so far." The following is an example of dialogue given by Ford and Thompson (1996, p.144) where the the slashes represent points of syntactic completion:

> D: I mean it's it's not like wi:ne/ it doesn't taste like wine/ but it's
>
> W: fermented./
>
> D: White/ and milky/ but it's fermented/

The authors found that syntactic completion was a cue for turn-taking, particularly when used in conjunction with prosodic cues (discussed below) such as rising or falling pitch. They also observed an interplay between prosody and syntax, whereby a speaker who wishes to retain the conversational floor at a natural point of syntactic completion, may use prosody to signal that they wish to continue their turn:

> V: She didn't know/ what was going on/ about why they didn't change the knee/.

In this example Ford and Thompson (1996, p.148) there is a point of syntactic completion after "going on/" but not prosodic completion. The speaker's pitch frequency stays approximately flat. The authors argue that this signals to the listener that there is more to come.

The syntax of a formal language can be represented using phrase-structure grammars, which are a type of context-free grammar (Chomsky, 1957). A phrase structure grammar defines a set

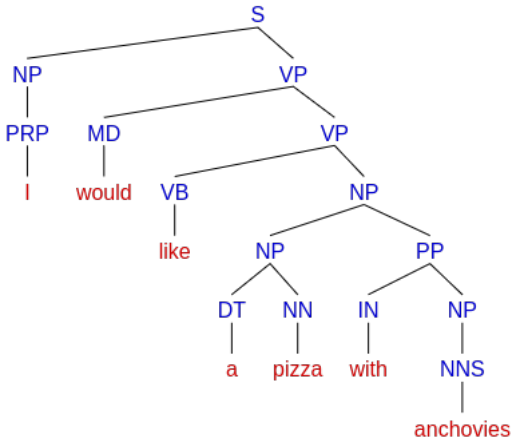


Figure 2.2: An example parse tree of a sentence's phrase structure.

of syntactic categories as well as a set of production rules that can be used to generate other syntactic categories or lexical items. The syntactic categories that are used in phrase-structure grammars consist of part-of-speech (POS) categories (e.g. noun (NN), verb (VB), modal (MD), determiner (DT), preposition (IN), etc.) or phrasal categories (e.g. noun phrase (NP), verb phrase (VP), adjectival phrase (AdjP), etc.). The production rules defined in phrase-structure grammars are of the form "$\text{category} \rightarrow \text{category}^*$" whereby the symbol on the left is rewritten as one or more symbols on the right. An example of a sentence parsed with a phrase-structure grammar is shown in Fig. 2.2. The tree is generated from the production rule "S→NP VP", where S is a starting symbol.

The parse tree shows how POS tags can approximate phrase structure. In this example there are two points of syntactic completion:

I would like a pizza/ with anchovies/.

The two points of syntactic completion correspond to points in the sentence at which a parse is well-formed. We can see that the first syntactic completion point (after "pizza") corresponds to a well-formed parse since the sub-graph "NP→DT NN" can be substituted for the parent NP. It is also the point where the head word ("pizza") forms a maximal projection in the noun phrase ("a pizza").

In a spoken dialogue scenario, these syntactic completion cues would be associated with turn-yielding. Additionally, we can consider parses that are not well-formed to be associated with turn-holding. For example, if we imagine a spoken dialogue scenario, a mid-turn pause after the incomplete utterance "I would like a pizza with ..." is unlikely to elicit a turn-switch,

since the preposition "with" requires an NP complement. However, a similar pause after "I would like a pizza" might elicit a response (e.g. "Which pizza would you like?").

A number of papers have reported POS features as being useful for predicting turn-taking behaviours. Koiso et al. (1998) analysed POS uni-grams in a Japanese task-based corpus and found that they were strong predictors of turn-taking behaviour. Gravano and Hirschberg (2011) analysed POS bi-grams as features for predicting turn-taking, and both found them to be useful, particularly when used in conjunction with prosodic features. Meena et al. (2014) also used bi-grams of the last two POS tags before a silence as features to predict both turn-holding and turn-yielding. While these papers present supporting evidence that syntactic cues play a part in turn-regulation, it is worth re-iterating that spoken conversational dialogue does not tend to follow established syntactic rules in the way that written discourse does. A turn (according to Sacks et al. (1974)) can be comprised of sentence fragments. The presence of fillers (discussed below) and self-repairs in natural conversation also differentiates the syntax of conversational dialogue from that of written discourse.

Prosody Prosody is defined as information about temporal, pitch, and energy characteristics of utterances that are independent of the words (Shriberg et al., 1998). Prosodic patterns of stress and pitch in speech have been extensively studied as turn-yielding and turn-holding cues. Falling or rising pitch at the end of a speaker's turn was associated with turn-yielding in Japanese (Koiso et al., 1998) and English (Gravano and Hirschberg, 2011). Koiso et al. (1998) found that flat, rise-fall, and flat-fall pitch patterns were associated with turn-holding. Wennerstrom and Siegel (2003) also found that rising or falling pitch patterns were associated with turn shifts, while plateaus, low-rises, and partial falls correlated with turn holding. Lowering of intensity at the end of an utterance was found to be a turn-yielding cue by Gravano and Hirschberg (2011).

Wightman et al. (1992) presented results that show syllable lengthening (final lengthening) at the end of phrase boundaries but do not relate them explicitly to turn-taking. The experiments by Gravano and Hirschberg (2011) supported these results and found that there is relatively more lengthening of syllables and phonemes in the final words directly preceding mid-turn pauses (MTPs), than end-of-turns (EOTs). This means that an increase in speaking rate (shorter syllables and shorter phonemes) before turn boundaries is associated with turn-yielding. This contradicts previous hypotheses by Duncan (1972).

There is some debate in the CA literature over the role of prosody as a turn-taking cue. The original paper of Sacks et al. (1974) proposed that prosody plays an important role in

turn-taking coordination, albeit a secondary role to the one played by syntax. Other researchers have presented experimental results that challenge the importance of prosody. De Ruiter et al. (2006) conducted experiments in which utterances were extracted from a spoken Dutch corpus and participants were asked to press a button when they anticipated a turn ending. The utterances were manipulated such that in one experimental condition the pitch information of utterances was filtered out, and in another experimental condition the words were masked but the pitch information remained. The results showed that the accuracy of turn-ending predictions was unaffected when the pitch information was filtered, and severely degraded when words were masked. The authors concluded that "lexicosyntactic structure is necessary (and possibly sufficient) for accurate end-of-turn projection, while intonational structure, perhaps surprisingly, is neither necessary nor sufficient" (De Ruiter et al., 2006, p.531).

Bögels and Torreira (2015) conducted a set of experiments that present a contrasting view of the role of prosody to the one proposed by De Ruiter et al. (2006). In their experiments (also conducted in Dutch) they use a similar button-pressing setup. They record scripted pairs of questions with two versions: (1) a long version, such as "So you're a student at Raboud University?" and (2) a short version, such as "So you're a student?". The short version had turn-final pitch patterns on the final syllable of "student" whereas the long version did not. The two versions were manipulated so that the long version was cut after the word "student" to create a new short version, and the original short version was extended to include the extension in the long version ("at Raboud University?") to create a new long version. The results from this button-pressing experiment showed that participants anticipated turn-endings to occur mid-way through the altered extended version. They also anticipated turn-endings to occur much later in the altered short version than the original short version. The experiment presents strong evidence that listeners do in fact rely on prosody to anticipate turn endings.

The findings of Bögels and Torreira (2015) do not completely contradict those of De Ruiter et al. (2006) since in the experiments of De Ruiter et al. (2006) the only prosodic information that was removed was pitch. Other prosodic information involving temporal and energy characteristics were left intact. The experiments of De Ruiter et al. (2006) still remain strong evidence that listeners rely on syntactic cues for turn-taking projection. The findings of Bögels and Torreira (2015) show that syntax is not necessarily sufficient. Their findings also support the previously discussed evidence from Ford and Thompson (1996) that there is an interaction between syntax and prosody.

Acoustics Acoustic properties of speech have been investigated as turn-taking cues by Ogden (2001), Gravano and Hirschberg (2011), Kane and Yanushevskaya (2014), and Heldner et al. (2019). Ogden (2001) showed that vocal creak was associated with turn-yielding in Finnish. Gravano and Hirschberg (2011) found that three voice quality features were associated with turn-yielding: shimmer, jitter, and harmonic-to-noise ratios. Kane and Yanushevskaya (2014) reported similar findings in their analysis of spectral features. Heldner et al. (2019) showed that turn-yielding is characterized by lower degree of periodicity that is sometimes accompanied by the presence of creaky voice.

Timing Information about the timing of turns and silences has been considered a turn-taking cue in and of itself (Raux and Eskenazi, 2012; Meena et al., 2014; Meshorer and Heeman, 2016). Meena et al. (2014) reported that silences after longer turns correlate with turn-yielding. Razavi et al. (2019) showed that turn length ratio is more useful for EOT prediction than absolute turn-length. They proposed that this suggests that turn-length should be seen as a user-specific feature rather than a global feature. Meshorer and Heeman (2016) also found that using the relative turn length and relative floor control of a speaker could be used to predict the speaker's future turn-taking behaviour. Intuitively, it makes sense that the longer the turn is, the more likely the next silence will be an EOT. However, (perhaps counterintuitively) Raux and Eskenazi (2012) report that *longer* turns are strong indictions of turn-holding. They attribute this observation to the type of data that was being analysed, which was phone conversations between an operator who supplied bus travel information to a user. They found that, in their data, the turns tended to be either very short or much longer. Due to this bimodal distribution, the silences after longer turns by the user were more likely to be MTPs than EOTs. Since studies report different functions of turn-length cues it is likely that these cues are dependent on the situation-specific timing distributions.

The most common turn-taking cue used in IVAs to classify a user's EOT is silence. However, Yngve (1970) notes that silence by a speaker is not itself a turn-taking cue. This is supported by corpus studies which have shown that the lengths of MTPs tend to be longer than the gaps during turn switches (Heldner and Edlund, 2010). Therefore, the use of silence on its own as an EOT signal (as is used in many IVAs) is not necessarily a robust solution.

Lexicon Lexical features have been proposed as turn-taking cues (Duncan, 1972; Sacks et al., 1974; Beňuš, 2009; Gravano and Hirschberg, 2011; Razavi et al., 2019). Duncan (1972) proposes

that stereotyped expressions such as "you know" or "but uh" are used by speakers as turn-taking signals. Duncan's theories were supported by the the findings of Gravano and Hirschberg (2011) where the authors investigated how word uni-grams and bi-grams predicted turn-switches at pauses. They found that these lexical features were strong cues for predicting both turn-holding and turn-yielding. The strength of lexical features as turn-taking cues can be partially attributed to their utility in recognizing fillers and backchannels. *Filled pauses* or *fillers* (discussed in more detail in the following section) such as "um" or "uh", are articulations by a speaker that mark hesitation in human communication (Corley and Stewart, 2008). Fillers have been associated with turn-taking by Sacks et al. (1974), Beňuš (2009), Gravano and Hirschberg (2011), and Razavi et al. (2019). When used by a speaker who currently holds they floor they are associated with turn-holding. They can also be used at the start of an intended turn to grab the floor. *Backchannels* (also discussed in more detail in the following section), are short vocalizations such as "yeah" or "uh-huh" that encourage the previous speaker to continue speaking. Another lexical feature that was analysed by Dethlefs et al. (2016) is information density. Dethlefs et al. (2016) calculated the information density at different points in a speaker's utterance using N-grams. They found that points of high information density were associated with turn-holding while points of low information density were considered more acceptable for overlap.

Semantics *Semantic completion points* are points in an utterance which are a "complete response to the previous speaker's turn" (Hjalmarsson, 2011, p.28). These are distinct from syntactic completion points since they require an understanding of the dialogue context. For example, a sentence fragment such as "the apple" can be a full response to a question such as "Which piece of fruit would you like?" in which case the turn is semantically complete and can be considered a turn-yielding cue. However, in the context of a different question such as "What did you think of that apple?", the sentence fragment is likely to be the start of a longer response e.g. "the apple was delicious."

While a number of studies have proposed semantic features as turn-taking cues it can be difficult to perform semantic analyses of conversations in an objective manner (Oreström, 1983). Raux and Eskenazi (2012) examine how well partial ASR hypotheses match the current dialogue context. At a silence of 200 ms they take the latest ASR result, parse it using a grammar, and then match it against the expected agenda of the dialogue manager. For example if the SDS asked the question "where are you leaving from?" and the user's parsed response corresponds with a place, then the expectation level of the response is high and it is likely to be a semantically

complete response. However, if the parsed user response is "yes" or "no" the response has a low expectation level and it is less likely that the response is semantically complete. They also find that ASR hypotheses that contain positive markers such as "yes" or "sure" were more likely to be semantically complete than other utterances, particularly negative responses which are usually succeeded by an explanation or qualification of the negative response. Other investigations that have analysed semantic features include those of Atterer et al. (2008), Schlangen et al. (2010), Gravano and Hirschberg (2011), and Razavi et al. (2019).

Discourse Structure Studies have reported dialogue act (DA) history information as being useful for predicting turn-taking behaviour (Koiso et al., 1998; Cathcart et al., 2003; Gravano and Hirschberg, 2011; Beňuš et al., 2011; Skantze, 2012; Meena et al., 2014). For example, Meena et al. (2014) found that context of the previous dialogue act uttered by the SDS was a useful predictor for turn-switch behaviour at the user's next end-of-utterance. They observed that if the previous DA by the SDS was a request for clarification, then if the user's response was short (shorter responses are more likely to be "yes" or "no") the SDS should respond with an acknowledgement. Raux and Eskenazi (2012) found that whether the previous question by the SDS was an open question, was a good predictor of turn-internal pauses in the user response. This is likely explained by the fact that responses to open questions (e.g. "What can I do for you?") tend to be longer than closed questions ("Where are you leaving from?") or confirmation prompts ("Leaving from the airport. Is this correct?").

Breathing There have been a number of papers that have reported on respiratory turn-taking cues (McFarland, 2001; Rochet-Capellan and Fuchs, 2014; Ishii et al., 2016b; Włodarczak and Heldner, 2016a) Rochet-Capellan and Fuchs (2014). Rochet-Capellan and Fuchs (2014), Ishii et al. (2016b), and Włodarczak and Heldner (2016a) report that inhalations are associated with turn-holding in German, Japanese, and Swedish (respectively). Rochet-Capellan and Fuchs (2014) found that the duration of inhalations is often reduced by speakers during MTPs as a way of minimizing the possibility of an interruption. Włodarczak and Heldner (2016a) observed that inhalations have higher acoustic amplitudes before a speaker change. Exhalations were also shown to be longer at silences where the speaker was yielding the turn (McFarland, 2001).

Respiratory features can be used to not only predict whether a speaker is about speak, but also what they are about to say. Torreira et al. (2015) and Włodarczak and Heldner (2016a) found that short utterances similar to feedback expressions (e.g. backchannels) were often produced

on residual air. Longer responses were found to require a planned inhalation.

Vision Recently, the growing interest in embodied agents such as graphical agents (e.g. Traum and Rickel (2002); Gratch et al. (2007); Hjalmarsson and Oertel (2012)) and humanoid robots (e.g. Al Moubayed et al. (2012); Inoue et al. (2016)) has provided motivation to exploit the visual turn-taking cues that are used in human face-to-face interactions. Eye-gaze was proposed as a source of turn-taking cues by Kendon (1967). In the paper he notes that speakers tend to look away from their conversational partners at the beginnings of their turns, especially when the turn is going to be long. Speakers then tend to look back at their partners within the vicinity of the end of their turn. As such, *gaze aversion* can be considered a turn-holding cue while looking at the addressee can be considered a turn-yielding cue.

A number of other facial gestures have been associated with turn-taking. Ishii et al. (2016a) found that mouth-opening was associated with turn-grabbing during Japanese conversations. They note that directly preceding a turn, the speaker will often open their mouth slightly. This is also connected with breathing behaviours discussed above. Eye-brow raising was associated with turn-grabbing by Cavé et al. (1996) and may be linked with attention-signalling (Guaïtella et al., 2009). Gestures such as nods Bavelas and Gerwing (2011) and eye-blinks Hömke et al. (2018) have also been associated with listener feedback behaviour that encourage a speaker to continue.

2.1.3 Dialogue Acts

When we view each utterance in a conversation as being an action that is performed by the speaker, the actions are referred to as *dialogue acts*. In this subsection we discuss aspects of dialogue acts in spoken conversations that are relevant to modelling turn-taking in SDS.

Dialogue Acts Dialogue acts (e.g. *statement*, *question*, *agreement*) provide a means by which it is possible to represent dialogue, both from the perspective of conversational analysis, and from an dialogue modelling perspective. They provide a means by which to represent functionalities that a dialogue system can perform. A formal definition of a dialogue act given by Bunt (2009, p.1) is: "a unit in the semantic description of communicative behaviour in dialogues, specifying how the behaviour is intended to change the information state of the addressee through his interpretation of the behaviour. " A wide variety of dialogue act annotation taxonomies have been proposed (Stolcke et al., 2000; Bunt, 2009; Bunt et al., 2010). Typically, if a dialogue

system uses a dialogue act representation, the types of dialogue acts that it will be able to perform will usually be tailored for the specific use-context of the system (e.g. *recommend-movie*, *query-food-preference*).

However, in real human-human dialogue, labelling utterances for dialogue acts may not be straightforward. An utterance in dialogue may serve multiple communicative functions at the same time (Bunt, 2007; Petukhova, 2011). For example, here is an excerpt from Malchanau (2018, p.20):

> C1: I would suggest we do not allow smoking in public places
>
> B1: Uh-uhu
>
> C2: What do you think?
>
> B2: Uhm... yeah ... it's a bit difficult for me

In the B1 turn the speaker acknowledges the suggestion in C1, however it may also have the function of stalling for time. When speaker C asks for a reaction in C2, the response B2 has multiple functions. The "Uhm..." filler segment can be interpreted as a stalling component. The segment "yeah..." is both a stalling component as well as positive feedback. Then "it's a bit difficult for me" is both an answer to the question in C2, while also being a negative response to C1.

This multifunctional aspect of dialogue is also complicated when multiple modalities are involved. Inflections in tone can change change the semantic content of an utterance. For instance, the meaning of the utterance "Are you mad?" can be changed from a query of whether a person is angry, to an accusation, by changing the prosodic inflections and dialogue context. Therefore, analysing the utterance from a purely lexical standpoint may not provide all the information necessary to correctly interpret its intended functionality.

Dialogue is often structured in such a way that the type of dialogue act that will succeed another is predefined. Pairs of dialogue acts where the second dialogue act is constrained by the first are referred to as *adjacency pairs*. Adjacency pairs were first discussed by Schegloff and Sacks (1973), and are defined in Levinson (1983, p.303) as:

> Sequences of two utterances that are: (i) adjacent (ii) produced by different speakers (iii) ordered as a first part and a second part (iv) typed, so that a particular first part requires a particular second (or range of second parts).

Common examples of adjacency pair types include: *question-answer* ("What is the time?" "Five

o'clock"), *offer-acceptance* ("Would you like Tea?" "Yes, please."), and *greeting-greeting* ("Hi." "Hello.").

The types of dialogue acts that are involved with turn exchanges also affect the distributions of turn switch timing offsets (Strömbergsson et al., 2013; Kendrick and Torreira, 2015; Heeman and Lunsford, 2017). As mentioned previously, the varying levels of cognitive load required to answer different types of questions (e.g. *wh-question*, *yes-no-question*) can influence the distributions of response offsets (Strömbergsson et al., 2013). However, it has also been shown that the type of response (e.g. *yes*, *no*, *agree*, *disagree*) can influence the timing. In particular, *dispreferred* responses, where the response goes against the expected response to the previous turn, have been associated with latencies that are greater than 800 ms (Kendrick and Torreira, 2015). It has also been shown that listeners ascribe semantic meaning to silences, such that dispreferred responses are anticipated after longer response offsets (Bögels et al., 2019). The timing offsets in turn exchanges are therefore dependent on the context of both the starting turn and the responding turn.

Grounding *Grounding* is an important component of spoken dialogue. It is establishing what the dialogue participants agree on and acknowledging that a message is being understood. *Backchannels* are a common form of grounding. Backchannels are short utterances spoken by a listener such as "Yeah" or "Uh-huh" that communicate to the speaker that the listener is paying attention and encourage the speaker to continue (Yngve, 1970; Duncan, 1972; Ward and Tsukahara, 2000). The generation of backchannels by SDSs can be used to aid the coordination of the interaction by signalling to the user that the SDS is registering the user's speech. Backchannels are particularly useful when the user takes long turns, such as in the case of a user placing a long restaurant order. The detection of backchannels uttered by a user is also important, so that they are not misconstrued as barge-in.

Backchannels have been reported to typically occur near TRPs (Goodwin, 1981) as a signal to the speaker that the listener does not intend to take a turn. Schegloff (2000) considers backchannels (referred to in his paper as *continuers*) to not be full turns but rather a signal that the listener wishes to pass on taking a turn. Schegloff regards them as independent of the normal "one-at-a-time" rules defined by Sacks et al. (1974) (discussed previously in Section 2.1.1). As such, the occurrence of backchannels in overlap with a speaker's turn does not violate the normal rules of turn-taking. Support for Schegloff's view is presented in a corpus study by Levinson and Torreira (2015) where they found that 73% of overlaps occurred in the presence of

a backchannel. They note that in approximately half of these overlaps it was the main speaker that started speaking again in overlap with the backchannel. They also found that overlapping backchannels often occurred after a TRP or a period of silence, suggesting that their timing is sensitive to cues in the main speaker's turn. Several researchers have proposed methods of detecting these backchannel cues during speech (e.g. Ward and Tsukahara (2000); Morency et al. (2010); Dethlefs et al. (2016); Truong et al. (2011)). Ward and Tsukahara (2000) found that low-pitch regions were a cue for backchannels. Truong et al. (2011) found that gaze was a strong backchannel cue. Dethlefs et al. (2016) proposed that there is a relationship between information density and suitable places for backchannels or barge-ins in spoken conversation.

There are sometimes inconsistencies in how backchannels are labelled since the definitions of what constitutes a backchannel are often ambiguous. In the labelling scheme used by Gravano and Hirschberg (2009, p.1020) they distinguish between backchannels and other DAs as utterances that only indicate "I'm still here / I hear you and please continue." However, this labelling scheme is subjective since it is up to the labeller to interpret whether there is any indication of agreement in the utterance based on prosodic cues and pragmatic context. Jurafsky et al. (1998) note that prosodic inflections play a role in resolving ambiguities about dialogue act types for backchannels. There are situations where a "Yeah" utterance could be perceived as an encouragement for the speaker to continue, an agreement, or both, based on prosodic inflections.

Fillers *Fillers* or *filled pauses* such as "uhh..." or "hmm..." are another important component of natural spoken dialogue. They signal to the listener that the speaker is in the process of formulating a response. They can be employed by SDSs as a way of compensating for processing delays and have been found to make interactions more natural (Goble and Edwards, 2018; Lala et al., 2019b). Nakanishi et al. (2018) analyse the types of fillers used in Japanese spoken conversation. They argue that certain forms of fillers are used within certain contexts. The forms are Proper ("um"), demonstrative ("so"), adverbial ("well"), and notice ("oh"). Notice fillers are used more during switches while demonstrative and proper are used more in turn-keeping. They also found that linguistic information about the following turn was important to the prediction of what type of filler was used during a turn-switch.

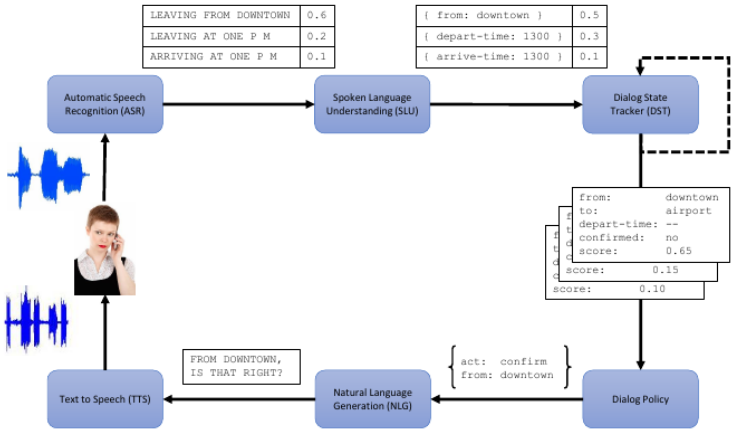


Figure 2.3: The components of a task-based SDS. From Williams et al. (2016).

2.2 Dialogue Systems

In this section we discuss the relevant details about dialogue systems. We include discussions about text-interactive dialogue systems since their architectures are relevant to the design of spoken ones as well. In some cases the same dialogue system can be interacted with using both text and speech. We discuss two main types of dialogue systems: *task-based* dialogue systems and *social chatbots*. We distinguish between the two types by noting that task-based SDSs are generally optimized to efficiently complete a joint task with the user, whereas social chatbots are designed to maintain the engagement of the user for extended periods of time. Therefore in task-based systems it is desirable for the interaction to run swiftly and efficiently, while in chatbots it is desirable for the conversation to be extended and mimic the properties of human social conversation. The distinctions between these two types of systems can often be blurred. For example, chatbots often contain task-based components (playing music, managing calendars, etc.) where it may be desirable for a sub-task to be completed in an efficient manner. In these cases, the completion of sub-tasks in a timely manner is desirable for the maintenance of the longer interaction as a whole. The turn-taking models that we propose in this thesis can potentially be applied to both varieties of dialogue systems. After introducing these two main types of dialogue systems we then present an overview of research on *incremental processing* in SDSs.

2.2.1 Task-Based Dialogue Systems

Most of the commercial SDSs (such as IVAs) can be considered task-based dialogue systems. The architecture used in task-based SDSs is typically a variant of the slot-based dialogue system shown in Fig. 2.3. There are six components that are usually used in these SDSs: automatic speech recognition (ASR), spoken language understanding (SLU), dialogue state tracker (DST), dialogue policy, natural language generation (NLG), and text-to-speech (TTS). Jointly, the DST and the dialogue policy are often referred to as the dialogue manager (DM). The role of the NLU component is to identify the intents of the user and extract the associated information. The DST maintains the state of the dialogue, which is the systems belief state over the essential information that has been spoken by the user and the system so far. The dialogue policy component uses the dialogue state to decide on the next action that the dialogue system will take. The NLG receives a semantic representation from the DM that is then converted into natural language. The natural language is then converted into speech using the TTS component.

Slot-based SDSs (e.g. Raux et al. (2005); Young et al. (2013)) typically use dialogue acts as the semantic representations of the user utterances (as parsed by the SLU), as well as the output of the DM that is sent to the NLG component. In implementations such as the Ravenclaw architecture (Bohus and Rudnicky, 2009) turn-taking modelling is integrated into the DM component. In general the components of task-based architectures are trained separately rather than end-to-end. The dialogue policy component is usually trained using reinforcement learning (RL). Markov decision processes (MDPs) have been used (Walker, 2000; Levin et al., 2000; Singh et al., 2002), as well as partially observable Markov decision processes (POMDPs) (Williams and Young, 2007), to train the dialogue policy. POMDPs have generally been favoured over MDPs since they can be used to account for ASR confidence scores.

2.2.2 Social Chatbots

Social chatbots can either use variants of the task-based architecture described above, or alternatively they can use end-to-end architectures. Microsoft's XiaoIce (Zhou et al., 2020) (described in Section 1.1.1) uses an *options over MDPs* (Sutton et al., 1999) which is a variant of the task-based architecture described above in which decisions are learned in a hierarchical manner. The end-to-end architectures are typically based on encoder-decoder models (Vinyals and Le, 2015; Sordoni et al., 2015) that were originally developed for machine translation (Sutskever et al., 2014). Vanilla encoder-decoder models tend to produce predictable, repetitive, and dull

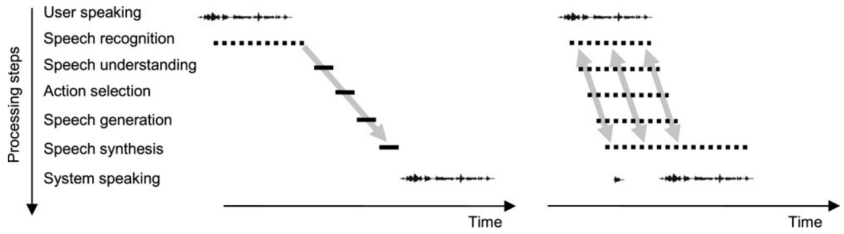


Figure 2.4: The difference between non-incremental SDS processing (left) and incremental processing (right). Dashed lines indicate incremental processing. Typically ASR is performed incrementally even in non-incremental SDSs however the partial recognition results are not used. From Skantze and Hjalmarsson (2013).

responses like "I don't know" or "OK" that bring the conversation to a halt. This can be addressed in part by changing the objective function to maximum mutual information (MMI) or altering the beam search in the decoder to encourage diverse responses (Li et al., 2016). Encoder-decoder models also suffer from an inconsistency across responses. Serban et al. (2016) developed a hierarchical recurrent encoder-decoder (HRED) model to help maintain context over multiple responses. Transformer models (Vaswani et al., 2017) have also achieved good results in modelling open-domain chat. More recently Adiwardana et al. (2020) used a large 2.6 billion parameter transformer to train their model called "Meena", achieving close-to-human ratings on an evaluation metric that measured sensibleness and specificity (SSA). The authors propose that these types of models could be used for making interactions with computers more human-like.

2.2.3 Incremental Processing

The main principle behind incremental systems is that the components of the SDS process their respective inputs in small chunks throughout the interaction with a user. They are not limited to just making decisions after a user's EOT has been detected. Since chunks from the user's speech can be processed while the user is speaking, this enables a reduction of the processing delays needed to react to the user's speech as shown in Fig. 2.4. Incremental processing has been found to produce interactions that are perceived as being more natural than interactions with systems that adopt a turn-based processing strategy (Skantze and Hjalmarsson, 2010; Skantze and Schlangen, 2009; Tsai et al., 2019).

The ability of incremental systems to react to partial ASR results enables them to potentially perform a number of naturalistic behaviours that are typically unavailable to non-incremental systems. For instance, the reduced processing delays allow for faster endpointing decisions (Skantze and Hjalmarsson, 2013). However, many of the benefits of incremental systems lie in

their potential to make predictions about likely user turn continuations. For example, certain words are associated with a high probability for the following word or the continuation of the sentence. If the word "San" appears in an english utterance, the word "Francisco" is of very high likelihood to be the continuation. There are other possibilities (e.g. San Diego), however we can be almost certain that the continuation will at least be the name of a place. Using this information, incremental systems (and humans) can plan their utterance before the word "Francisco" and restrict the response search space. The same has been shown for full utterances (Bögels et al., 2018). If certain key words of a spoken question are arranged in such a way that they occur earlier, rather than later in the utterance, the listener is able to respond more quickly.

The system can exploit these likely turn-continuations to anticipate the user's EOT. Using the predictions the system can generate fast turn-switches that involve either small response offsets or small amounts of intentional overlap with the end of the user's turn (Dethlefs et al., 2012). The ability of being able to generate overlap is useful for generating naturalistic backchannels and feedback behaviours. Being able to react to incomplete user utterances also allows the detection of user backchannels that occur while the system is speaking. These user backchannels might normally be considered barge-in by a non-incremental system, but if the system can classify them as backchannels based on early predictions, the system can avoid unnecessarily stopping its turn. Incremental processing also enables principled interruptions of a user in order to repair an error, or miscommunication. Interruptions can also be used to avoid unnecessary turn continuations by the user. For example if we consider the following hypothetical interaction of a user with an incremental SDS that takes orders at a pizza restaurant:

U: I would like to order two pizzas. One with anchovies, the other...

S: I'm sorry we're all out of pizza.

The interruption by the system increases the efficiency of the dialogue and is a potentially desirable behaviour.

Most incremental SDSs are designed around being able to make decisions based on partial ASR hypotheses. Many modern commercial and open-source ASR systems operate in an incremental fashion (Baumann et al., 2017) where partial recognition hypotheses are output after they are spoken by the user (with some variable amount of latency). An issue with these ASR hypotheses is that previously output ASR hypotheses may be revised as the user's speech progresses. Unstable ASR partials can have a knock-on effect to the other components in the processing chain. This means that in order to avoid reacting to unstable ASR partials, it is de-

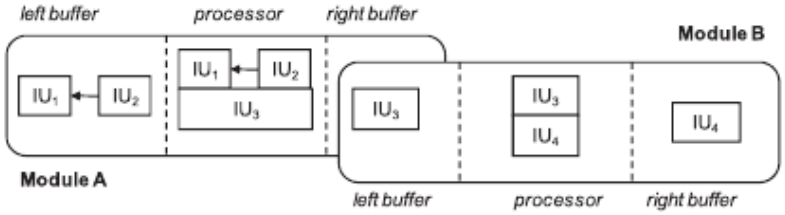


Figure 2.5: Two incremental modules connected together. From Skantze and Hjalmarsson (2013).

sirable for the components of incremental systems to have ways of revising previous component outputs, as well as have confidence measures for their outputs. Methods for determining when to finalize ASR partials have been proposed by Raux and Eskenazi (2008) and Lu et al. (2011).

A theoretical framework for designing incremental SDSs that can handle repair was proposed by Schlangen and Skantze (2011). Their framework is based on encapsulating the functionality of the standard SDS components into incremental *modules*. Modules consist of a *left buffer*, a *processor* and a *right buffer* and are connected together as shown in Fig. 2.5. Modules are designed to process *incremental units* (IUs) which are defined as the smallest chunks of information that can trigger the output of a module. For example, an incremental unit for the input to an SLU component could be a word, while an incremental unit for the input to the TTS could be segments of speech similar to IPUs (Skantze and Hjalmarsson, 2013). IUs can be grouped together to denote their joint functionality using *same-level links*. For example words can be grouped into utterances. IUs that were used to create other IUs are connected together via *grounded-in links*. These grounded-in links are used for the purpose of repair when information gets revised. An example given by Schlangen and Skantze (2011) is when a user is speaking "forty-four", the ASR module may output an IU with an initial hypothesis of "four" at the beginning of the user's utterance when it is incomplete. This IU may then be used in the SLU that causes further downstream IUs to be processed by other modules. Once the ASR module corrects the initial hypothesis to "forty-four", the downstream IUs that were grounded in the initial "four" IU can be revoked using the grounded-in links. Implementations that are directly based on the incremental framework of Schlangen and Skantze (2011) have been presented by Baumann and Schlangen (2012), Skantze and Hjalmarsson (2013), Kennington et al. (2014), and Michael and Möller (2019).

A number of papers have presented approaches to individual components found within incremental SDS pipelines. Incremental ASR was discussed by Baumann et al. (2017). They note

that while most ASR systems are already incremental, many off the shelf solutions do not output information that may be important for downstream applications, such as disfluencies and timing information. Shivakumar et al. (2019) propose an RNN-based incremental SLU component that runs in parallel with the ASR results. The issues associated with incremental NLG were discussed by Baumann and Schlangen (2013). They note that in most cases however, it is acceptable to have a non-incremental NLG. Generating incremental TTS is complicated by the fact that the quality of the synthesis often depends on knowing longer segments of an utterance in order to model co-articulation, stress patterns, and sentence level intonations (Taylor, 2000). Baumann and Schlangen (2012) present a way of making a non-incremental TTS system incremental.

When trying to emulate naturalistic turn-taking behaviours in SDSs, part of the appeal of incremental systems is that they do not have to rely on waiting for the final results of an ASR component for a system to respond. The final output of an ASR system could be delayed by more than a second after a user has finished speaking. But if the DM has already formulated a response, endpointing can be performed and a response can be triggered without waiting for the final ASR results. This motivates the desire for designing independent turn-taking models that don't only rely on features which require ASR. In the following section we give an overview of these turn-taking models.

2.3 Turn-Taking Models

In this section we discuss the relevant aspects of turn-taking modelling for SDSs. Firstly, we discuss the different varieties of turn-taking decisions that can be made in SDSs and how they are made. We then discuss neural approaches to turn-taking modelling. This is then followed by a more in-depth description of continuous turn-taking (CTT) modelling which is relevant to the models proposed in subsequent chapters. We then discuss the datasets used in this thesis.

2.3.1 Turn-Taking Decisions

Perhaps the most common turn-taking modelling decision that is made is *endpointing* or *end-of-turn* (EOT) detection. This involves making the decision between whether a detected threshold segment of silence by the user is an EOT or an MTP. Some form of machine learning model is typically employed using features that can be extracted from the user's speech and the dialogue context to make these predictions. There is usually a trade-off between the latency of the

response and how often the SDS interrupts the users, referred to as *false cut-ins* (FCIs).

In a model proposed by Ferrer et al. (2002), decision trees were applied in a frame-by-frame manner during silences to detect an EOT. The model used prosodic and linguistic features extracted from the user's speech in combination with the length of the silence so far. As prosodic features they use F0 summary statistics calculated from the speaker's utterance (Sönmez et al., 1998). As linguistic features, they used an N-grams with an added EOT token.[2] Ferrer et al. (2002) were able to control the trade-off between latency and FCIs by adjusting the pause threshold parameter used in the decision tree. Raux and Eskenazi (2008) refine this thresholding approach by using contextual dialogue features to optimize the choice of threshold. They first cluster silences based on dialogue features, then a single threshold is set for each cluster. The objective is to have low-threshold clusters that contain mostly EOTs and short MTPs, and longer threshold clusters that contain long MTPs and very few EOTs.

A cost-based approach to making endpointing decisions was proposed by Raux and Eskenazi (2009) as an alternative to thresholding. Their *finite-state turn-taking machine* (FSTTM) uses the observation of Sacks et al. (1974, p.704) that participants in a conversation attempt to "minimize gaps and overlaps" to design a state-based model of turn-taking. Their model assigns costs to the system being in either a "gap" or an "overlap" state. The trade-off between latency and FCIs can then be controlled by changing the ratio between the two costs. The FSTTM model is discussed in detail in Chapter 5 where it informs a control process we propose that uses POMDPs.

Since barge-in can be used to improve the efficiency of interactions, several publications have proposed barge-in models. In (Dethlefs et al., 2012) they use hierarchical reinforcement learning to generate backchannels and barge-ins. The model used information density to identify suitable points for both backchannels and barge-in. Zhao et al. (2015) proposed an incremental turn-taking model for barge-in that used RL that was trained through user simulation. Khouzaimi et al. (2015) also used a simulation and RL to make incremental turn-taking decisions. They introduced a "scheduler" module into the incremental SDS pipeline that uses all new partial ASR hypotheses to decide whether to grab the floor or not. The scheduler policy is trained to maximize the overall task success rate as well as minimize the overall time taken for the interaction. In the simulated interactions, the time is minimized through the use of barge-in to correct simulated non-understandings or invalid user requests.

[2]It is worth noting that, unless otherwise specified, the term "linguistic features" in this thesis typically refers to lexical features (i.e. uni-grams or N-grams) rather than more abstract features such as grammatical features (i.e. tense, number, person).

2.3.2 Neural Turn-Taking Models

In many of the turn-taking models described above, features that capture turn-taking cues are input into traditional classifiers such as logistic regression (Raux and Eskenazi, 2009), naive Bayes (Razavi et al., 2019), SVMs (Gravano and Hirschberg, 2011), or decision trees (Ferrer et al., 2002). An issue with using these types of classifiers for turn-taking modelling is that they are non-sequential. Many of the turn-taking cues described in Section 2.3.1 rely on particular sequential organization, such as specific pitch inflections in prosody, or specific patterns of syntactic or lexical tokens. To capture this sequential information in traditional classifiers, these patterns must be summarized using hand-engineered summary features. For example, for prosodic information, a common approach is to calculate descriptive statistics (e.g. mean or slope) of either pitch or intensity over a window of 200 to 500 ms from the end of a speaker's IPU (Gravano and Hirschberg, 2011; Meena et al., 2014; Razavi et al., 2019). The choice of the window length that these statistics is calculated over will affect the performance of the classifier. If the size of the window is too large the relevant information may be smoothed. If the window is too small, important details may be omitted. To model lexical or syntactic patterns, bi-grams or tri-grams can be used. However, it is not feasible to use N-gram models to capture longer term dependencies that may be important for detecting semantic or syntactic completeness. These temporal considerations can be considered a limitation of non-sequential models.

The use of sequential models such as RNNs allows these patterns to be learned without having to hand-engineer the features. In recent years there has been a large amount of interest in using RNNs to model turn-taking (Maier et al., 2017; Skantze, 2017b; Masumura et al., 2017). Maier et al. (2017) used an LSTM (Hochreiter and Schmidhuber, 1997) with automatically extracted acoustic and linguistic features as input features to make frame-based sequential predictions of three classes: speech, MTP, and EOT. They combined the LSTM predictions with a thresholding-based approach to calculate the trade-off between latencies and FCIs. They found that the combination of linguistic and acoustic features used in the LSTM produced better results together than when used separately. They concluded that the LSTM was able to benefit from a fusion of the of the two modalities.

Masumura et al. (2017) proposed their stacked time-asynchronous sequential network (STASN) to perform endpointing after every user IPU in call centre telephone interactions. Their STASN model uses individual LSTMs to independently model acoustic features, phonetic features (senone bottleneck features), and lexical embeddings. Their model takes the final LSTM hidden states

from each of the feature-level LSTMs, concatenates them, and then uses them as an input to another LSTM that makes the binary EOT/MTP prediction. Masumura et al. (2018) proposed changes to their STASN model by incorporating the operator's dialogue context and perform endpointing without using lexical features.

Skantze (2017b) proposed a continuous turn-taking (CTT) model in which the future speech activity of speakers is predicted sequentially at 50 ms timesteps using an LSTM. The objective is to predict a vector of the future ground truth speech activity labels (60 frames or 3 seconds). The model is notable in that it aims to predict future speaker behaviour, rather than simply reacting to behaviour that has already occurred. The trained model is able to capture general information about the turn-taking behaviours in the data. The general information can be used to make endpointing decisions as well as other predictions such as whether a user's utterance is likely to be short like a backchannel. The model is trained using acoustic and POS features that are concatenated and input to the LSTM at each timestep. The reported results indicate that, while both acoustic and POS features contribute to the overall best performance, the acoustic features alone achieve results that are close to the combined best. In a comparison with traditional classifiers on the same dataset, CTT was able to achieve substantially better performance than non-sequential classifiers (naive Bayes, logistic regression) on an endpointing prediction task. We describe CTT in more detail below in Section 2.3.3 since it is used extensively in other chapters.

When surveyed together, it becomes apparent that sequential neural models such as RNNs usually report strong performance using acoustic features that model prosodic behaviours. This contrasts with the observation that prosodic features tend to not be as effective as linguistic features when used in non-sequential traditional models. For example, textual completion was found to have larger absolute value logistic regression coefficients than prosodic and acoustic features by Gravano and Hirschberg (2011). Razavi et al. (2019) achieved better performance using lexical and syntactic features in non-sequential classifiers (SVM, decision trees, naive Bayes) than using prosodic features. Raux and Eskenazi (2012) go so far as to say that, in the presence of good semantic features, the prosodic features might be redundant.

While sequential neural models such as CTT have been reported to outperform models based on hand-crafted features (Skantze, 2017b), there are also drawbacks to their use. Since RNNs operate on the basis of learning over-parameterized non-linear functions, our ability to gain useful scientific insight on behavioural patterns that have been learnt by the model is reduced. Studies

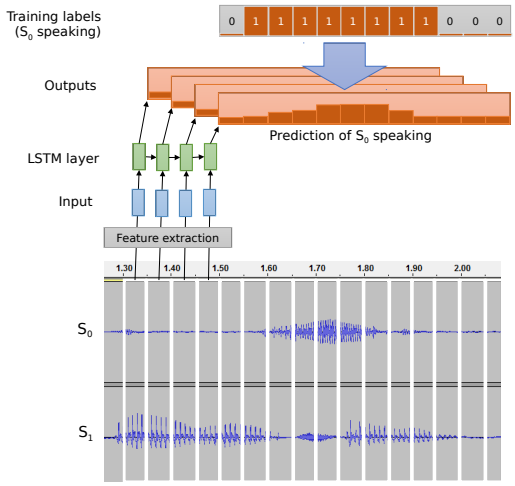


Figure 2.6: At each 50 ms frame, the model makes probabilistic predictions for individual future frames within a window of length N. Figure from (Skantze, 2017b).

such as that of Gravano and Hirschberg (2011), can draw specific conclusions about prosodic and linguistic turn-taking cues observed in data by analysing the co-occurrence of hand-crafted features with turn-taking behaviours. With neural approaches, the objective is to allow the network to automatically learn feature representations that allow us to predict these turn-taking behaviours. Through the process of automatically learning these representations, we lose some of our ability to interpret what has been learnt by the model. While it is still possible to draw some conclusions based on which features are supplied as input to the model, the exact nature of how these features are used, and how features interact with one another, is obfuscated. There is therefore a trade-off between model performance and our ability to draw conclusions about the behaviours learnt by the models.

This loss of interpretability when using "black box" neural models (Lipton, 2016) has several other consequences for turn-taking modelling. We lose the ability to easily debug problems when models make mistakes. With models based on hand-crafted features, if we discover that the model is making a critical mistake when exposed to certain conditions it is typically straightforward to adjust model parameters to address the mistake. With neural models it is impractical to manually adjust model weights to similarly address the mistake. Relatedly, with neural models there is often uncertainty about how the models will generalize to new data that is outside of the original training distribution.

2.3.3 Continuous Models

In this subsection we describe CTT as proposed by Skantze (2017b) in detail. Fig. 2.6 shows how LSTM networks are applied to make CTT predictions. The main objective is to predict the future speech activity annotations of one of the speakers in a dyadic conversation using input speech features from both speakers (S_0, S_1). At each timestep (t) of size 50 ms, speech features (x_t) are extracted and used to predict the future speech activity of one of the speakers. The future speech activity is a 3 second window comprising of 60 frames of the binary annotations for frames $t+1$ to $t+60$. We represent the ground truth objective for a given timestep t as $\boldsymbol{y}_t = [y_{t+1}, y_{t+2}, ..., y_{t+N}]$ where y_t is the ground truth at timestep t, and $N = 60$. The output layer of the network uses a sigmoid activation to predict a probability score for the target speaker's speech activity at each future frame. The network uses a single LSTM layer with a variable number of hidden nodes. The dataset is divided into non-overlapping sequences of length $T = 1200$ timesteps (60 seconds) and each conversation in the data is used twice, with the positions of S_0 and S_1 swapped. Skantze (2017b) proposes using a mean absolute error (MAE) loss, where the loss for a sequence from timestep t to $t+T$ is:

$$L_{\text{MAE}} = \frac{1}{NT} \sum_{t}^{t+T} \sum_{n=1}^{N} |\boldsymbol{Y}_{t,n} - \hat{\boldsymbol{Y}}_{t,n}| \tag{2.1}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_t, \boldsymbol{y}_{t+1}, ..., \boldsymbol{y}_{t+T}]$ is a matrix containing the ground truth objective vectors, and $\hat{\boldsymbol{Y}}$ is a matrix of the predicted vectors. The features used were: the current voice activity (VA) from each frame, pitch features, power, spectral stability, and POS. The POS features were represented using a one-hot encoding of 59 different POS tags. The POS features were set to 0 by default. Then, 100 ms after a word was spoken, the corresponding feature was set to 1 for one frame. The 100 ms lag was done to simulate the delay caused by ASR. The HCRC MapTask corpus (described below in 2.4) was used to train and evaluate the model.

Prediction at Pauses The prediction at pauses task (PAUSE) represents the standard turn-taking decision made at brief pauses in the interaction to predict whether the person holding the floor will continue speaking (HOLD) or the interlocutor will take a turn (SHIFT). To make this decision, all points where there is a pause of a minimum set length are located. Then all of these instances where only one person continued within a one second window directly after the pause are selected. The predicted output probabilities within the window for each of the speakers at the frame directly after the pause are averaged. The speaker with the higher average score

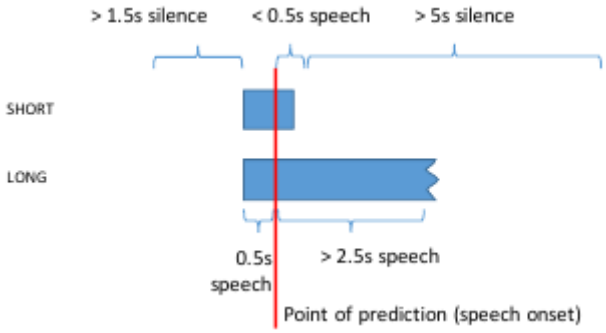


Figure 2.7: Predictions at onsets. Definitions of SHORT and LONG utterances. Figure as in (Skantze, 2017b)

is selected as the predicted next speaker, giving a binary HOLD/SHIFT classification for which F-scores are reported. For all reported F-scores, "weighted" F-scores are used, where the metrics are calculated for both labels and a weighted average is used to take into account imbalance between labels.

Prediction at Onsets Fig. 2.7 shows the prediction at onsets task (ONSET) which represents a prediction of the length of an utterance after an initial period of speech. It represents a useful decision for estimating how long the upcoming utterance will be. It categorizes onset predictions into SHORT and LONG, where SHORT utterances can be considered similar to backchannels. For an utterance to be classified as short, 1.5 seconds of silence by a participant has to be followed by a maximum of 1 second of speech, after which the speaker must be silent for at least 5 seconds. For the utterance to be classified as long, 1.5 seconds of silence must be followed by at least 2.5 seconds of speech. The point at which the predictions are made is 500ms from the start of the utterance. The prediction is made by taking the mean of the 60 output nodes from the sigmoid layer and comparing them to a threshold value. During the training stage this threshold is determined by finding the value that best separates the two classes.

2.4 Datasets

In this section we give descriptions of the corpora used in this thesis. Details of how the datasets are prepared, in terms of segmentation and feature extraction, differ in many of the experiments and so are discussed on a chapter by chapter basis. The corpora were selected based primarily on three main criteria. Firstly, we are interested in the interactions between different modalities, so it is desirable that the datasets contain more than one modality (by modality here we mean

either audio recordings, text transcriptions, or video recordings). It is also important that the quality of the target modality in the dataset be of a high standard. By this we mean, for example, that the audio has clean separation between channels, the transcripts are reliable and have good timing, and that the video recordings are taken from appropriate angles. Secondly, the size of the datasets needed to be large enough to allow the training of deep learning models. Thirdly, we deliberately chose to use corpora that featured naturalistic human-human interaction as opposed to human-machine dialogues (e.g. Raux et al. (2005)) or dialogues that featured acted or affected (e.g. McKeown et al. (2010)) interactions. Since our aim is to develop models that can enable naturalistic interactions with SDSs it was reasoned that the use of human-machine dialogues would affect the realism of the turn-taking behaviours. Although the use of human-machine data (as in Raux et al. (2005)) might in some respects be easier to model (due to a constrained vocabulary and predictable dialogue structure), the data is also skewed by the original SDS implementation used during the data collection. Flaws in different SDS components could cause the users to attempt to compensate for these flaws and, in the process, produce unnatural behaviours (e.g. the "computer talk" discussed by Fischer (2006)). Based on the three criteria listed above, we selected the following three datasets.

HCRC MapTask Corpus The HCRC MapTask corpus (Anderson et al., 1991) is a publicly-available corpus of conversational task-based dyadic interactions in Standard Scottish English. The participants take part in a co-located joint task where the participants sit facing one other and are each given a piece of paper with a map on it. The maps are placed on stands such that the other person's map can't be seen. The maps have approximately a dozen labelled features (e.g. "oak forest", "green bay") on them. One map has a route drawn on it and the other does not. One participant, the "giver", instructs another participant, the "follower", on drawing the route that is drawn on the giver's map. Sometimes differences in the features were introduced between the two maps. In half of the interactions a barrier was placed between them to prevent eye-contact. The corpus is comprised of 128 dyadic conversations with approximately 18 hours of speech. Each participant's speech was recorded using close-talk microphones on individual channels with minimal crosstalk using 16-bit 20 kHz audio. The corpus includes the following annotations: orthographic transcriptions that include accurate word timings, part-of-speech, disfluencies, fillers, gaze, and dialogue acts (referred to in the corpus as "dialogue moves").

Switchboard Corpus The Switchboard corpus (SWBD) (Godfrey et al., 1992) is a corpus of telephone conversations in Standard American English that was originally collected in 1990 by Texas Instruments. The version of the corpus we use is Switchboard-1 Release 2 as distributed by the Linguistic Data Consortium (LDC) (Godfrey and Holliman, 1993). The corpus consists of 2438 dyadic telephone conversations with 543 different speakers and approximately 260 hours of speech. In each call, two participants were asked to discuss a topic that was selected from a list of 70 different topics (e.g. crime, music, football) by an automated prompting system. The data collection process was constrained such that no two participants conversed with each other twice, and no single participant spoke on the same subject more than once. The audio for each speaker was recorded using an 8 kHz sampling rate and mu-law encoding. The audio quality varies depending on the conversation. Some calls contain cross-talk between the channels and some contain audible echo.

There have been several transcription and annotation efforts since the recording of the corpus was completed. In this thesis we use two different sets of annotations. The first is the Mississippi State University (MS-State) transcriptions (Deshmukh et al., 1998) which provide word-level transcriptions of the utterances for all of the 2438 conversations. These are used for the extraction of segmentation, timing, and lexical annotations. The second set of annotations is the NXT-format switchboard corpus (NXT) (Calhoun et al., 2010) which is an effort to merge annotations from different sources in one central resource. The NXT corpus includes annotations for syllables, phones, syntax, disfluency, dialogue acts, coreference, animacy, and prosody. We use the NXT corpus for the dialogue act annotations that were originally part of the SWBD-DAMSL set of annotations (Jurafsky et al., 1997). Since the SWBD-DAMSL annotations does not include word timing information (which is necessary for our work in Chapter 6), the NXT corpus provides a way of linking the timing information of the MS-State transcriptions with the dialogue act annotations of SWBD-DAMSL. The NXT corpus includes a reduced set of 642 of the original conversations included in SWBD-DAMSL.

In addition to the SWBD corpus, we also use the HUB5 evaluation set (LDC, 2002) which consists of a set of 20 telephone conversations between speakers who were not part of the original SWBD data collection effort. This HUB5 set is commonly used for evaluating ASR systems trained on SWBD since it allows the training and test sets to have no overlap in speaker identity.

Figure 2.8: Recording setup and camera configurations for the MAHNOB Mimicry Database. Figure from Bilakhia et al. (2015)

Mahnob Mimicry Database The MAHNOB Mimicry Database (MMD) (Bilakhia et al., 2015) is an audio-visual corpus of dyadic social interactions where participants either discuss a sociopolitical topic or participate in a tenancy-agreement role-playing scenario. The corpus consists of 54 dyadic conversations between 70 participants totalling 11 hours in length. The participants were recruited from the staff and students at Imperial College London. Although the conversations were conducted in English, the participants were from a wide array of different national backgrounds, which included Spanish, French, Greek, English, Portuguese, and Romanian. Accordingly, a variety of different accents and dialects are present in the corpus. The participants sat in the same room with arrays of cameras configured to capture facial behaviours and gestures (shown in Fig. 2.8). The visual recordings were made using seven cameras for each of the participants and one overview camera that captured both participants at the same time (15 cameras in total). The video was captured at a 58 Hz sampling rate with 1024 by 1024 resolution. The audio recordings were made using separate head-mounted microphones at a 48 kHz sampling rate. Although the quality of the recordings is high, there is often noticeable cross-talk between the microphones. In our experiments in Chapter 4 we investigate the prediction of turn-taking behaviours using audio and visual features. To do this we needed speech activity labels for the MMD corpus. Since there are no speech transcriptions available for the dataset and automatic VAD predictions were unreliable (due to the cross-talk) we manually labelled the dataset for speech activity. The manual speech activity annotation process consisted of first using an automatic approach based on intensity thresholding and VAD predictions. The automatic predictions were then manually revised by an annotator to account for false and missed detections.

2.5 Conclusion

In this chapter existing research relevant to the development of turn-taking models was presented. The turn-taking dynamics of human-human conversations were discussed in Section 2.1. This included an overview of CA theory regarding turn-taking, as well as a description of the different types of turn-taking cues that have been identified in the literature. In Section 2.2 a description of the different types of dialogue systems, and their components, was presented. This included a discussion of incremental systems, which are the primary use-case for the turn-taking models presented in this thesis. In Section 2.3 an overview of the literature on different types of turn-taking models, as well as the different types of turn-taking decisions, was presented. Finally in Section 2.4 the datasets that are used in this thesis are described.

Chapter 3

Features for Continuous Turn-Taking Modelling

3.1 Motivation and Related Work

From the literature surveyed in Section 2.3, it can be observed that sequential neural models such as CTT have good capabilities for modelling acoustic and prosodic turn-taking cues. There is also the added appeal that these models can capture generalized information about the turn-taking behaviours in the data. This generalized information can be used for multiple different types of decisions, not just traditional endpointing decisions. However, sequential modelling raises issues about the treatment of input features. For instance, in the reviewed literature, many of the traditional non-sequential models (e.g. Raux and Eskenazi (2012); Gravano and Hirschberg (2011); Meena et al. (2014)) found that using linguistic features such as syntactic, lexical, and semantic features were more useful than prosodic features. This precedence of linguistic over prosodic features is not observed in the literature on neural sequential models, where prosodic features tend to contribute comparatively more to the overall performance.

There is the possibility that this discrepancy is due to the choice of features that have been used in sequential neural models, as well as how they have been represented. In the original paper on CTT modelling, Skantze (2017b) used sequences of POS tags (represented as one-hot vectors) to improve the performance of the prosody-only model. If we consider the types of turn-taking cues that can be used by the model when we include POS input features, these cues can be grouped into three categories. First, there are cues that can be inferred from the

POS uni-grams, whereby some POS tags are more or less likely to be followed by an EOT (e.g. EOTs are unlikely to happen after determiners or conjunctions). Second there are cues that can be derived from sequences of POS tags (as opposed to just the uni-grams). These cues stem from the observation that, when modelled sequentially, POS features can approximate phrase structure, as discussed in Section 2.1.2. By modelling phrase structure it may be possible for the network to infer points of syntactic completion. Third, there are also interactions between syntax and prosody (see the discussion in Section 2.1.2).

While Skantze (2017b) presents evidence that POS features improve the performance of a prosody-only model, the issue of whether other linguistic features improve the performance is not investigated. Lexical features are able to capture information regarding what type of dialogue act is being spoken. For example, "okay" or "yeah" have been associated with turn-yielding in the data analysed by Gravano and Hirschberg (2011), while fillers such as "uhh" or "um" are associated with turn-holding. It is also worth noting that if enough examples of lexical patterns are used to train a model, it is possible that POS features will become redundant. It is therefore worthwhile investigating representations of lexical items as input features for CTT prediction.

Another issue that is not addressed by Skantze (2017b) is the question of which acoustic features are most suitable for CTT modelling. In much of the literature on prosodic features in non-sequential models, the prosodic information is represented as summary statistics over the final 200 to 500 ms of an IPU (e.g. Gravano and Hirschberg (2011); Meena et al. (2014); Razavi et al. (2019)). In comparison, acoustic features in CTT models have the potential to model more complex information. They can theoretically capture much longer dependencies if needed, as well as more detailed prosodic patterns that cannot easily be summarized in statistics such as mean and slope. They can also potentially capture information about the dialogue structure since they can capture how long a speech segment (such as an IPU) has lasted for. It may even be possible for the acoustic features to be used by the model to detect short linguistic structures such as backchannels without having to wait for ASR. This change in the functionality of acoustic and prosodic features within the model makes it useful to look at whether traditional prosodic features are still optimal for CTT.

The change of functionality of the features also poses the question as to which features are useful for which types of decisions. Since CTT is able to capture general turn-taking information that can be applied to make a variety of different types of decisions, it is desirable to know the features that are most useful for a given task. Skantze (2017b) proposed two different types

of decisions (endpointing and LONG/SHORT prediction at onset). It is desirable to know how feature choices affect the performance of these decisions and others.

In this chapter we significantly extend the original CTT models proposed by Skantze (2017b). We perform an investigation of four speech feature categories: acoustic, phonetic, syntactic, and lexical. We look at the interactions between the different features and propose optimal subset choices for different tasks. We use sequential forward selection (SFS) to investigate which of the acoustic features would be most useful for practical implementation. This chapter addresses the additional challenge of making turn-taking decisions at points of overlapping speech. Since these models can be applied to a wide variety of turn-taking prediction tasks, this investigation is also of relevance to the study of which modalities are important for different types of turn-taking decisions. We propose modifications to the loss function that improve performance. We look at the effect that the role of the dialogue-task participant has on the model prediction performance. Additionally, we look at the impact of speaker familiarity on model prediction performance. Our primary aim in this investigation though is to inform the feature choices in future designs of turn-taking models for SDSs.

3.2 Model Details

The CTT model we use in this chapter is a re-implementation of the one proposed in Skantze (2017b) (described in Section 2.3.3) with some modifications. Skantze (2017b) used truncated back-propagation through time (TBPTT) which involved using long training sequences of $T = 1200$ and performing back-propagation every 200 timesteps in order to reduce the memory requirements. An issue with this approach is that if the data is segmented such that there is a decision to be made (e.g. endpointing) early in the sequence (say at the 2nd timestep of the training sequence), the hidden state of the LSTM will only have 100 ms worth of feature information to make this decision. As an alternative we use sequences of length 800 (40 seconds) *without* truncation, during training while also randomizing the batching. During testing, rather than randomizing the batching, we process all of the conversations in a batch sequentially, preserving the LSTM hidden state between successive batches and only resetting it at the start of a new conversation. This allows the LSTM to maintain contextual information across batches.

Secondly, while originally in Skantze (2017b) mean absolute error (MAE) was proposed as an objective function, in our implementation the networks were trained to minimize a binary cross-entropy (BCE) loss. Using the notation introduced in Section 2.3.3, the BCE for a CTT

Frequency	pitch; jitter; centre frequencies of formants 1, 2, and 3; bandwidth of first formant
Energy	loudness; shimmer; harmonics-to-noise ratio (HNR)
Spectral	MFCCs 1-4; spectral flux; alpha ratio; Hammarberg Index; spectral slope 0-500 Hz and 500-1500 Hz; relative energy of formants 1, 2, and 3;

Table 3.1: Acoustic features from the eGeMAPs feature set (Eyben et al., 2016) that are used in our experiments.

sequence is given as:

$$L_{\text{BCE}} = \frac{-1}{NT} \sum_{t}^{t+T} \sum_{n=1}^{N} \boldsymbol{Y}_{t,n} \log(\hat{\boldsymbol{Y}}_{t,n}) + (1 - \boldsymbol{Y}_{t,n}) \log(1 - \hat{\boldsymbol{Y}}_{t,n}) \tag{3.1}$$

We found that using the BCE loss significantly improved the predictive performance of the networks due to BCE directly measuring the difference between class distributions rather than an absolute error distance. The performance increases are validated below in Section 3.6.3. The model was implemented in Python using the PyTorch framework and our code is available online[1].

3.3 Features for Turn-Taking

3.3.1 Acoustic Features

As acoustic features, we use low-level descriptors from the eGeMAPs (Eyben et al., 2016) feature set extracted with the OpenSmile toolkit (Eyben et al., 2010). The 21 features in the set include energy (e.g. loudness, shimmer), frequency (e.g. pitch, jitter), and spectral (e.g. spectral flux, MFCCs) features. The full list of the features and their classifications are given in Table 3.1. All the frame-steps and frame-sizes for the sample windows were changed from their default settings to accommodate the 50ms timestep and ensure that there was no overlap of samples between adjacent windows. All other settings were kept the same. The features were then normalized using z-scores on a per-file basis.

3.3.2 Linguistic Features

We investigate two types of linguistic features: Part-of-Speech (POS), and word embeddings. POS has been found to be a good predictor of turn-switches in the literature (Gravano and

[1] www.github.com/mattroddy/lstm_turn_taking_prediction

Hirschberg, 2011). As a comparison, we also use the word transcriptions supplied with the dataset. The comparison is relevant because automatic systems for POS tagging would need the words (from an ASR system) to extract these features. We question whether this extra processing step is indeed necessary, or whether raw word features without the POS tags will suffice. This would be a particular advantage to real-time systems since the added computation of POS tags from the words would be avoided.

For both the words and POS tags, we used the annotations supplied with the data. The number of POS tags was 59, and the number of word tags was 2501. The raw data was represented as an enumerated vocabulary with an added zeroth element representing no change in state. In an effort to simulate the conditions of a real-time system, the linguistic features were not provided to the system until 100 ms after the end of the word. In both cases, the raw features are transformed into separate embeddings of length 64 using added linear network layers that are jointly trained with the rest of the network. The embeddings allow the network to learn representations of the features that are specific for the turn-taking prediction task.

3.3.3 Phonetic Features

For phonetic features we use the bottleneck layer output of a DNN trained to classify senones (tied tri-phone states). Masumura et al. (2017) found that senone bottleneck features (BNFs) outperformed MFCCs in an endpointing system based on stacked RNNs. In our implementation, the DNN takes stacked inputs of a central frame with 5 context frames on either side of the central target frame. For each frame 12 MFCCs and their first and second order derivatives are calculated, leading to an input vector of size 396 per target label. To train the DNN we use the senone posterior outputs from a GMM-HMM speech recognition system. The HMM system was trained using a Kaldi recipe on 100 hours of the LibriSpeech corpus (Panayotov et al., 2015). We used a DNN with five hidden layers with a bottleneck of size 64 in the fourth layer. The other four hidden layers had 512 hidden nodes and used tanh activation functions. The output layer used a sigmoid activation over the 3448 clustered senone states. Since our ASR system uses frame steps of 10 ms and our prediction system uses 50ms frame steps, we use element-wise averaging within each 50ms window. We also delay all of the BNF features by 60ms to compensate for the context that is used in their calculation.

3.3.4 Voice Activity

The speech transcriptions included with our data were used as a ground-truth for the 60 speech activity predictions. They were also used as a voice activity (VA) feature.

3.4 Prediction tasks

We test the performance of our networks using three turn-taking prediction tasks that are pertinent to SDSs. The first two (prediction at pauses, prediction at onset) were proposed by Skantze (2017b) and described in described in Section 2.3.3. The third, prediction at overlap, is introduced in this work. We test predictions at pauses of both 500 ms (PAUSE 500) and 50 ms (PAUSE 50). 500 ms has been identified in the literature as being a common pause threshold in many commercial speech technologies (Heldner and Edlund, 2010). 50 ms can be considered a threshold that more closely models natural conversational speech since it can potentially generate decisions within the 200ms modal response threshold. The number of samples in the test set for PAUSE 50 is 6533 (4203 HOLDs, 2330 SHIFTs). The number of samples in the test set for PAUSE 500 is 2467 (1496 HOLDs, 971 SHIFTs). The majority baseline F-score (always HOLD) is 0.5037 and 0.4579 for PAUSE 50 and PAUSE 500 respectively. For the prediction at onset task (ONSET), The number of samples in the test set is 476 (238 SHORTs, 238 LONGs). The majority baseline F-score (always SHORT) is 0.3333.

Prediction at Overlap

Prediction at overlap (OVERLAP), shown in Fig. 3.1, is introduced in this chapter as a decision to specifically model points where the dialogue system is holding the floor and a user begins speaking while the system is speaking. It makes decisions as to whether it is appropriate for the system to stop speaking or to continue. The decision to continue would be in the case where the overlapping utterance can be considered to be similar to a backchannel. To make this HOLD/SHIFT prediction, decision points are identified were there has been at least 1.5 seconds of speech by a participant that includes a period of 100ms of overlapping speech in the last two frames. From this decision point, 10 frames between 400 and 900 ms in the future are selected as the decision window. If there is only one person speaking we label this decision as either HOLD or SHIFT. To make predictions on this task the means of the output probabilities from the two participant's networks are computed for the 10 frames within the decision window. The

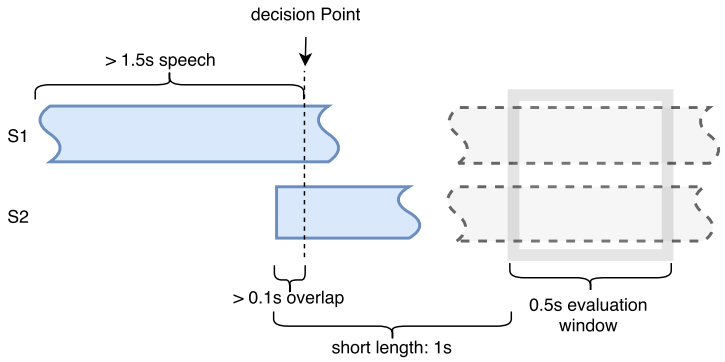


Figure 3.1: Decisions at overlap.

two means are then compared to produce a HOLD/SHIFT prediction. The number of samples in the test set is 313 (143 HOLDs, 170 SHIFTs). The majority baseline F-score (always SHIFT) is 0.3823.

3.5 Experimental Setup

We use the HCRC map task corpus (described in Section 2.4) for our experiments (Anderson et al., 1991). We used 96 conversations as training and 32 conversations for testing. Additionally, we used 32 conversations of the training set as a held-out test set during hyperparameter searches. In selecting the splits between the sets, our main consideration was maintaining speaker independence between the sets. We balanced variables such as gender, and whether the participants could see each other or not.

We trained models using different combinations of the feature groupings and compared the results. To find optimal hyperparameters for each experimental result, a grid search procedure was used. Networks with hidden node sizes of [20, 40, 60, 80, 100], learning rates of [0.001, 0.01], and L2 regularization values of [0.001, 0.0001] were trained and tested on the held-out portions of the training set. The networks were trained using the ADAM optimizer (Kingma and Adam, 2014), with a batch size of 128 and early stopping. Since performance fluctuated between test runs (due to factors such as random weight initialization and randomized batching), once the optimal learning rate and regularization values for a feature set were selected, each hidden node size was trained and tested three times. Once the optimal number of hidden nodes was selected, the network was then trained and tested 10 times on the full train/test split. The averaged F-scores (f_1) and losses are shown in Tables 3.2 and 3.3. The three best results for

	BCE loss	f1 50ms	f1 500ms	f1 over	f1 short long
Acous	**0.5506**	0.7685	0.8021	**0.6955**	**0.7820**
Phon	0.5645	0.7501	0.7968	0.6670	0.7391
Pos	0.6265	0.6454	0.6280	0.5097	0.6015
Words	0.5842	0.7110	0.7293	0.5965	0.7089
Ling	0.5823	0.7165	0.7442	0.6185	0.7125
Acous Phon	0.5575	0.7644	0.7924	0.6836	0.7724
Acous Words	***0.5499***	***0.7811***	**0.8126**	0.6876	0.7698
Acous POS	0.5507	0.7664	0.8031	**0.6901**	***0.7841***
Acous Ling	**0.5502**	**0.7793**	0.8107	***0.6983***	0.7650
Phon Ling	0.5589	0.7702	**0.8128**	0.6562	0.7505
Acous Phon Ling	0.5508	**0.7788**	***0.8173***	0.6880	**0.7768**

Table 3.2: Performance of feature types (voice activity excluded)

each metric are shown in bold with the best one in italics. 'Ling' corresponds to the use of both POS and word features. We report the performance of voice activity separately due to its potential masking of the performance of other features. In our discussion we use independent two-tailed t-tests to report on the statistical significance of the difference between the means of metrics.

In addition to testing different feature groupings, we run an experiment using the widely used sequential forward selection (SFS) algorithm (Kudo and Sklansky, 2000) on the 21 acoustic features to investigate which, and how many, of these features are most useful to turn-taking prediction. During the selection process, at each step of the algorithm we use similar grid-search and testing procedures to those described above. The BCE losses of the first 10 choices are shown in Fig. 3.2.

3.6 Discussion

3.6.1 Feature Set Comparison

A conclusion we can draw from the results in Tables 3.2 and 3.3 is that acoustic features are a good first choice for training a system that can perform well on most of our selected turn-taking metrics. All of the best performing feature sets for the studied metrics include acoustic features, with only one exception (the VA Phon Ling combination on PAUSE 500). The added inclusion of linguistic features should be done with consideration for the overall goal of the system and the processing involved in ASR. POS features have been found in previous studies as good turn-taking features (Gravano and Hirschberg, 2011; Koiso et al., 1998). We find that including

	BCE loss	f1 50ms	f1 500ms	f1 overlap	f1 short long
VA	0.5761	0.6800	0.7066	0.5548	0.6281
VA Acous	**0.5454**	0.7926	0.8089	***0.7106***	**0.7767**
VA Phon	0.5559	0.7765	0.8029	0.6751	0.7426
VA POS	0.5656	0.7374	0.7344	0.6210	0.6939
VA Words	0.5572	0.7708	0.7701	0.6545	0.7216
VA Ling	0.5573	0.7693	0.7705	0.6506	0.7169
VA Acous Phon	0.5513	0.7824	0.8034	**0.7035**	0.7704
VA Acous Words	***0.5449***	***0.7959***	**0.8153**	0.6912	0.7721
VA Acous POS	**0.5456**	0.7909	0.8055	**0.7055**	***0.7823***
VA Acous Ling	0.5461	**0.7945**	0.8127	0.6826	0.7674
VA Phon Ling	0.5505	0.7903	***0.8209***	0.6825	0.7538
VA Acous Phon Ling	0.5468	**0.7951**	**0.8189**	0.6943	**0.7722**

Table 3.3: Performance of feature types (voice activity included)

word features in addition to acoustic features improves the performance of the networks on the standard turn-taking decisions PAUSE 50 and PAUSE 500 more than POS features ($p < 0.05$ in all cases). However, POS features are more useful for the ONSET decisions ($p < 0.05$), with the best performance in both tables being achieved with POS tags. So if we intend for the SDS to be able to discern whether an utterance will be short like a backchannel, the results indicate that it would be useful to include POS tags. Concerning phonetic features, the results indicate that their inclusion aids the performance of predictions on the PAUSE 500 metric when used in conjunction with other features. They are less useful at the PAUSE 50 prediction.

The results imply that good performance on our OVERLAP metric most strongly relies on acoustic features. While the best score on this metric in Table 3.2 is achieved with the combination of acoustic and linguistic features, this mean is not significantly different from the mean of the score for acoustic features alone ($p = 0.49$). We therefore suggest that the network is mainly relying on acoustic information in both cases.

In comparing the results of the two tables, it is perhaps unsurprising that all of the best results on the different metrics are achieved with the inclusion of VA features. The fact that the network can achieve reasonably good performance with just the VA features suggests that it is learning general information about turn distributions within the data. Similar conclusions were drawn by Raux and Eskenazi (2012) where they found that turn lengths and number of pauses so far in the turn were good predictors of turn-taking behaviour.

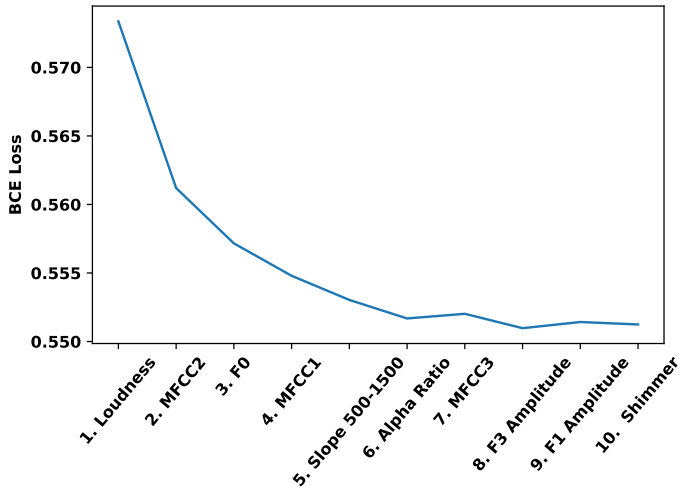


Figure 3.2: Sequential feature choices with the loss for each consecutively chosen feature.

3.6.2 Sequential Forward Selection

The SFS experiment gives us further insight into which of the acoustic features are the most useful for turn-taking modelling. Fig. 3.2 indicates that after the inclusion of the 8th feature, the loss stops to decrease. This suggests that the most important acoustic features from this set of 21 features are loudness, F0, low order MFCCs, and spectral slope features. F0 and loudness are both traditional features that are established in the literature as indicators of turn-taking behaviour (e.g (Duncan, 1972; Gravano and Hirschberg, 2011)). MFCCs have also been shown to be good features for the classification of short listener responses (Neiberg and Truong, 2011) while spectral slope features can be considered to contain similar information to these low-order MFCCs. The results indicate that these traditional acoustic features are good choices for continuous turn-taking prediction.

3.6.3 Baseline Performance Improvements

We achieved better performance than (Skantze, 2017b) on all metrics, apart from the prediction at onset metric. Examination showed that their train/test split was slightly different, and did not use fully speaker-independent sets. When we ran our experiments using their original split, we achieved the best performance on this metric, along with the others but recommend the speaker independent splits as more realistic. To validate our decision to change the loss function from MAE to BCE, we performed a comparison of the two using networks trained on acoustic and

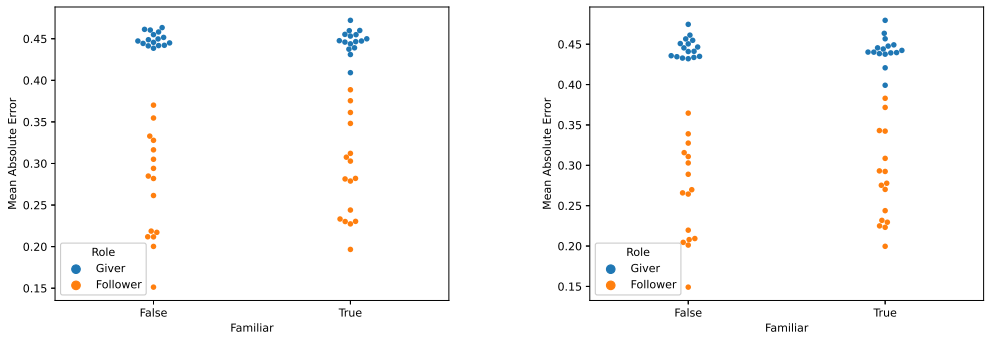
word features. All settings were identical except for the loss functions. For PAUSE 50 the mean F-score for the MAE networks was 0.748, while it was 0.7811 for the BCE networks ($p \ll 0.001$). In all other F-score metrics we observed similar statistically significant improvements.

3.6.4 Effect of Role and Familiarity

In this sub-section we investigate the effects of two factors on the predictive performance of the trained networks: (1) the effects of conversational role and (2) speaker familiarity. In the turn-taking paper of Sacks et al. (1974), the authors suggest that the organization of turn-taking could be affected by both the identities of interlocutors, as well as their familiarity. The effect of conversational role was studied by Bull and Aylett (1998) where the authors examined the timings of turn-switch offsets in the HCRC map task corpus. They found that transitions from giver to follower had a longer average offset times than transitions from follower to giver. The authors suggest that this was due to the silences that occur while the follower is carrying out the task of drawing on the map.

When interacting with an unfamiliar dialogue partner, it is more likely that we will be less accustomed to their dialogue habits and idiosyncrasies (e.g. accent, mannerisms, colloquialisms) than if we were interacting with a partner we knew well. Due to increased language comprehension latencies (Magyari et al., 2014), this lack of familiarity might manifest itself in the presence of fewer overlaps and longer turn-offsets. Ward et al. (2018) suggest that the differences between CTT model performance across datasets could be explained by the varying levels of familiarity. However, Shriberg et al. (2001) performed analyses of overlap in large-scale telephone corpora and found that there was no difference in the amount of overlap in corpora that featured familiar dialogue pairs (the CALLHOME corpus (Canavan et al., 1997)) and unfamiliar dialogue pairs (SWBD).

The HCRC map task corpus is designed in such a way that half of the test set conversations were between conversants who were familiar with one another, and the other half were unfamiliar with one another. In Fig. 3.3 we show plots of the average frame-based performance (MAE) for the 32 test set conversations. We show the plots produced by a network trained only with acoustic features on the left, and plots produced by a network trained on both acoustic and lexical features on the right. T-tests comparing MAE under the familiar and unfamiliar conditions did not show any significant effects ($p = 0.78$ in the acoustic-only condition, and $p = 0.76$ in the acoustic-plus-lexical condition).



(a) Only acoustic features

(b) Acoustic plus lexical features

Figure 3.3: Plots of the mean absolute error (MAE) of different target speakers averaged across full conversations. The x-axes correspond to whether the interlocutors were familiar with one another. The role of the target speaker (whether they were a giver or a follower) is denoted by colour.

Regarding the effect of conversational role on MAE, we can see that the CTT networks have lower MAE for followers than givers ($p \ll 0.001$ in both only-acoustic and acoustic-plus-lexical conditions). One possible explanation for why there is higher MAE for the giver role is rooted in the lopsided nature of the task: the givers will typically be speaking much more than the followers. It is possible that the network finds it harder to make predictions for speakers that are continuously speaking than for followers who only speak rarely. When we consider that the MAE value is calculated according to Eq. 2.1, which is based on predictions up to three seconds into the future, it is likely a difficult task to predict where up-coming MTPs by the current speaker (many of which will be short) will occur. It is likely much easier to make predictions for the follower, who will generally be silent for a much larger proportion of the interaction than the giver. However, this proposed explanation is only speculative and would require further investigation to confirm the hypothesis.

3.7 Conclusion

In this chapter we presented an investigation of features for CTT and made improvements to the model proposed by Skantze (2017b). We investigated using lexical embeddings and phonetic features that had not yet been applied to CTT. The results showed that for the overall BCE test loss, lexical embeddings always outperformed syntactic features. Also, when lexical and syntactic features were used together with acoustic features, the syntactic features did not improve

the overall performance compared to just acoustic and lexical features. This supports the idea mentioned in Section 3.1 that lexical features can potentially make POS features redundant. However, in contrast, it can also be observed that our best performance on the ONSET predictions is achieved using POS features combined with VA and acoustic features. This supports the idea that the specific choice of which type of linguistic feature to use (e.g. words or POS) should be considered within the context of the demands of the SDS itself, as the utility of features differ based on what type of turn-taking prediction is being made.

The literature on turn-taking models reports that generally linguistic features are better predictors of turn-taking behaviours than prosodic or acoustic features (e.g. Gravano and Hirschberg (2011); Raux and Eskenazi (2012); Meena et al. (2014)). CTT models show the opposite result, that acoustic features improve performance much more than linguistic features. This conflict poses several questions: Is there a deficiency in the way that continuous models represent linguistic information that causes them to perform worse? Is the lexical, semantic, and syntactic information that was useful in non-sequential models still present in the continuous models? Or are sequential continuous models simply better at modelling acoustic and prosodic turn-taking information, to the point that acoustic features become more useful than linguistic features? There is also the possibility that the acoustic features could be learning information from the data about the words themselves. In the next chapter we propose a modified approach to how modality-specific information is temporally represented that attempts to address some of these issues.

Chapter 4

Temporal Considerations for Continuous Turn-Taking

4.1 Motivation and Related Work

In the previous chapter we presented an analysis of different features and how they contributed to the performance of CTT models. In our experiments, acoustic features generally contributed the most to the overall performance as measured in the BCE loss as well as our other metrics. The most useful features tended to be traditional acoustic features (e.g. intensity, F0, and lower order MFCCs) that are associated with the detection of prosodic and acoustic turn-taking cues. While linguistic features were useful, they did not contribute as much to the decisions as the acoustic features. This contrasts with non-sequential models where syntactic, lexical, and semantic turn-taking features generally outperform prosodic and acoustic features (e.g. Gravano and Hirschberg (2011), Raux and Eskenazi (2012), Meena et al. (2014), and Razavi et al. (2019)). This raises a number of questions that were introduced at the end of the last chapter concerning how CTT models handle turn-taking cues from different modalities. Primary among these concerns is the question of how raw features should be processed such that the model can learn representations that allow it to detect these cues.

To attempt to address these issues we first make the observation that human spoken language during a speaking turn can be conceptualized as having a hierarchical structure: phonemes are used to create words, words are used to create IPUs, IPUs are used to create turns. Each level of this hierarchy operates at a different timescale, with lower level abstractions operating at a faster

temporal granularity than higher levels. The turn-taking cues described in Section 2.1.2 can be associated with different levels of this hierarchy: prosody and acoustics can be associated with operating at a fast phoneme-level, lexical and syntactic cues can be associated with operating at a slower word-level, while semantic cues can be considered to operate at an even slower IPU-level or turn-level.

This hierarchical structure of spoken language presents difficulties for CTT modelling. In the original formulation of CTT (as well as the one used in the last chapter) all of the features are input to the model at a 50 ms frame rate. In a study of speech tempo in Dutch Quené (2007) reports an average syllable duration of 239 ms, whereas words occurred at a rate of 159 words-per-minute (WPM), or one every 2.7 seconds. While the temporal rate of syllables and words varies depending on many factors, we propose that it is likely that the choice of temporal granularity used for CTT will affect how well certain cues can be modelled. For example, if we wish to capture prosodic cues which occur during syllables (such as pitch patterns), it is reasonable to assume that modelling pitch over a long, word-rate temporal granularity will not be as effective as modelling at the standard 50 ms rate. Doing this would smooth over the finer-grained prosodic inflections that are important to forming turn-taking predictions.

While the 50 ms rate may be more suited for prosodic features, we propose that there are limitations to modelling linguistic features at this rate. The linguistic representation used in the last chapter involved training word embeddings that were triggered for one frame, 100 ms after the word occurred. A separate word embedding which represents a lack of change was used for the rest of the frames. This creates a representation with long gaps between embeddings that represent words. If we use the 159 WPM average rate from Quené (2007), during utterances the average rate of a word embedding for a speaker should be one every 54 frames. We propose that, due to the vanishing gradient problem (Hochreiter et al., 2001), this makes it difficult for the LSTM to model the long-term dependencies that exist in natural language. In traditional non-sequential models this issue is less apparent since sequential linguistic features are often represented as summaries e.g. filled slots (Raux and Eskenazi, 2012), or N-grams (Gravano and Hirschberg, 2011).

We propose a way to address this problem by using a multiscale RNN architecture, in which modalities are modelled in separate sub-network LSTMs that are allowed to operate at their own independent timescales. The intention is to aid the network in modelling turn-taking cues at different levels of the hierarchy. A master LSTM is used fuse the modalities and form predictions

at a regular rate by taking as input a concatenation of the current hidden states of the sub-networks. This allows the hidden states of the sub-networks to be updated at an appropriate temporal rate. Multiscale RNNs were proposed by Schmidhuber (1992) and Hihi and Bengio (1996) as a way of hierarchically modelling temporal representations. Hihi and Bengio (1996) proposed using multiscale RNNs as a way of learning long-term dependencies.

The independent modelling of modalities at different timescales also presents an opportunity for modelling features that are not directly part of linguistic hierarchy described above. As discussed in Section 2.1.2, visual cues such as eye gaze, gestures, and facial expressions are important for turn-taking regulation in face-to-face interactions. These cues are distinct from acoustic and linguistic cues in that they can occur at any point in the conversation independently from speech. For example, a listener may generate visual feedback cues that signal to a speaker to continue (e.g. nods) without speaking. Video information is also typically processed at different timescales from acoustic features, with frame rates between 30 and 60 frames-per-second being common. While it would be possible to sample visual features at the frame rate of the acoustic features (or vice versa), the use of independent sub-network LSTMs means that information sources are not forced to adhere to one uniform temporal rate.

In this chapter we present significant extensions to the work presented in the last chapter and the original CTT model proposed by Skantze (2017b). We investigate the performance of our proposed multiscale architecture on two datasets that contain two different combinations of modalities. We look at the influence of modelling modalities in separate sub-networks and using separate timescales. We find that there are significant performance benefits to modelling linguistic features at a slower temporal rate, and in a separate sub-network from acoustic features. We also find that our approach can be used to incorporate gaze features into turn-taking models, a task that has been previously found to be difficult (De Kok and Heylen, 2009).

4.2 Multiscale Continuous Turn-taking Prediction

To address the problems discussed above, we modify the original CTT architecture to include a variable number of sub-network LSTM cells that process features from separate modalities independently. The sub-networks are allowed to process the input features from the separate modalities at different timescales. An example network configuration that uses a master LSTM ($\boldsymbol{h}^0$) and two sub-network LSTMs ($\boldsymbol{h}^1$,$\boldsymbol{h}^2$, each assigned a modality) is shown in Fig. 4.1. We use superscripts to denote the index of modalities $m \in M$, and subscripts to index timesteps

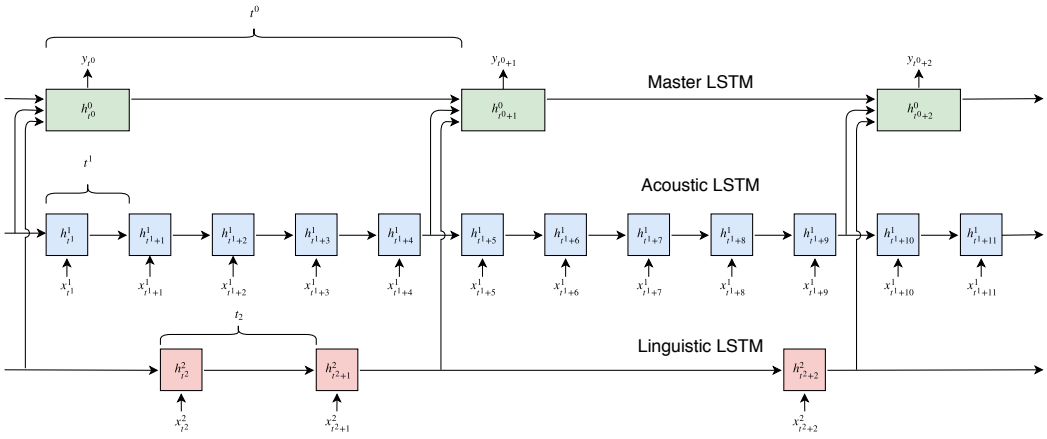


Figure 4.1: Multiscale RNN architecture

(represented using the notation t^m). At each timestep of the master LSTM (t^0), the current states of the sub-network LSTMs are concatenated and fed into the master LSTM. The hidden state update process for the network is shown in Algorithm 1. By feeding the current states of the sub-networks into the master LSTM, we are effectively performing a sampling operation, represented in the algorithm by the step $\boldsymbol{h}^m_{t^0+1} \longleftarrow \boldsymbol{h}^m_{t^m+1}$. The sampling operation can either increase or decrease the temporal resolution of the individual modalities, depending on the timescales used. For example, in Fig. 4.1, the temporal resolution of the first sub-network (h^1) is decreased since we sample it at a regular rate every five t^1 timesteps. The temporal resolution of the second modality could either be increased or decreased by the sampling process since the features have irregular timesteps. The processing of features at a slower update rate potentially allows the network to better retain information. The model was implemented using the PyTorch framework and our code is available online[1].

Input: h_t, x_{t+1}
Output: y_{t^0+1}
for $m \in M$ **do**
  for $t^m : t^0 \leq t^m \leq t^0 + 1$ **do**
    $h^m_{t^m+1} \longleftarrow LSTM(x^m_{t^m+1}, h^m_{t^m}; \Theta^m)$
  end
  $h^m_{t^0+1} \longleftarrow h^m_{t^m+1}$
end
$h^0_{t^0+1} \longleftarrow LSTM([h^m_{t^0+1}, ..., h^M_{t^0+1}]^T, h^0_{t^0}; \Theta^0)$;
$y_{t^0+1} \longleftarrow \sigma(h^0_{t^0+1}; \Theta^\sigma)$

Algorithm 1: Multiscale continuous turn-taking prediction

[1] www.github.com/mattroddy/lstm_turn_taking_prediction

4.3 Experimental Design

To assess the performance of our multiscale approach, we test it on two different datasets. In each dataset, features from two separate modalities are investigated by training models using a variety of different network configurations. The HCRC map-task corpus (MTC) Anderson et al. (1991) is used to examine linguistic and acoustic modalities while the Mahnob Mimicry Database (MMD) (Bilakhia et al., 2015) is used to examine visual and acoustic modalities. In this section we discuss the details of the datasets and how features were extracted.

4.3.1 Map-Task corpus

For the MTC experiments we use the same training/validation/testing splits that were used in Chapter 3. We use the same same set of eGeMAPs (Eyben et al., 2016) acoustic features described in Section 3.3.1. However we extract the features at two different temporal resolutions: 10 ms and 50 ms. We use these two different temporal resolutions to investigate which one is more useful for our turn-taking models. In our results tables and discussion we refer to these as "Acous 10ms" and "Acous 50ms". For the linguistic features we use the 64-length word-embeddings described in section 3.3.2. However three different temporal rates for the processing of linguistic features are tested in our experiments. In our discussion and results below, "Ling 50ms" refers to using word features that have been sampled at regular 50ms intervals. "Ling 10ms" refers to using word features that are sampled at a faster rate of 10ms. "Ling Asynch" refers to using an irregular update rate, where the LSTM only processes the linguistic features when a new word is available. In other words, in a real-world implementation, the hidden state of the linguistic LSTM is only updated when there are new results from the ASR.

4.3.2 Mahnob Mimicry Database

The MMD is an audio-visual corpus of 54 dyadic conversations which is described in detail in Section 2.4. The participants are either assigned discussion topics or roles to play in a loosely defined role-playing scenario. When splitting the data into training and test sets, we balanced the number of role-playing and discussion conditions in the training and test set. We used 39 conversations for training and 15 for testing. Since there are no speech transcriptions available for the dataset, we manually labelled the dataset for speech activity. The procedure we used for extracting acoustic features for the MMD was the same as that followed for the MTC in section

4.3.1.

Visual Features The Mahnob dataset has 7 different camera angles per participant. We use the angle that is labelled "Facenear2" for all of our extracted features. We automatically extract visual features using the OpenFace toolkit (Baltrušaitis et al., 2016). Figure 4.2 shows an example visualization of the types of gaze direction predictions that are generated by the tool. During informal exploratory experiments, we found that the automatically extracted gaze features performed better than other features extracted with the toolkit (e.g. facial action units, pose). We therefore used the gaze features along with a confidence score as our visual input feature. The gaze features consist of two three-dimensional vectors per participant, which represent the gaze direction predictions (x, y, z) for each eye. The two output gaze vectors produced by the OpenFace toolkit have unit length, and no post-processing was applied to them.

The confidence score output by OpenFace is an estimate of the probability that a face has been detected in a given video frame. There are frames in the video files when conversation participants tilt their heads at angles that make it difficult for OpenFace to detect a face. During these frames the gaze predictions are unreliable. The use of confidence scores as input features is motivated by the desire to allow the network to learn to rely less on the gaze features during frames with low confidence. When concatenated together the visual features are a 14-dimensional vector, 6 for the eye gaze predictions for each participant and then 1 confidence score per participant.

The video in the MMD uses a high frame rate of 58Hz. We perform a comparison of using features at this high frame rate and using features that are averaged over 50ms frame windows. In the results tables and the discussion below we refer to the high frame rate video and the averaged video features as "Visual 58Hz" and "Visual 50ms" respectively.

4.3.3 Experimental Procedure

We test the impact of using three different network configurations with multiple combinations of modalities at different temporal resolutions. The three network configurations are: "no subnets", which corresponds to an early fusion approach in which the modalities are fed directly into a single LSTM (shown in Figures 4.3 and 4.4); "one subnet", which corresponds to the use of only one sub-network LSTM; and two subnets, which corresponds to the use of separate LSTM sub-networks for the individual modalities (shown in Figures 4.1, 4.5, and 4.6). We note that

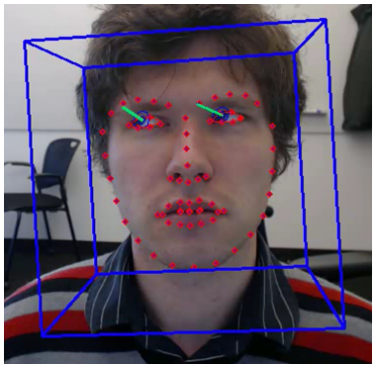


Figure 4.2: An example of visual features extracted with the OpenFace toolkit. Image from Baltrušaitis et al. (2016).

combinations such as "Ling 50ms" with "Acous 10ms" are not possible when using the "no subnets" and "one subnet" configurations since the features are being input into the same LSTM and cannot operate at different temporal resolutions. Apart from changing the architecture of the model described in Section 3.2, we reduced the sequence length to $T = 600$. We also included dropout (Srivastava et al., 2014) directly preceding the linear output layer.

Grid searches for three hyperparameters (hidden node size, dropout, and L2 regularization) were performed for each network configuration. In order to limit the influence of parameter count changes between the different network configurations, the hidden node count in a given network was limited to a sum of 150. Once the hyperparameters for a network are chosen, we train the network five times and report the mean values of the different evaluation metrics in Tables 4.1 and 4.2. The best performing modality combination for a given network configuration is shown in bold and the best overall performance is shown in italics. In our discussion below we use two-tailed t-tests to report on the difference between the means of metrics.

4.4 Discussion

Looking at the results from the fusion of linguistic and acoustic modalities shown in Table 4.1, it is clear that there are significant benefits in modelling acoustic and linguistic modalities separately using different timescales. Our best performance on all evaluation metrics is achieved using our multiscale approach where features from the two modalities are modelled at separate rates. Comparing the BCE loss of the best performing early-fusion result (7) with the best multiscale result (11) gives a statistically significant improvement ($P < .001$). Comparing the performance of the acoustic feature timescales, we observe that the faster rate of 10ms consistently performs

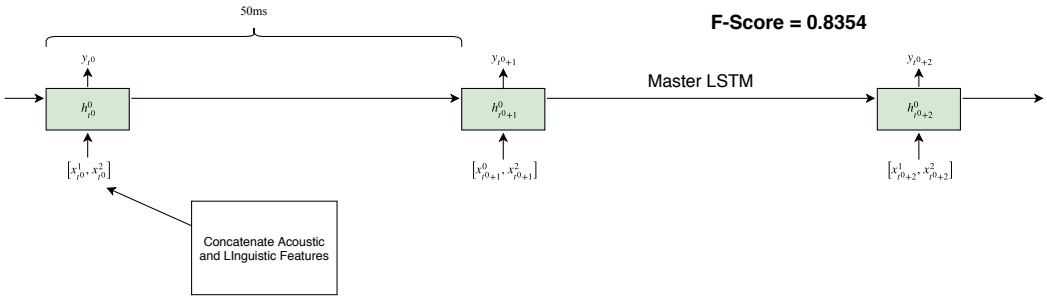


Figure 4.3: Network using the "no subnets" configuration, running at a 50 ms temporal granularity.

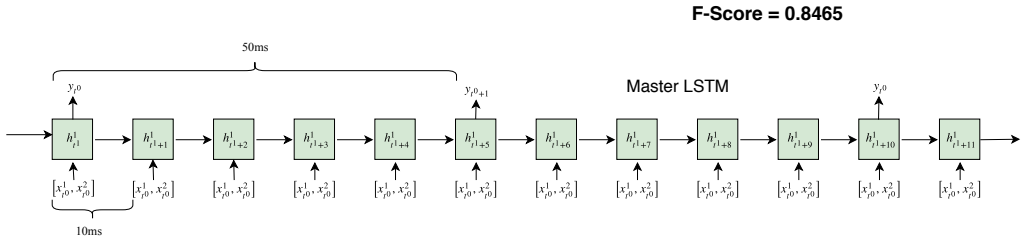


Figure 4.4: Network using the "no subnets" (early fusion) configuration, running at a 10 ms temporal granularity.

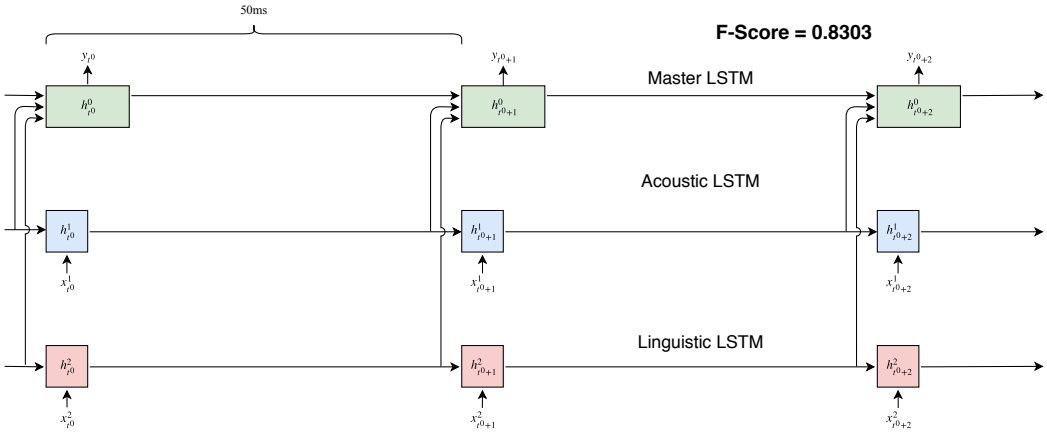


Figure 4.5: Network using the "two subnets" configuration with both modalities running at a 50 ms temporal granularity.

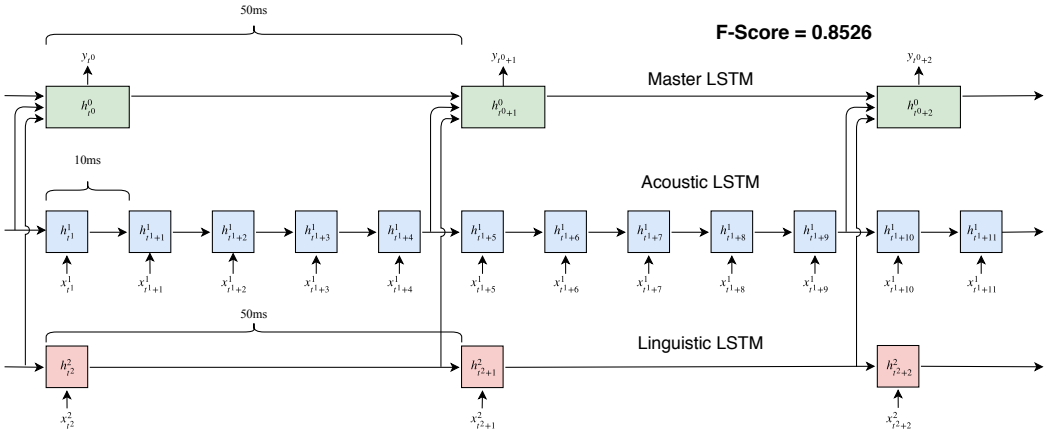


Figure 4.6: Network using the "two subnets" configuration with one modality running at 10 ms and the other running at 50 ms.

better than the slower 50ms rate. Looking at the performance of the three different linguistic timescales in (3,4,5), we see that processing linguistic features at the slower regular rate of 50ms achieves the best performance. Comparing the BCE loss of (3) and (5) suggests that sampling linguistic features at a fast temporal rate makes it difficult for the network to model longer term dependencies ($P = .004$). The effect of processing modalities on their own in separate subnetworks without the added gain of using separate timescales is inconclusive when we examine (6) and (10). Using a single subnet as an added layer also does not yield significant differences to the early fusion approach. We conclude that the main advantage in using our multiscale approach on a combination of acoustic and linguistic modalities is its ability to fuse the two modalities when the linguistic features are operating at a slow 50ms timescale and the acoustic features are operating at a fast 10ms timescale.

Looking at the results from the fusion of visual and acoustic modalities shown in in Table 4.2, we were able to achieve our best BCE loss using our multiscale approach to fuse acoustic features at a 10ms timescale and visual features at a 58Hz timescale. Comparing this result (9) with our best "no subnets" result (2) gives a statistically significant improvement ($P = 0.035$). We note that using early fusion with gaze features (5) does not add any value when compared to acoustic features on their own (1). The results also indicate that the faster 58Hz gaze features perform better than the averaged 50ms visual features when used in conjunction with the acoustic features. This suggests that we lose relevant information by averaging the gaze features within a timestep.

	BCE loss	f1 50ms	f1 500ms	f1 onset
No Subnets (Early Fusion)				
(1) Acous 50ms	0.5456	0.7907	0.8165	0.7926
(2) Acous 10ms	0.5351	0.8154	0.8428	0.8126
(3) Ling 50ms	0.5779	0.7234	0.7547	0.7249
(4) Ling Asynch	0.5839	0.7101	0.7341	0.7174
(5) Ling 10ms	0.5823	0.7072	0.7391	0.7111
(6) Acous 50ms Ling 50ms	0.5411	0.7957	0.8354	0.8101
(7) Acous 10ms Ling 10ms	**0.5321**	**0.8194**	**0.8465**	**0.8141**
One Subnet				
(8) Acous 50ms Ling 50ms	0.5414	0.7922	0.8366	0.8020
(9) Acous 10ms Ling 10ms	**0.5317**	**0.8237**	**0.8480**	**0.8128**
Two Subnets (Multiscale)				
(10) Acous 50ms Ling 50ms	0.5420	0.7916	0.8303	0.8019
(11) Acous 10ms Ling 50ms	***0.5291***	***0.8323***	0.8526	***0.8236***
(12) Acous 50ms Ling Asynch	0.5416	0.7949	0.8385	0.7993
(13) Acous 10ms Ling Asynch	0.5296	0.8307	***0.8553***	0.8232
(14) Acous 10ms Ling 10ms	0.5310	0.8285	0.8470	0.8189

Table 4.1: Map-task corpus experimental results

4.5 Conclusion

In this chapter we have shown that there are considerable benefits in using multiscale architectures for continuous turn-taking prediction. When fusing linguistic and acoustic modalities, the architecture allows acoustic features to be modelled at a fast temporal rate (which is better-suited to capturing prosodic inflections) while modelling linguistic features at a slower rate (which is better-suited to capturing long-term linguistic dependencies). When fusing visual and acoustic modalities, our multiscale approach allowed the use of high frame-rate visual features without resorting to averaging. Given that gaze features have been found to be difficult to integrate into turn-taking systems (De Kok and Heylen, 2009) our results present a promising direction for future investigation. Unfortunately, currently there is a lack of suitable large-scale audio-visual datasets that have high quality video, audio, and orthographic annotations. In the following chapters we therefore focus on the linguistic and acoustic modalities.

While overall we achieved better results using our multiscale approach when fusing linguistic and acoustic modalities, we should note that much of this improvement on the MTC can be attributed to being able to model the finer grained acoustic modality at the faster rate. Using the asynchronous linguistic features performed the best on the PAUSE 500 task (13) but the performance wasn't better than the combination of 10 ms acoustic features and 50 ms linguistic

	BCE loss	f1 50ms	f1 500ms	f1 onset
No Subnets (Early Fusion)				
(1) Acous 50ms	0.4433	0.8665	0.9230	0.8668
(2) Acous 10ms	**0.4348**	***0.8851***	**0.9343**	**0.8685**
(3) Visual 50ms	0.5840	0.7858	0.8154	0.6445
(4) Visual 58Hz	0.5941	0.7726	0.8031	0.6560
(5) Acous 50ms Visual 50ms	0.4497	0.8651	0.9159	0.8526
Two Subnets (Multiscale)				
(6) Acous 50ms Visual 50ms	0.4443	0.8637	0.9198	0.8711
(7) Acous 10ms Visual 50ms	0.4337	**0.8840**	***0.9347***	***0.8784***
(8) Acous 50ms Visual 58Hz	0.4437	0.8634	0.9216	0.8721
(9) Acous 10ms Visual 58Hz	***0.4332***	0.8831	0.9343	0.8762

Table 4.2: Mahnob corpus experimental results

features (11) on the other metrics. In Chapter 6 we revisit the question of how linguistic features should be treated in continuous models.

Chapter 5

Continuous Turn-Taking Decisions With POMDPs

5.1 Motivation and Related work

5.1.1 Motivation

In Chapters 3 and 4 we proposed ways of improving CTT performance as measured by the overall loss and metrics representing different turn-taking decisions. The metrics we looked at (PAUSE 50, PAUSE 500, OVERLAP, and ONSET) give an indication of how the model might perform in a live implementation. The PAUSE 50 and PAUSE 500 metrics represent endpointing decisions where a user silence has been detected, and a judgement must be made as to whether the user is going to continue or not (HOLD/SHIFT). While these endpointing decisions are useful, they are still constrained by fact that these pauses must be detected before making a decision. As such, these types of decisions still belong to the traditional reactive paradigm. One of the appeals of continuous models is that the user's end of turn can potentially be anticipated, allowing for fast turn switches that result in small gaps or small amounts of overlap. However, so far none of the models we have investigated attempts to make these type of predictive decisions.

In the finite state turn-taking machine (FSTTM) approach of Raux and Eskenazi (2009) they take a principled approach to turn-taking decisions in which the relative cost of being in an overlap is weighed against the cost of having a long gap between turns. The basic functionality of the FSTTM works as follows: first, a pause by the user is detected. Then, features extracted from

the user's utterance (as well as contextual features) are used to make a probabilistic prediction as to whether the silence is an EOT (logistic regression models were used for this EOT model by Raux and Eskenazi (2009)). This probability score is then used as input to the FSTTM to decide *how long* the system should wait until it can take a turn. If the user begins speaking again before the predicted time then the system doesn't take a turn. The higher the probability that the pause is an EOT, the shorter the predicted time will be. The length of time that the system waits also depends on predetermined costs that have been assigned to either interrupting or having long gaps after the user has finished. The cost ratios and the EOT model therefore determine the overall latency vs. false-cut-in (FCI) performance of the system. The better the EOT model is, the better the characteristics of the trade-off.

This ability to control the trade-off between latency and FCIs is very useful in practical SDS implementations where, in some cases, reliability may be more important than speed, or vice versa. However, there are issues regarding how this trade-off can best be controlled in continuous models. Firstly, if we wished to use the output of a CTT model with an FSTTM, we would need to find a way of generating probabilistic EOT predictions from the CTT model. In previous chapters, the HOLD/SHIFT predictions were made by a greater-than/less-than comparison of the $\boldsymbol{y}_t$ vectors for each of the speakers in a conversation (see Section 2.3.3). Converting $\boldsymbol{y}_t$ prediction vectors into EOT probabilities is not straightforward. It is therefore desirable to find a way of *probabilistically* modelling EOTs using continuous models.

Another issue is that FSTTMs are designed with non-sequential models in mind, where a single prediction of how long the system should wait before taking a turn is made when a user silence is detected. To fully take advantage of the continuous nature of CTT models, it is desirable to make these types of decisions incrementally, before a user silence is detected. Given that the system is prepared to take a turn, at each frame during the user's turn the system should be able to predict whether it is likely that the user will still be speaking during the next frame and weigh up the cost of either interrupting the user or having a long gap.

POMDPs can be used to sequentially make decisions when there is underlying uncertainty about the current state of a system. We propose that POMDPs present a suitable replacement to FSTTMs for making endpointing decisions within a continuous context. Since they are sequential, they can be used with the incremental outputs of continuous models. POMDPs take into account the probability of being in future states when making decisions at each timestep. This predictive aspect of POMDPs potentially allows CTT model predictions to be employed

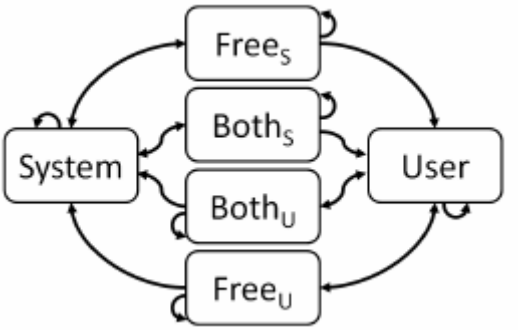


Figure 5.1: The six-state model of turn-taking used in FSTTMs. From Raux and Eskenazi (2009)

iteratively in the calculation of the state likelihoods of future horizons. The rewards for being in states can also be manually controlled which allows for flexible control of the latency vs FCI trade-off.

5.1.2 Overview

In this chapter we show how predictive continuous turn-taking decisions can be made using POMDP-based control to achieve better latency vs FCI trade-off characteristics than thresholding, classification (as per Skantze (2017a)), or FSTTMs (Raux, 2008; Raux and Eskenazi, 2009). In general, exact solutions to POMDPs in continuous observation spaces are intractable (Papadimitriou and Tsitsiklis, 1987), and even approximate solutions can be difficult (Sunberg and Kochenderfer, 2017). We employ three techniques that exploit domain-specific knowledge to make our problem tractable: (1) We discretize the observation space using derived probabilistic expressions that generate the observation probabilities from neural network predictions. (2) We propose a neural architecture that outputs predictions that are designed to be used to derive these observation probabilities. As a by-product of this, we develop an output classifier for turn-taking classification that outperforms previous CTT classification approaches and is more suitable to real-world applications. (3) We use past, future, and present predictions made by our RNN model to recursively calculate the observation probabilities at each future horizon. Using these three techniques, we can efficiently evaluate models by sweeping across cost values to observe the trade-off between latency and false-cut-ins (FCIs).

5.1.3 Finite State Turn-Taking Machine

Our POMDP decision process builds on the state-based representation used by the FSTTM. We also use the FSTTM as a comparison method in our experiments, so in this sub-section we describe it in detail. The FSTTM models turn-taking using a non-deterministic finite-state machine (FSM) that consists of six states, as shown in Fig. 5.1. For the purposes of modelling endpointing, we are only interested in modelling the possible transitions from the User state to the System state. This excludes the $\mathrm{Free_S}$ state and the $\mathrm{Both_S}$ state. The four states which we use are: $\mathbb{S} \in \{SYSTEM, USER, FREE, OVERLAP\}$ which we abbreviate when used in equations as S, U, F, O, respectively. We use these same four states in our POMDP described below. There are two actions available to the dialogue system: $\mathbb{A} \in \{GRAB, WAIT\}$ which influence the state transitions, abbreviated as G and W when used in equations. The FSM assumes that there cannot be a direct transition between the $USER$ and $SYSTEM$ states, but that the intermediary states, $FREE$ or $OVERLAP$, must be transitioned through. This corresponds to the assumption that turn-switches with no gap and no overlap will never occur. A reward is associated with available actions in each state. The reward structure is based on the principle proposed by Sacks et al. (1974) that participants in a conversation attempt to minimize gaps and overlaps. The action with the highest expected reward can then be selected at any time t during the conversation using:

$$argmax_{a \in \mathbb{A}} \sum_{s \in \mathbb{S}} p(s|z_t) R(s,a) \tag{5.1}$$

where $p(s|z_t)$ is a model of the probability of a state occupation given the observations z_t, and $R(s,a)$ is the reward for taking one of the two actions while in this state.[1] When this state-based model is used in the context of endpointing we assume that the FSM begins in the *USER* state and the objective is to determine the point when the user should start speaking. The observations z_t are decomposed into observations that are available at the start of the pause z (which consist of features extracted from the user's utterance and the dialogue context), and observations made during the pause. In the case of the original FSTTM specification of Raux and Eskenazi (2009), the only observation made during the pause consists of the length of the pause itself (τ). More specifically, during a detected pause by the user, it is observed that $d \geq \tau$

[1] In this summary of the FSTTM system of Raux and Eskenazi (2009), we adapt the original formulas to use reward functions $R(\cdot,\cdot)$ instead of cost functions $C(\cdot,\cdot)$. This was done so that the notation is consistent with standard formulations of POMDPs, which are commonly given in terms of reward functions (e.g. Kaelbling et al. (1998)). We adapt the original formulas of Raux and Eskenazi (2009) by assuming $R(\cdot,\cdot) = -C(\cdot,\cdot)$.

where d is the total duration of the pause.

To use the FSTTM for endpointing, silences of a given length (e.g. 200 ms) after a user's utterance are identified using a VAD and then the reward of choosing either $GRAB$ or $WAIT$ is evaluated. The reward of $WAIT$ during $FREE$ is a negative value decreases linearly with the length of the pause $R(F, W) \cdot t$. The reward of the $GRAB$ action while in the $USER$ state, $R(U, G)$, is set to a negative constant. The rewards of $WAIT$ during $USER$ is set to zero, as is the reward for and $GRAB$ during $FREE$. The FSTTM assumes an initial state of $USER$ and we try to predict whether the user will continue speaking or whether the user has relinquished their turn. The probability that the user will not continue speaking is given by:

$$p(F|z, d \geq \tau) = \frac{p(d \geq \tau|z, F)p(F|z)}{p(d \geq \tau|z)} \tag{5.2}$$

$$= \frac{p(F|z)}{p(d \geq \tau|z)} \tag{5.3}$$

where $p(d \geq \tau|z, F)$ is the probability that the total duration of the pause d lasts at least τ ms (the length of the user's observed pause) given that the floor-state is $FREE$. In the FSTTM specification of Raux and Eskenazi (2009), the value of $p(d \geq \tau|z, F)$ is assumed to be equal to 1.0. This equivalence can be interpreted as making a modelling assumption that, given that the user has relinquished their turn (i.e the FSM has entered the $FREE$ state), the pause will continue forever.

The numerator of Eq. 5.2, $p(F|z)$, is the probability that the system is in the $FREE$ state given the speech features observed during the previous IPU, and $p(d \geq \tau|z)$ is the probability the pause lasts at least τ ms. The probability that the user will continue speaking, given the observations and current silence length, is:

$$p(U|z, d \geq \tau) = \frac{p(d \geq \tau|z, U)p(U|z)}{p(d \geq \tau|z)} \tag{5.4}$$

The probability $p(d \geq \tau|z, U)$ is modelled using the knowledge that the distribution of turn-internal pauses can be approximated using an exponential distribution (Jaffe and Feldstein, 1970; Raux and Eskenazi, 2008):

$$p(d \geq \tau|z, U) = e^{\frac{-\tau}{\mu(z)}} \tag{5.5}$$

The parameter $\mu(z)$ is the expected length of the turn-internal pause given the observations.

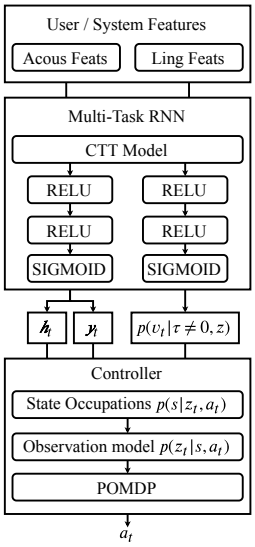


Figure 5.2: Overview of the information flow in the decision process.

We use the mean of the turn-internal pauses of our validation set as a constant value for $\mu(z)$.

Using Eq. 5.1 and description of the rewards given above, the expected reward of $WAIT$ is given by:

$$\frac{p(F|z)R(F,W)\cdot\tau}{p(d\geq\tau|z)} \tag{5.6}$$

The expected reward of $GRAB$ is:

$$\frac{p(d\geq\tau|z,U)(1-p(F|z))}{p(d\geq\tau|z)}R(O,G) \tag{5.7}$$

To perform endpointing we find the predicted value of t_0 when the reward of $WAIT$ exceeds the reward of $GRAB$. This can be solved by finding the solution to:

$$t_0 = \frac{R(F,W)}{R(O,G)}e^{\frac{-t_0}{\mu(z)}}\cdot\frac{1-p(F|z)}{p(F|z)} \tag{5.8}$$

This expression cannot be solved analytically but it can be solved using gradient-based iterative methods such as Newton's method (Abadi et al., 2013; Abdi, 2007).

5.2 CTT Decisions using POMDPs

5.2.1 Overview of the Predictive Model

Our approach to making continuous endpointing decisions with POMDPs relies on a modified CTT model to output predictions that are used as input to the POMDP. We first discuss the changes we made to standard CTT architecture before describing the POMDP controller in section 5.2.2. Using the notation introduced in 2.3.3, the first main difference is that the network predicts y_t, the voice activity for the current frame as well as future frames (i.e. $\boldsymbol{y}_t = [y_t, y_{t+1}, ..., y_N]$, rather than $\boldsymbol{y}_t = [y_{t+1}, y_{t+2}, ..., y_N]$). It therefore acts as a combined current and future voice activity predictor (VAD). We do this to be able to continuously estimate the expected length of a pause that has occurred by using a stored history of past y_t values $\mathbf{h}_t = [y_t, y_{t-1}, ..., y_{t-K}]$. By making the assumption $y_t \perp y_{t+1}$, the expected length of a pause at any given point in the conversation is given by:

$$E[\tau|\mathbf{h}_t] = L \sum_{i=1}^{K-1} i \prod_{n=0}^{i-1} p(y_{t-n} = 0) \prod_{j=i}^{K} p(y_{t-j} = 1) \tag{5.9}$$

where L is the length of a single time-step. This continuous estimate of the expected length of an observed pause is used to approximate the probability that the total duration of the pause will last longer than the current pause length given that the user will continue speaking:

$$p(d \geq E[\tau]|z_t, U) = e^{\frac{-E[\tau|\mathbf{h}_t]}{\mu(z_t)}} \tag{5.10}$$

This makes use of the exponential distribution from Eq. 5.5.

The second main difference is that the network is trained not only to predict the future voice activity of the user $\boldsymbol{y}_t$, but also directly predict whether there will be a SHIFT when the speaker's IPU ends. This conditional probability $p(v_t|\tau \neq 0, z_t)$ is modelled as the output of a sigmoid activation layer that is optimized in conjunction with the $\boldsymbol{y}_t$ predictions. We use the notation $p(v_t = 0|\tau \neq 0, z_t) = 1 - p(v_t = 0|\tau \neq 0, z_t)$ to denote the network prediction for HOLD. The training data for $p(v_t|\tau \neq 0, z_t)$ is generated automatically from a frame-based IPU segmentation of the dataset. The labelling scheme (shown in Fig. 5.3) labels all silent frames in a conversation as either HOLD (0) or SHIFT (1). We also label *n_pre* frames preceding the end of each IPU with the HOLD/SHIFT values. For example, *n_pre*=2 corresponds to 100 ms before the end of the IPU. All other frames are labelled as undefined. At training time we only back-propagate the

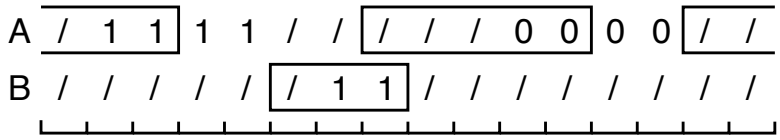


Figure 5.3: Continuous HOLD/SHIFT labelling procedure (*n_pre*=2)

error for the defined labels. To make HOLD/SHIFT classification-based decisions on our test set, we choose threshold values for $p(v_t|\tau \neq 0, z_t)$ that maximize the performance of classifications on our held-out validation set.

The motivation for labelling *n_pre* spoken frames is that we would like our network to predict turn-switches before they happen. If we only predict at silence values, the turn-switch predictions directly preceding the end of the IPU will not be as accurate and we will have less no-gap/no-overlap transitions, which will lower the overall mean latency. Since many turn-relinquishing cues tend to be located near the end of IPUs, we find that including these spoken frames improves the performance. There is a trade-off however, since including too many spoken frames risks including noisy information that is irrelevant to making decisions as to who will take the floor in the next IPU.

5.2.2 POMDPs for turn-taking prediction

Our control process is a short-horizon POMDP that uses the predictions from our CTT model to generate the observation probabilities that allow the POMDP to make decisions. It uses the two separate types of outputs from the network ($\boldsymbol{y}_t$ and $p(v_t|\tau \neq 0, z_t)$) to predictively combine them during the calculation of observation probabilities at future horizons. POMDPs are described by the tuple $(\mathbb{S}, \mathbb{A}, \Omega, T, R, Z, \gamma)$ where $\mathbb{S}$ is a set of states (same four possible endpointing states described by Raux and Eskenazi (2009)); $\mathbb{A}$ is a set of actions; Ω is a set of possible observations; T is the state transition function $T(s, a, s') = p(s'|s, a)$; R is the reward function $R(s, a)$; Z defines the observation model $Z(s', a_{t-1}, z_t) = p(z_t|a_{t-1}, s')$; and $\gamma \in [0, 1)$ is a discount factor. In POMDPs the true state of the process is unknown. The observations at each time step are used to maintain a distribution over the possible states b_t. The belief state is updated at each time step using the belief state update function $b_t = \beta(b_{t-1}, a_{t-1}, z_t)$, which is defined as the following Bayesian filter:

$$b_t(s') = \beta(b_{t-1}, a_{t-1}, z_t)(s') = \frac{1}{p(z_t|b_{t-1}, a_{t-1})} Z(s', a_{t-1}, z_t) \sum_{s \in \mathbb{S}} T(s, a_{t-1}, s') b_{t-1}(s) \tag{5.11}$$

where $p(z_t|b_{t-1}, a_{t-1})$ is a normalizing constant that uses:

$$p(z_t|b_{t-1}, a_{t-1}) = \sum_{s' \in \mathbb{S}} Z(s', a_{t-1}, z_t) \sum_{s \in \mathbb{S}} T(s, a_{t-1}, s') b_{t-1}(s) \tag{5.12}$$

The actions that the system takes are determined by a deterministic mapping from belief states to actions $\pi(b) \in A$. The value of some policy π is then given by

$$V^{\pi}(b_t) = R_B(b_t, \pi(b_t)) + \gamma \sum_{z_{t+1} \in \Omega} p(z_{t+1}|b_t, \pi(b_t)) V^{\pi}(b_{t+1}) \tag{5.13}$$

where

$$R_B(b, a) = \sum_{s \in \mathbb{S}} b(s) R(s, a) \tag{5.14}$$

Transition Function (T) The transition function $T : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \mapsto [0, 1]$ gives the probability of transitioning from state s to state s' if the action a is performed $T(s, a, s') = p(s'|s, a)$. The two available actions to the system when endpointing are $WAIT$ and $GRAB$. When conditioned on the system action a of the previous time step, the belief space is partitioned into two non-overlapping spaces delineated by the dialogue system's spoken states {SYSTEM, OVERLAP} and the system's silent states {USER, FREE}. Since we are only interested in predicting the optimal point to perform a single $GRAB$ action we can restrict the transition function to never transition back from to the spoken states once a $GRAB$ action has been performed. This corresponds to assigning a probability of zero to the transition from the speaking states to the silent states. It should be noted that transitions from OVERLAP to SYSTEM have non-zero probabilities. This allows GRAB actions to be performed when there is a non-zero probability that the the action will initially result in an overlap but eventually end up in a SYSTEM state at a subsequent horizon. This enables fast turn switches with small amounts of overlap. However, transitions from SYSTEM to OVERLAP are assumed to have zero probability. This is a necessary result of the aforementioned assumption (introduced by Raux and Eskenazi (2009)) that once the user has relinquished their turn they will not restart their turn. Given these constraints, the transition function probabilities are then estimated on the validation set using maximum

likelihood estimation.

While encoding assumptions into the transition and observation models might seem like bad practice, it is commonly done when building real-world POMDP systems (Young et al., 2013). It serves as a method of using domain knowledge to impose strong priors. Young et al. (2013) give an example from a restaurant recommendation SDS where the user action is the statement "I want a Chinese restaurant" can be used to set all probabilities for system responses that don't recommend a Chinese restaurant to zero.

Observation Model (Z) The observation model $Z(s', a_{t-1}, z_t) = p(z_t|a_{t-1}, s')$ gives the probability of the observations z_t given that action a_{t-1} is performed, and the resulting state is s'. In classical POMDP specifications, observations consist of discrete events (e.g. words), or sequences of discrete events (e.g. utterances). An issue typically encountered with this approach is that the number of possible observations can easily become too large to model directly and typically domain-specific properties of the task must be exploited to make the problem tractable (Young et al., 2013). More recently, neural networks have been used to replace the different components of POMDP systems (Hausknecht and Stone, 2015; Ha and Schmidhuber, 2018; Gangwani et al., 2019). This allows observation models to be compactly represented using far fewer parameters.

For our problem of turn-taking decision-making, we would like to use acoustic and linguistic features extracted from the user's speech in a frame-by-frame manner. Hypothetically, one option for calculating $p(z_t|a_{t-1}, s')$ might be to use the most recent ASR results, as well as extracting summary acoustic features (e.g. discrete variables representing high or low pitch). However, even with a small vocabulary, calculating the probability of a sequence of words and features (conditioned on $a_{t-1}$and s') is intractable since it would require a summation over all possible combinations of sequence observations. We therefore must resort to methods of approximating $p(z_t|a_{t-1}, s')$, which we achieve by estimating $p(s'|a_{t-1}, z_t)$ using a neural network, and then applying Bayes rule.

The conditional state occupation probabilities $p(s'|a_{t-1}, z_t)$ are estimated using the neural network outputs $p(v_t|\tau \neq 0, z_t)$ and $\boldsymbol{y}_t$. The derivations of the conditional state probabilities are given later in this chapter in section 5.2.3. Since POMDPs require observation probabilities in the form $p(z_t|a, s')$ we employ an approach commonly used in ASR neural network acoustic

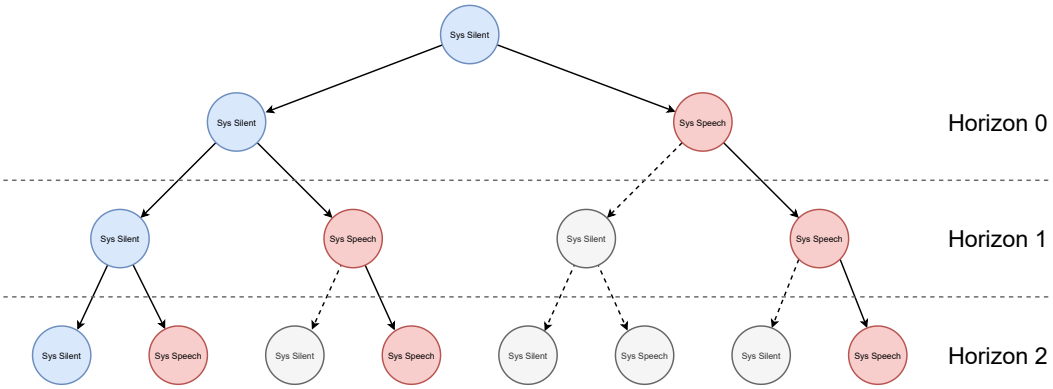


Figure 5.4: A diagram showing a three-step conditional plan.

modelling for HMMs, where scaled likelihoods (e.g. Morgan and Bourlard (1995)) are used:

$$\frac{p(z_t|a_{t-1}, s')}{p(z_t|a_{t-1})} = \frac{p(s'|a_{t-1}, z_t)}{p(s'|a_{t-1})} \tag{5.15}$$

As was assumed by Morgan and Bourlard (1995), the probability of the observations given an action $p(z_t|a_{t-1})$ is taken to be constant. We can then use:

$$p(z_t|a_{t-1}, s') = \frac{1}{C}\frac{p(s'|a_{t-1}, z_t)}{p(s'|a_{t-1})} \tag{5.16}$$

where $C = \sum_{s'} \frac{p(s'|a_{t-1}, z_t)}{p(s'|a_{t-1})}$. The state probabilities $p(s'|a_{t-1})$ can be calculated using the relative frequency count of each resulting state given an action.

In order to calculate the conditional state probabilities for each state $p(s'|a_{t-1}, z_t)$ we can use the domain knowledge that for any given action a_{t-1} there are only two possible states at time t, and that the probabilities for these two states must be complementary. More specifically, if $a_{t-1} = WAIT$ then:

$$p(s' = FREE|a_{t-1} = WAIT, z_t) = 1 - p(s' = USER|a_{t-1} = WAIT, z_t) \tag{5.17}$$

and if $a_{t-1} = GRAB$:

$$p(s' = SYSTEM|a_{t-1} = GRAB, z_t) = 1 - p(s' = OVERLAP|a_{t-1} = GRAB, z_t) \tag{5.18}$$

Solving the POMDP Fig. 5.4 shows a diagram of all possible conditional plans for $\lambda = 2$, starting from a silent state. The nodes in red represent the states during which the system is

speaking (SYSTEM and OVERLAP). The nodes in blue represent the states in which the system is silent (USER and FREE). The nodes in grey correspond to state paths that are not allowed to occur under the assumed restriction that we cannot transition back from a speaking state once the GRAB action has been performed. As mentioned above, this corresponds to setting zero transition probabilities for speaking to non-speaking state transitions. The graph defined in the region labelled "Horizon 0" corresponds to making a decision to GRAB or WAIT that is based solely on the current belief state, and not on any future horizons. Using Eq. 5.13, the value of a policy based only on the current belief state is given by:

$$V^{\pi}(b) = R_B(b, \pi(b)) \tag{5.19}$$

This is equivalent to setting the discount factor in Eq. 5.13 to zero.

In order to calculate the value of actions in future horizons, we use the future predictions of the neural network in the recursive calculations of V^{π}. At each new horizon, predictions of $\boldsymbol{y}_t$ for future time-steps are used to calculate new values for $E[\tau|\mathrm{h}_t]$ as well as calculate a new probability that the user is still speaking $p(\tau = 0)$. These values are then used along with with the current prediction for $p(v_t|\tau \neq 0, z_t)$ to calculate new observation probabilities $p(z_{t+1}|b_t, \pi(b_t))$ at each horizon. The new observation probabilities are also used to estimate the future belief state b_{t+1}. We can then calculate the expected reward for each conditional plan within a specified horizon λ, and select the plan with the highest expected reward. This use of the future predictions to make decisions (rather than solely basing them on the predictions for the current frame) allows the POMDP to plan ahead. By having a system that only reacts to the current frame we are also limiting the minimum possible latency to one frame-step. Whereas, by using a predictive model we can potentially achieve zero and negative latencies (although, we currently consider negative latencies to be FCIs).

5.2.3 Continuous State Occupation Models

As mentioned in section 5.2.2, the observation probabilities $p(z_t|a_{t-1}, s')$ are estimated from the conditional state occupation probabilities $p(s'|a_{t-1}, z_t)$ using scaled likelihoods (see Eq. 5.16). In this subsection we describe how we calculate $p(s'|a_{t-1}, z_t)$ for each of the four states using the neural network outputs.

Silent States: User state occupation probability We model the probability that we are in the $USER$ state as the union of the probability that the user is speaking, with the probability that the user is *not* speaking but will start speaking again:

$$p(\mathrm{U}|z_t, a_{t-1} = W) = p(\tau = 0|z_t, a_{t-1} = W) + p(\tau \neq 0, \mathrm{HOLD}|z_t, a_{t-1} = \mathrm{W}) \quad (5.20)$$

The estimates of the probabilities of $\tau = 0$ and $\tau \neq 0$ both use the network's VAD prediction for the current frame y_t. The probability that the user will start speaking again, $p(\mathrm{HOLD}|z_t, a_{t-1} = \mathrm{W})$, is modelled using the neural network's estimate for the probability of whether there will be a SHIFT when the speaker's IPU ends $p(v_t|\tau \neq 0, z_t)$, as well as using the estimate for whether the current pause will last longer, given that it is a turn-internal pause $p(d \geq E[\tau]|z_t, U)$. Similarly to the FSTTM approach discussed in Section 5.1.3, we supplement the trained turn-taking model that captures *spoken* turn-taking cues with domain knowledge of the distribution of pauses. Using $p(v_t|\tau \neq 0, z_t)$ alone is not sufficient for two main reasons:

1. In using a model trained using supervised learning (rather than reinforcement learning) the network can only accurately model observations sequences that are observed in the training data. Since we are training on human-human conversational corpora (in order to use large training corpora to capture acoustic and linguistic turn-taking cues), the network cannot learn that, within the context of an SDS, long stretches of silence after a user's utterance indicate that the user has relinquished their turn. In other words, the network is not presented with training data that represents the knowledge that longer silences by the user indicate an end of turn.

2. While the network is capable of forming predictions based on behaviour that is observed during the spoken parts of the interaction (as well as incorporating knowledge about short pauses of the type observed in the data), over longer stretches of silence the network predictions for $p(v_t|\tau \neq 0, z_t)$ will trend towards chance (average) predictions. This means that when the user has finished their turn, the prediction of $p(v_t|\tau \neq 0, z_t)$ in the vicinity of the end of their IPU must be high enough to trigger the GRAB action. If it is not triggered, $p(v_t|\tau \neq 0, z_t)$ will only decrease over time. This will cause the system to hang indefinitely in belief states that choose the WAIT action. Meaning that, if the neural network does not detect any turn-switch cues (or they are only weakly detected) when the user intends

to relinquish their turn, the POMDP will have no way to recover since the GRAB action will never be chosen.

Due to these two reasons, the use of $p(v_t|\tau \neq 0, z_t)$ on its own does not accurately model pauses and gaps in SDS conversations. We incorporate domain knowledge of the distribution of pauses using:

$$\begin{aligned} p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W}) &= p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W}, \mathrm{F})p(v_t = 1|\tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \dots \\ &\quad \dots + p(d \geq E[\tau]|z_t, \mathrm{U})p(v_t = 0|\tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \quad (5.21) \\ &= p(v_t = 1|\tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \dots \\ &\quad \dots + p(d \geq E[\tau]|z_t, \mathrm{U})p(v_t = 0|\tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \quad (5.22) \end{aligned}$$

where $p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})$ denotes the combined probability that a pause will continue, agnostic of whether it is a turn-internal pause or the user has relinquished the turn. It is worth noting that in Eq. 5.21, $p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W}, \mathrm{F}) = 1$ under the previously stated assumption that if we are in the FREE state a pause will continue until the GRAB action is performed by the system. The domain knowledge of pause distributions is incorporated into our HOLD predictions using:

$$\begin{aligned} p(\mathrm{HOLD}|\tau \neq 0, z_t, a_{t-1} = W) &= p(v_t = 0|d \geq E[\tau], \tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \\ &= \frac{p(d \geq E[\tau], v_t = 0|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})} \quad (5.23) \\ &= \frac{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W}, v_t = 0)}{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})} \dots \quad (5.24) \\ &\quad \dots \times p(v_t = 0|\tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \\ &= \frac{p(d \geq E[\tau]|z_t, \mathrm{U})p(v_t = 0|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})} \quad (5.25) \end{aligned}$$

Therefore, from Eq. 5.20:

$$\begin{aligned} p(\mathrm{U}|z_t, a_{t-1} = W) &= p(\tau = 0|z_t, a_{t-1} = W) + p(\tau \neq 0|z_t, a_{t-1} = W) \dots \quad (5.26) \\ &\quad \dots \times \frac{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W}, \mathrm{U})p(v_t = 0|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})} \end{aligned}$$

Silent States: Free state occupation probability We model the probability that we are in the F state by the probability that the user is not speaking and will not start speaking again:

$$p(\mathrm{F}|z_t, a_{t-1} = W) = p(\tau \neq 0, \mathrm{SHIFT}|z_t, a_{t-1} = \mathrm{W}) \tag{5.27}$$

where:

$$\begin{aligned} p(\mathrm{SHIFT}|\tau \neq 0, z_t, a_{t-1} = W) &= p(v_t = 1|d \geq E[\tau], \tau \neq 0, z_t, a_{t-1} = \mathrm{W}) \\ &= \frac{p(d \geq E[\tau], v_t = 1|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})} \quad (5.28) \\ &= \frac{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W}, \mathrm{F})p(v_t = 1|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})} \\ &= \frac{p(v_t = 1|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})} \quad (5.29) \end{aligned}$$

So:

$$p(\mathrm{F}|z_t, a_{t-1} = W) = p(\tau \neq 0|z_t, a_{t-1} = W)\frac{p(v_t = 1|\tau \neq 0, z_t, a_{t-1} = \mathrm{W})}{p(d \geq E[\tau]|z_t, a_{t-1} = \mathrm{W})} \tag{5.30}$$

It can be shown that the sum of $p(\mathrm{F}|z_t, a_{t-1} = W)$ and $p(\mathrm{U}|z_t, a_{t-1} = W)$ will equal one. In real-world implementations of the system, it may be easier to simply calculate $p(\mathrm{F}|z_t, a_{t-1} = W)$ from Eq. 5.30 and then use $p(\mathrm{U}|z_t, a_{t-1} = W) = 1 - p(\mathrm{F}|z_t, a_{t-1} = W)$.

Spoken States: System state occupation probability Given that the GRAB action was performed in the previous timestep, the SYSTEM state occupation probability is calculated using the neural network VAD prediction for whether the user is silent:

$$p(\mathrm{S}|z_t, a_{t-1} = G) = p(\tau = 0|z_t, a_{t-1} = \mathrm{G}) \tag{5.31}$$

Spoken States: Overlap state occupation probability Given that the GRAB action was performed in the previous timestep, the OVERLAP state occupation probability is calculated using the neural network VAD prediction for whether the user is speaking:

$$p(\mathrm{O}|z_t, a_{t-1} = G) = p(\tau \neq 0|z_t, a_{t-1} = \mathrm{G}) \tag{5.32}$$

5.3 Implementation Details

5.3.1 Data

We perform our experiments using the Switchboard 1 dataset Godfrey et al. (1992). We train only on the files with from the MS-State transcriptions and evaluate using the HUB'5 set for evaluation. The utterance labels are used to create frame-based labels for speech activity. From the speech activity labels we extract utterances (IPUs) based on pause thresholds. We use a pause threshold of 50ms, which is shorter than is commonly used but is more appropriate for the current context of evaluating fast turn-switch performance. For evaluation we exclude IPUs that end in overlap, as well as IPUs that are succeeded by pauses of longer than 2 seconds. The IPUs are then labelled for HOLD/SHIFT according to the labelling scheme discussed above and shown in Fig. 5.3.

5.3.2 Training Procedure

The neural network is trained to predict a vector of $N = 20$ voice activity labels from the current time-step to 19 steps in the futures. We add a secondary loss to predict the HOLD/SHIFT labels where we weight the secondary loss by half the weight of the voice activity predictions. The network is trained to minimize binary-cross-entropy loss with sequence lengths of 800 frames. As linguistic features we use an enumerated vocabulary that is used to learn an embedding of dimension 128 that is jointly optimized with the rest of the network. As acoustic features we use 40 log-mel filter banks as well as 17 features from the eGEMAPs Eyben et al. (2016) feature set (excluding the MFCCs). These features are concatenated with the features from the previous frame to create an input acoustic vector of size 114. The network has the following settings: LSTM layers all have a hidden unit sizes of 128; we use dropout of 0.2 between layers; a learning rate of 1e-5; and two RELU layers for each multitask prediction. We use early-stopping, where we evaluate on a held-out validation set until the loss stops decreasing and then test once on the full test set at the end.

5.3.3 Testing

POMDP Evaluating conversational turn-taking models in non-real-world settings requires making the assumption that, at each IPU, the user is speaking and the SDS would like to decide if, and when, it should take a turn. For traditional non-sequential classification approaches this consists of training classifiers to make a decision once a pause of a given threshold is observed. In the continuous context, this decision is complicated by the fact that we would like these decisions to incorporate predictions of whether a user is *about* to stop speaking. Therefore, the SDS could potentially interrupt the user before they are finished speaking. To take this into account, we use $\mathrm{FCI} = \frac{\mathrm{Interruptions+FP}}{\mathrm{IPU\ count}}$, where $\mathrm{Interruptions}$ are the number of correctly classified turn switches that were triggered before the user was finished, and FP is the number of incorrectly classified turn switches. In our test data, 68.6% is the maximum FCI that can be achieved with the non-continuous models, whereas the continuous models can potentially interrupt all of the IPUs.

The output of the POMDP is a series of actions at each time-step during the user's IPU, and in the silence that comes after the IPU. We consider the first frame at which the G action is chosen to be the endpoint decision. To be able to endpoint *after* the pause is completed we pad the predictions for $\boldsymbol{y}_t$ during each IPU with two seconds of silence and repeat the last prediction for $p(v_t|\tau \neq 0, z_t)$. This is an approximation of the assumption made in the FSTTM of Raux and Eskenazi (2009) that once the user is finished speaking, they will not start speaking again. We acknowledge that there are limitations with this evaluation scheme but regard it as the most straightforward method of making endpointing decisions with continuous models after the user's pause has finished. Another potential option would be to generate the neural-network outputs using artificial silence.

For the POMDP settings we set the length of the history $\mathtt{h}$ to be $K = 20$ and used a discount factor of $\gamma = 1$. The reward settings that we used for the POMDP were: $R(F, W) = -\epsilon$, $R(S, G) = \epsilon$, $R(U, W) = 1$, $R(O, G) = -1000$ with all other rewards set to zero. We sweep across values of ϵ to observe the latency vs. FCI trade-off.

Classification We devise a classification-based approach to making decisions with continuous predictions as a comparison method for our POMDPs. We take the output predictions for $p(v_t|\tau \neq 0, z_t)$ at the end of each IPU and calculate a classification-threshold σ that maximizes the number of correct classifications (on the validation set). We then sweep across the pause-

durations for the IPUs. If the pause-duration is less than the pause-threshold we classify it using $p(v_t|\tau \neq 0, z_t)$ and σ. If the pause-duration is greater than the pause-threshold, we classify it as SHIFT. Pseudo-code is given in Algorithm 2. We note that this approach assumes perfect VAD performance and is not predictive.

```
if current_pause < pause_threshold:
    if p(v_t|τ ≠ 0, z_t) > σ:
        decision = SHIFT
    else:
        decision = HOLD
else:
    decision = SHIFT
```

Algorithm 2: Pseudo-code for the classification-based decision approach.

FSTTM To enable comparisons, we evaluate the FSTTM controller using the final $p(v_t|\tau \neq 0, z_t)$ prediction from each IPU as $p(F|z)$ described by Raux and Eskenazi (2009). The value of $\mu(z)$ is calculated from the data and is the same value used for the POMDPs.

Baseline We calculate a baseline performance based on simple thresholding of pause durations to classify HOLD or SHIFT at for pause.

5.4 Discussion

In Fig. 5.5 we plot the latency vs. FCI curves for our POMDP controllers and the three other comparison methods. The dashed line indicates the 50 ms of silence after the user's IPU. Comparing the performance of the two POMDPs we can see that the inclusion of predictions at the first horizon greatly improves the POMDP decisions. Since using the POMDP without a horizon only makes decisions based on the current belief state as calculated by the Bayesian filter, future predictions cannot be taken into account. We observed that altering the value of $R(O, G)$ changed the decay rate of the horizon-one curve. We did not see any improvement in the predictions when using a horizon greater than one; the predictions were almost identical to those at the first horizon. This be explained by the notion that, for turn-taking prediction, where we are trying to achieve no-gap/no-overlap, knowledge of the current frame and the next frame is sufficient to make optimal decisions. Our proposed classification approach has strong

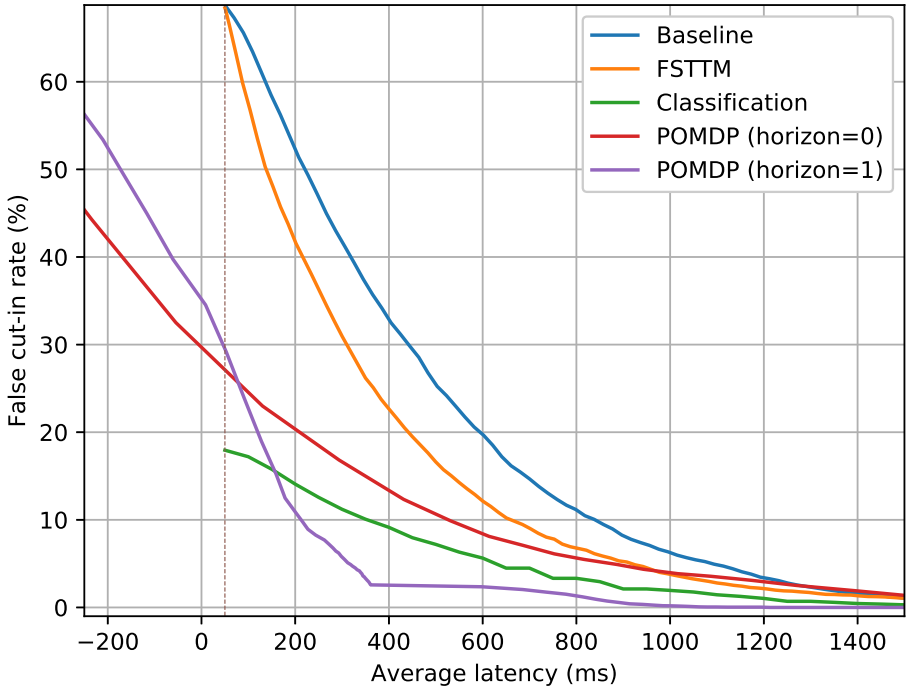


Figure 5.5: Latency vs. FCI curves for our POMDP controller and the three comparison decision making methods. The dashed vertical line represents 50 ms of silence after the user's IPU.

	PAUSE 50	PAUSE 250	PAUSE 500
Comparison Method (Skantze, 2017b)	0.832	0.868	0.885
Labelled Prediction (Ours)	0.840	0.871	0.884

Table 5.1: F-scores for HOLD/SHIFT classifications

performance, outperforming the POMDP with no horizon. However, as mentioned previously, the classification approach has an advantage in that it uses the ground-truth speech activity labels from the dataset. The error that would be introduced by using a VAD is not taken into account in this approach. The POMDP does not rely on binary VAD decisions to make decisions and instead makes use of the probabilistic prediction for speech activity that is output by the neural network, $p(\tau \neq 0 | z_t, a_{t-1} = W)$.

Table 5.1 shows how the performance of our labelled-prediction method compares with the previous comparison method from Skantze (2017b). We observed that our labelled method outperformed the comparison method at short pauses, while predictions converged towards those of comparison method as the pause got longer. This can be attributed to the previously mentioned behaviour of the RNN losing information (trending towards chance), when it observes stretches of silence.

5.5 Conclusion

While our current system shows how we can make endpointing decisions with no-gap and no-overlap, we note that in practical SDSs consideration should be given as to whether we would always want to respond as quickly as possible to the user. In certain cases it may be more natural or polite to artificially generate pauses before the system responds, even when the system is confident of what the rest of the user's utterance will be (see Leviathan and Matias (2018)). We argue that discovering the *appropriate* moment for the system to speak an utterance depends also on extra semantic factors not modelled here. For instance, the response time should depend on not only what the user has said but also what the system intends on saying. If the system is responding with a minimal response such as a backchannel, short amounts of overlap may be appropriate or desirable. If the system is responding to a sensitive question by the user, it may be more appropriate for the system to artificially insert a gap. In the following chapter we devise a generative model that is capable of generated response timings based on the context of both the user's turn and the upcoming system turn.

Chapter 6

Neural Generation of Dialogue Response Timings

6.1 Introduction

6.1.1 Motivation and Related work

In the previous chapter we proposed a control process for a modified CTT model. The control process employed the predictive capabilities of CTT models to make responsive turn-taking decisions. We proposed that one of the main benefits of the control process was that the trade-off between latency and FCIs could be flexibly controlled. Manipulating the reward parameter ϵ controlled the relative reward for grabbing the floor when it was free versus the cost of being in an overlap state. The predictive nature of the CTT model allowed the POMDP to anticipate upcoming user EOTs and respond with lower latencies and minimal overlap.This control process builds on the FSTTM of Raux and Eskenazi (2009) and is designed with the objective of avoiding interrupting the user while keeping the lengths of gaps and overlaps as low as possible.

However, this approach does not emulate naturalistic response offsets since in human-human conversation the distributions of response timing offsets have been shown to differ based on the context of the first speaker's turn and the context of the addressee's response (e.g. Sacks et al. (1974), Levinson and Torreira (2015), Heeman and Lunsford (2017)). It has also been shown that listeners have different anticipations about upcoming responses based on the length of a silence before a response (Bögels et al., 2019). If we wish to realistically generate human-

human offset distributions in SDSs it is necessary to design response timing models that take into account the context of the user's speech and the upcoming system response. For example, offsets where the first speaker's turn is a *backchannel* have been observed to occur in overlap more frequently than turn switches where both the first and second utterances are normal turns (Levinson and Torreira, 2015). Accordingly, if we apply this observation to SDSs, in the context of the user producing a *backchannel* it may be appropriate for the system's turn to overlap with the end of a user's utterance. It may even be desirable for the sake of faster communication.

As another example, in human-human conversations it has been observed that *dispreferred* responses (responses that are not in line with the suggested action in the prior turn (Levinson, 1983)) are associated with delays of over 700 ms, and that responses with gaps of over 300 ms are less likely to be associated with full acceptance (Kendrick and Torreira, 2015; Bögels et al., 2019). This has implications for how the timings of dispreferred system responses are perceived by the user. If we consider the following exchange in an automated booking scenario:

> User: Can I get a booking for tomorrow?
>
> System: No, I'm afraid we're booked up.

a rapid response from the SDS in the second utterance, that includes overlap or has very little gap, would be inappropriate and could be construed as rude. In short, the offsets between turns have semantic significance, and it has been shown that listeners are sensitive to these differences in timings (Bögels et al., 2019).

These details concerning the timings of responses are communicative elements of natural spoken language that is often ignored in SDS designs. As a result, there is a risk of generating a discordance between the response timings and the lexical and/or prosodic elements of the synthesized system turn. There is also the risk that the meaning or sentiment of a system's utterance could be interpreted in an unintended manner. There is therefore a motivation for matching a response timing with the semantics of a system response, in a similar way that the lexical (e.g. Angeli et al. (2010), Wen et al. (2015), Wen and Young (2019)) or prosodic (e.g. Taylor (2000), Skerry-Ryan et al. (2018)) aspects of a response can be generated to match the system's intended meaning.

6.1.2 Overview

We introduce Response Timing Network (RTNet), shown in Fig 6.1. The objective behind RTNet is to generate realistic response timings for SDS utterances that take into account the

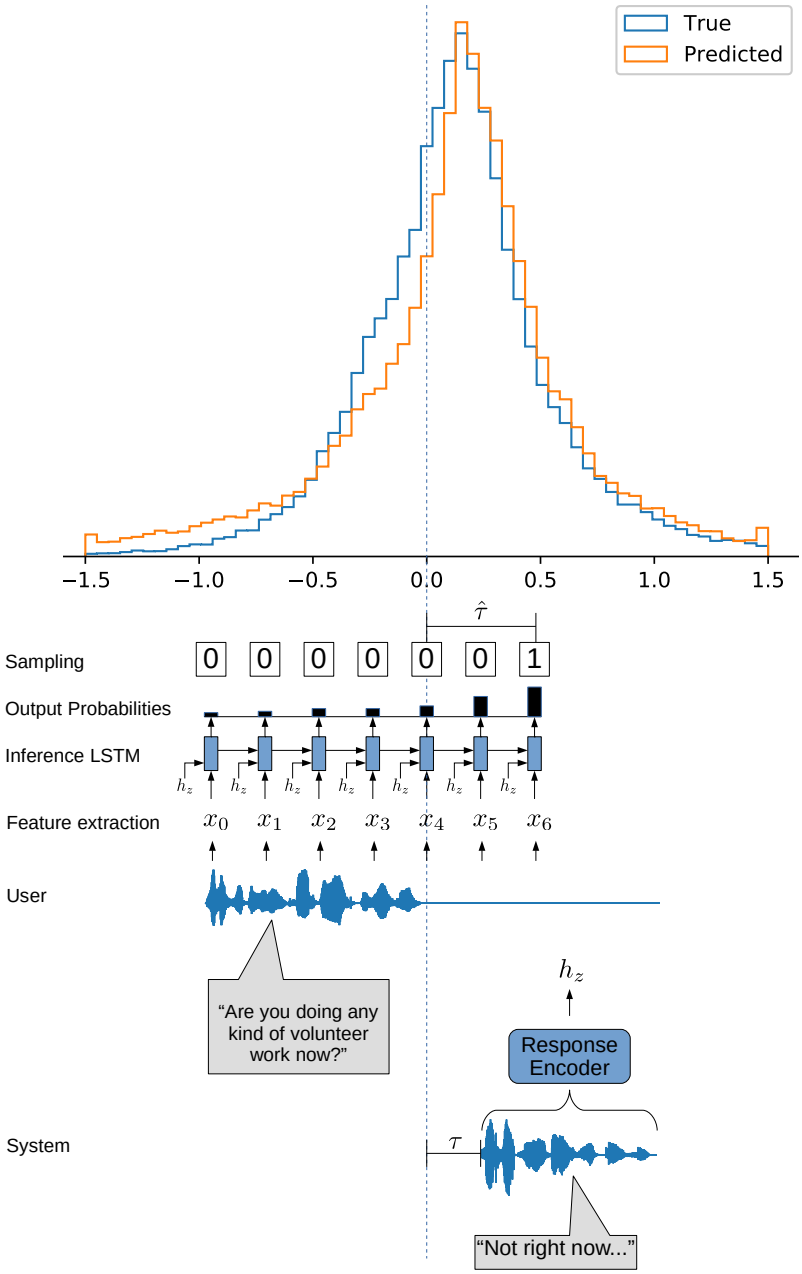


Figure 6.1: Overview of the proposed response timing network (RTNet). The network generates the distributions of turn-switch offset timings (shown at in the top distribution). There are two main components: a response encoder that generates an encoding of a system response h_z, and an inference LSTM that generates the probability of generating a response at a given frame (50 ms frame rate). The inference network takes as input a concatenation of user features x_n with the response encoding h_z. The output probabilities are sampled from to generate a turn switch offset $\hat{\tau}$. (N.B. the scale of the generated distribution offset timings and the inference LSTM prediction rate are not drawn to scale)

user's previous utterance and the semantics of the SDS response. The network operates using both acoustic and linguistic features extracted from the user and system turns. The two main components are an encoder, which encodes the system response (h_z), and an inference network, which takes a concatenation of user features and h_z.

The overall aim is to generate the distribution of timing offsets of real human conversations in a way that could be implemented in a spoken dialogue system. There are several notable challenges in this objective. Firstly, the distribution of turn-switch offsets (shown in the top of Fig. 6.1) is such that 30.1% of all offsets occur before the user has finished speaking, with many turn switches occurring 500ms or more before the user's utterance is finished. The network must therefore incrementally predict when the user is likely to finish speaking using the user's extracted acoustic and linguistic information. Secondly, it has been shown that the timing of responses depend on the semantic content of the user's utterance. For example, as mentioned previously, if the user is producing a *backchannel* there should be a higher probability that the system will begin their utterance in partial overlap with the end of the user's turn. To model this, the network must capture the semantic content of the user's turn and judge what type of offset is appropriate. Thirdly, since the semantic content of the system's response has an impact on the offset distribution, the encoder must be able to capture any relevant information in the response that affects the offset timing. For example, in the automated booking scenario above, the encoder should capture that the response is dispreferred.

RTNet operates within an incremental SDS framework (as discussed in Section 2.2.3) where information about upcoming system responses may be available before the user has finished speaking. RTNet also functions independently of higher-level turn-taking decisions that are traditionally made in the dialogue manager (DM) component. Typically, the DM decides when the system should take a turn and also supplies the natural language generation (NLG) component with a semantic representation of the system response (e.g. intents, dialogue acts, or an equivalent neural representation). The semantic representation is then converted into text by the NLG, and then speech by the speech synthesis (TTS) component. Any of the system response representations that are downstream from the DM's output representation (e.g. lexical or acoustic features) can potentially be used to generate the response encoding. Therefore, we assume that the decision for the system to take a turn has already been made by the DM and our objective is to predict (on a frame-by-frame basis) the appropriate time to trigger the system turn.

It may be impractical in an incremental framework to generate a full system response and then re-encode it using the response encoder of RTNet. The response encoder of RTNet assumes that the surface realization of the entire system response has been generated by the NLG and that the acoustic feature representation of the response has been created by the TTS system. This creates a computational bottleneck since the system cannot start executing the response until it has been processed by RTNet in its entirety. One of the appeals of incremental systems is that the system can start speaking before the entire response has been generated. RTNet introduces a non-incremental component into an incremental pipeline that disrupts the response generation process.

To address this issue, we propose an extension of RTNet that uses a variational autoencoder (VAE) (Kingma and Welling, 2014) to train an interpretable latent space which can be used to bypass the encoding process at inference-time. This extension (RTNet-VAE) allows the benefit of having a data-driven neural representation of response encodings that can be manipulated without the overhead of the encoding process. By incorporating a VAE at the end of the response encoder network we are able to create a vector representation of responses that learns to cluster similar dialogue acts together, without being trained explicitly to do so. This representation can then be used by the DM to generate appropriate timings for a given response.

At inference-time, the use of the latent space of the VAE removes the bottleneck introduced by the RTNet encoder. We no longer have to wait for the entire system response to be generated by the NLG and TTS components. All that is required is a semantic representation of the upcoming system response which can be used to sample from the trained latent space. The semantic representation can be the same as the one used by the NLG component. For example, if the upcoming system turn is a *disagree* dialogue act, the semantic representation of *disagree* will be passed to the NLG component which will generate the text, which will then be converted into speech by the TTS system. The use of a VAE enables us to sample a vector representation from the latent space based on the *disagree* semantic representation. As soon as the first incremental unit output of the TTS component has been produced, the response can be triggered (or not) based on the vector representation from the latent space. The vector representation can also be generated in parallel with the NLG and TTS components. In short, RTNet-VAE removes the bottleneck that would be required for RTNet to function.

The vector representation also does not suffer from the problems of rigid distinctions between dialogue act labels. As discussed in Section 2.1.3, in many instances the definition of what

constitutes a *backchannel* and what constitutes an *agreement* turn may be opaque (e.g. "Yeah", "uh-huh"). We argue that continuous vector representations are more suited to modelling these subtleties than discrete dialogue act labels. The continuous vector representation is also flexible in that representations of dialogue acts can be easily manipulated using vector algebra. For example, we can use interpolation between *disagreement* and *agreement* vectors to achieve perceptually intermediate offset distributions. The DM can then employ this property of the vectors to flexibly generate response timings.

Additionally, since the network captures conversational behaviours that it has observed over the whole training set, we can use the generative model to artificially generate multiple offsets examples for a given response-type or user-turn context. In section 6.4.6 we use this to analyse response behaviours our model has learned for dialogue acts with only a small number of labelled examples. By generating multiple examples of dialogue acts we can approximate distributions for dialogue acts where we only have a small number of labelled examples. This has potential benefits for conversation analysis (CA) since we can investigate the behaviours that the network has learned in a flexible manner that avoids the expense of performing manual annotations.

While the proposed models in this chapter are distinct from the CTT models used in other chapters in both their objectives and their architectures, they build on concepts explored in previous chapters. The encoder used in the RTNet models builds on the multiscale architecture proposed in Chapter 4 but proposes modifications, including a linguistic embedding representation for silence. The explicit representation of silence is motivated by the desire to represent the natural segmentation (e.g. clausal segmentation) that occurs during speech. The inference network also revises how lexical features are represented by introducing an <UNSPEC> token that captures when the result of a VAD has detected that a new word is being spoken by the user but an ASR result has not yet be received. The motivation behind this token is to represent temporal information about the user's speech in the linguistic modality. In terms of publications that the architecture in this chapter is related to, RTNet-VAE's architecture is similar to VAEs with recurrent encoders and decoders proposed by Bowman et al. (2016), Ha and Eck (2018), and Roberts et al. (2018). Our use of a VAE to cluster dialogue acts is similar to the approach used in Zhao et al. (2017). Our vector-based representation of dialogue acts takes inspiration from the "attribute vectors" used by Roberts et al. (2018) for learning musical structure representations.

6.1.3 Our contributions

In this subsection we give a summary of the contributions of the work in this chapter.

- We present a model (RTNet) for generating the offset distributions of human-human conversations in an SDS. The model takes into account the context of both the system's response and the user's turn.
- We extend RTNet with a variational autoencoder (RTNet-VAE) that allows the encoder to be bypassed by directly sampling from a trained latent space. This allows the model to be integrated with an SDS pipeline more easily.
- We show how the trained latent space of RTNet-VAE can be exploited to analyse behaviours that the network has learned. This has potential use-cases in conversation analysis to perform preliminary analyses of turn-taking behaviours without the need for large amounts of manual annotations.
- We introduce a way of treating silence in continuous models by using a token-based representation, where silence is also given its own token and a corresponding embedding is learned. This treats silence like another word token and is motivated by the fact that pauses carry their own communicative importance.
- We introduce a linguistic representation of user speech in continuous models, where the user is known to be speaking but there is no result from the ASR yet. This allows the linguistic modelling component to benefit from the knowledge of when the user is speaking and temporal aspects that were missing from previously discussed continuous linguistic representations.
- We present offline evaluations of the models. Since evaluation of generative models is known to be difficult we present a customized method for evaluating generated distributions using the KL divergence of the empirical distributions.
- We present the results of listener tests that are designed to evaluate whether listeners consider some response timings more natural than others given a dialogue context. The results show that in many of the test cases listeners significantly preferred some response timings over others depending on the context. We also show that in instances where listeners are sensitive to response timings it is likely that our system will generate response timings that are more realistic than a system that simply generates the mode of the dataset.

6.2 Methodology

6.2.1 Dataset

Our dataset is extracted from the SWBD corpus (Godfrey and Holliman, 1993) (described in Section 2.4). Switchboard consists of 2438 dyadic telephone conversations with a total length of approximately 260 hours. We use the MS-State resegmentation (Deshmukh et al., 1998) for the orthographic transcriptions and word timings in preparing our dataset. The dataset consists of *turn pairs* (defined below) of adjacent turns by different speakers as shown in Fig. 6.2.

6.2.1.1 Turn Pairs

Turn pairs are automatically extracted from orthographic annotations using the following procedure. We extract frame-based speech-activity labels for each speaker using a step-size of 50ms. A frame is labelled 1 if there is any speech by the target speaker during any part of the frame and 0 otherwise. Non-verbal vocalizations such as laughter are also labelled 0. The frame-based representation is used to partition each person's speech signal into *IPUs*. We define IPUs as being segments of speech by a person that are separated by pauses of 200ms or greater. IPUs are then used to automatically extract *turns*, which we define here as consecutive IPUs by a speaker in which there is no speech by the other speaker in the silence between the IPUs. It is worth noting that this definition means that in most conversations there will not be an equal number of turns by each speaker since some IPUs will be in *full overlap* (where an IPU begins and ends during one of the interlocutor's IPUs). It is also worth noting that we do not make a distinction between backchannels and other turns as is frequently done (e.g. Levinson and Torreira (2015), Lala et al. (2017)). We purposely don't make this distinction so that we can allow the network to learn it itself. A turn pair is then defined as being any two adjacent turns by different speakers. The earlier of the two turns in a pair is considered to be the *user turn* and the second is considered to be the *system turn*. Turn pairs are similar to *adjacency pairs*, which were discussed and defined in Section 2.1.3. The pairs in our dataset differ from the quoted adjacency pair definition from Levinson (1983) in that they are not required to have the fourth property of being typed. The dataset includes not only adjacency pairs, but also pairs of adjacent utterances that are unrelated, or where their relations do not fit a standard pairing of dialogue acts.

One of our aims is to allow the network to learn patterns in pairings of adjacent utterances

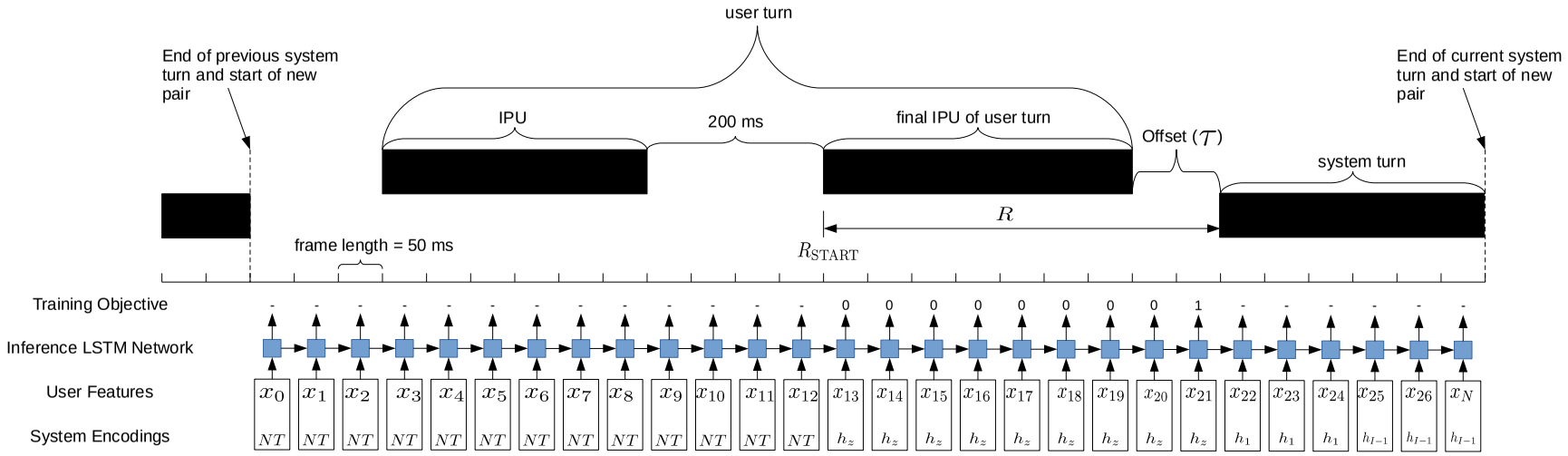


Figure 6.2: Diagram showing how the data is segmented into *turn pairs*, and how the inference LSTM makes predictions. The first turn in the turn pair is considered to be the *user turn* and the second turn is considered to be the *system turn*. The training objective is to minimize the loss between a sequence of zeros followed by a one and the output probability scores from the inference LSTM. The dashes (-) shown in the training objective represent frames that are excluded from the binary cross entropy (BCE) loss. The sequence of zeros begins at the R_{START} frame which has a minimum start time set at the start of the user's turn-final IPU. The objective sequence ends at the frame directly preceding the system turn start R_{END} and is labelled "one" to predict the start of the system turn in the following frame. In the system encodings, NT used to represent the <NONE> token, h_z is the response encoding, and h_n is used to represent the encodings of individual words in the system turn.

to generate realistic turn-switch offsets. We do not supply the network with labels for the types of adjacency pairs but rather allow the network to learn them itself. So, rather than having a limited hand-crafted lexicon of pairings, we hope to allow the network to learn standard and non-standard pairings. In section 6.4.5 we present evidence that the network is able to do this by clustering similar dialogue acts together in the latent space.

6.2.1.2 Training Objective

Our overall aim during training is to try to predict when the system turn will start. To do this, we try to predict the start of the system turn one frame ahead of the ground truth start time. The target labels in each turn pair is derived from the ground truth speech activity labels shown in Fig. 6.2. Each frame has a ground truth label $y \in \{0, 1\}$, which consists of the ground truth voice activity shifted to the left by one frame. As shown in the figure, we only include frames in the span R in our training loss. We define the span R as the frames from the beginning of the last IPU in the user turn to the frame immediately prior to the start of the system turn.

There are a number of factors as to why we include only these frames in our loss. As outlined in section 6.1.1, the target application for this system is for use with an incremental dialogue system where hypotheses of the continuation of the user turn can be made before the user is finished speaking. We view this system as being separate from a turn-taking system that makes decisions whether the system should take a turn or not. We assume that this decision to take the turn (or not) is handled by the DM and that the timing of the utterance is part of the NLG component. The DM decides *what* to deliver and the NLG module decides *how* to deliver it. Since the timing of an utterance has semantic significance, it is the role of the DM to output this semantic representation and then the role of the NLG to synthesize a response timing that is appropriate.

With this in mind, going back our data representation, we assume that pauses by the user in between IPUs that are not turn-final are pauses where the turn-taking component of the dialogue manager has decided not to take a turn. We assume that during these pauses the DM is intentionally waiting for the user to continue with their turn. We then assume that the necessary information for the system to formulate a response is supplied at some point during the turn-final IPU. As discussed in section 2.2.3, human listeners and incremental modules are often able to predict information about the continuation of a speaker's utterance. It is therefore reasonable that the system should be able to start formulating its response before the user is

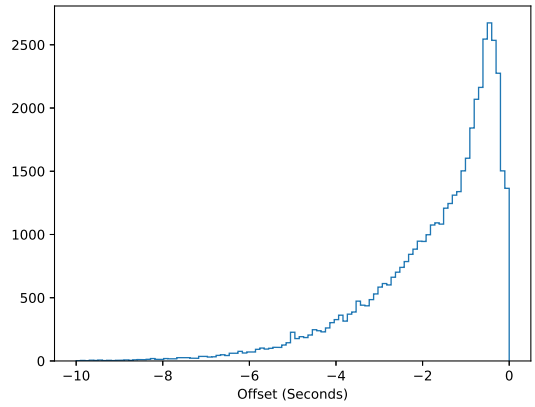


Figure 6.3: Histogram of the offsets from the start of user turn-final IPU to the start of the system turn.

finished speaking and during their turn-final IPU.

To simulate the assumption that the system response will be formulated at some point during the user's turn-final IPU (and any silence directly after it), we sample an index R_{START} from the span of R using a uniform distribution. We then use the reduced set of frames from R_{START} to R_{END} in the calculation of our loss. This is an important detail for several other reasons apart from our desire to simulate the system response formulation. Firstly, we found in informal experiments that when we used the full span of R without any randomization, the network was able to "cheat" by exploiting the distribution of turn-final IPU lengths (shown in Fig. 6.3). Without randomization it relied less on the input features and therefore captured less of the conversational behaviours that we wish to model. Randomization also improves the robustness of our network by effectively increasing the size of our dataset. While we assume that the distribution of R_{START} is given by a uniform distribution over the span of R, we acknowledge that this is a rough approximation of the true distribution that would be controlled by the SDS implementation. In reality, an addressee may begin formulating their response long before the start of turn-final IPU. An interesting avenue for further research would be to develop a more realistic model of this distribution using the dialogue context.

Another formulation of the training objective that was considered was to randomize R_{START} to be in within the span from the start of the user's turn-final IPU to the end of same IPU, rather than up to the ground truth system start time. However, if we use this objective, in the case of overlap this may result in a degenerate case where the ground truth system turn occurs before the sampled R_{START}.

An issue that arises if we increase the span of R to include more frames from before the start of the user's turn-final IPU is that the system is burdened with making turn-taking decisions that should be delegated to the turn-taking component of the DM. Since the system will encounter more pauses by the user it will be exposed to multiple transition relevance places (TRPs). For example, consider the situation where we wish to generate the timing of a system backchannel during a long stretch of user speech that includes many IPUs. There will be multiple pauses by the user, and (since the backchannel is likely to be generated somewhere in the vicinity of the user pauses) the distribution of likely positions for the backchannel will have multiple local maxima. Training the network in such a way results in instability since there are potentially multiple different "correct" predictions. It also results in generated distributions that are less meaningful since the system response from a turn pair could potentially be triggered long before the ground truth timing. It is for these reasons that we only include predictions during the turn-final IPU (with randomized R_{START}) in the calculation of our loss. We delegate the problem of deciding whether the system should speak or not to the DM.

6.2.2 Response Timing Network (RTNet)

As introduced in section 6.1.1, RTNet consists of an encoding network and an inference network that uses the response encodings, as well as acoustic and linguistic features from the user, to generate response timings. We will first describe the response encoder and the and then the details of the inference network.

6.2.2.1 Encoder

The encoder (shown in Fig. 6.4) fuses the acoustic and linguistic modalities from a system response using three bi-directional LSTMs. Similarly to the multiscale architecture proposed in Chapter 4, each modality is processed at independent timescales and then fused in a master Bi-LSTM which operates at the lexical temporal rate. The output of the master Bi-LSTM is a sequence of encodings $h_0, h_1, ...h_I$, where each encoding is a concatenation of the forward and backward hidden states of the Bi-LSTM at each word index. The linguistic Bi-LSTM takes as input the sequence of 300-dimensional embeddings of the tokenized system response. We use three special tokens in the linguistic Bi-LSTM that were not used in our original multiscale architecture: <SIL>, <WAIT>, and <NONE>. The <SIL> token is used whenever there is a gap between words that is greater than the frame-size (50ms). We introduce this silence token

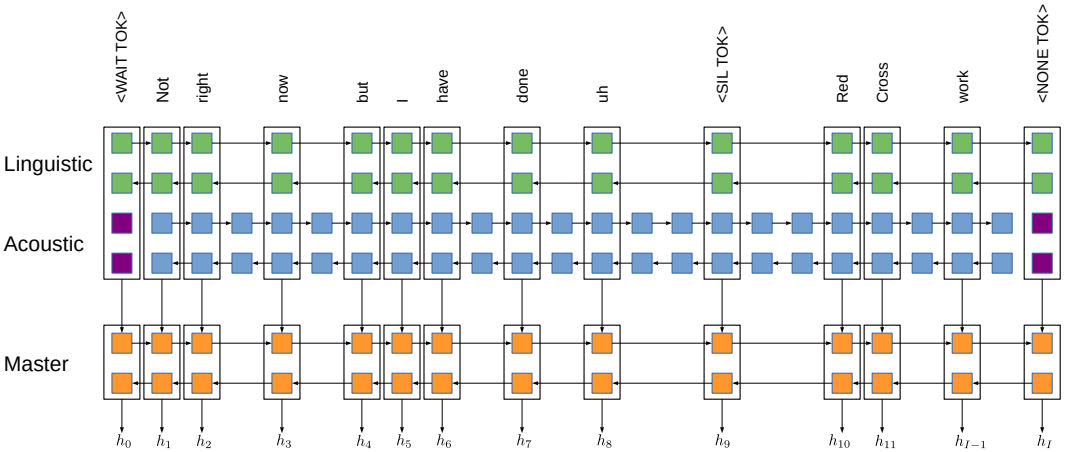


Figure 6.4: Response encoder network. The encoder consists of three stacked Bi-LSTMs: a linguistic LSTM, an acoustic LSTM, and a master LSTM that fuses the two modalities that operate at independent temporal granularities. The sequence of outputs from the acoustic LSTM is indexed at the time steps corresponding to the start frames of each word. These indexed acoustic hidden states are then concatenated with the hidden states of the linguistic LSTM and input into the master LSTM. We use special embeddings (shown in purple) to represent the acoustic hidden states corresponding to the first and last tokens (<WAIT> and <NONE>) of the system's turn.

to help the network learn clausal, syntactic, and temporal information about the response.

The <WAIT> and <NONE> tokens are inserted as the first and last tokens of the system response sequence respectively. The two tokens perform a similar functionality to the <EOS> and <SOS> tokens commonly used in encoder-decoder models (e.g. Prabhavalkar et al. (2017)). The concatenation $[h_0, h_1, h_I]$ is used to to produce the h_z encoding using a RELU layer as shown in Fig 6.5 (we refer to this layer as the *reduction layer*). The h_z encoding is used (along with user features) in the concatenated input to the inference network. It serves as a compact representation of the system's response that encodes the necessary information for the inference network to generate appropriate response timings. In the linguistic Bi-LSTM, the <WAIT> and <NONE> embeddings serve a slightly different purpose than traditional <EOS> and <SOS> embeddings. Since the <WAIT> embedding corresponds to the h_0 output of the master Bi-LSTM and the <NONE> embedding corresponds to h_I, the two embeddings serve as "triggering" symbols that allow the linguistic and master Bi-LSTM to know when to output relevant information accumulated in their cell states. In initial informal experiments we found that removing the <WAIT> and <STOP> tokens significantly degrades the performance.

The acoustic Bi-LSTM takes as input the sequence of acoustic features and outputs a sequence of hidden states at every 50ms frame. As shown in Fig. 6.4, we select the acoustic hidden states that correspond to the starting frame of each linguistic token and concatenate

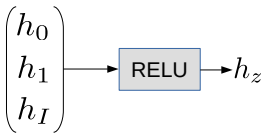


Figure 6.5: Reduction layer.

them with the linguistic hidden states. Due to the bi-directionality of the LSTM, the hidden state of the backward direction indexed at the frame that corresponds to the start of a word should contain information about the current word. The hidden state for the same frame in the forward direction should contain information about the previous word. When we supply the hidden states from these frames to the master Bi-LSTM we are supplying a representation of the acoustic properties of each word in the context of the whole utterance. The selection of these hidden states also increases the temporal granularity, allowing longer term dependencies to be modelled. Since there are no acoustic features available for the <WAIT> and <NONE> tokens, we train two embeddings that are the same size as the acoustic hidden states (shown in purple in Fig. 6.4). These embeddings are concatenated in the input to the master Bi-LSTM instead of acoustic hidden states.

6.2.2.2 Inference Network

The aim of our inference network is to predict a sequence of output probabilities $\boldsymbol{y} = [y_{R_{\text{START}}}, y_{R_{\text{START}}+1}, ..., y_N]$ using as input a response encoding h_z, and a sequence of user features $\boldsymbol{x} = [x_0, x_1, ..., x_N]$. We use a single-layer LSTM (shown in Fig. 6.2) which is followed by a sigmoid layer to produce the output probabilities:

$$[h_n; c_n] = \text{LSTM}_{\text{inf}}([x_n; h_z], [h_{n-1}; c_{n-1}]) \tag{6.1}$$

$$y_n = \boldsymbol{\sigma}(\boldsymbol{W}_h h_n + \boldsymbol{b}_h) \tag{6.2}$$

Since there are only two possible output values in a generated sequence (0 and 1), and the sequence ends once we predict 1, the inference network can be considered an autoregressive model where 0 is passed implicitly to the subsequent time-step. To generate an output sequence, we can sample from the distribution $p(y_n = 1 | y_{R_{\text{START}}} = 0, y_{R_{\text{START}}+1} = 0, ..., y_{n-1} = 0, X_{0:n}, h_z)$ using a Bernoulli random trial at each time-step. For frames prior to R_{START} the output is fixed to 0, since R_{START} is the point where the system has formulated the response. During training we minimize the BCE loss between our ground truth objective and our output predictions Y. We do not include frames prior to R_{START} in the calculation of our loss. For the frames

prior to R_{START} the inference network does not have access to the response encoding h_z so instead we use an embedding vector <NONE> (same size as h_z) that is jointly optimized with the rest of the network. For the frames after the response has been triggered, we concatenate the encoder hidden states for each of the words sequentially. Allowing the inference network to process frames before R_{START} allows it to collect potentially useful information from the user features in the inference LSTM cell state. During training the timings of the user's word encodings $(h_0, h_1, ..., h_I)$ are fixed to their original positions. During sampling, these timings are shifted by the difference between the sampled response offset and the true offset.

The inference network performs a similar function to a decoder in a standard encoder-decoder architecture (Cho et al., 2014; Sutskever et al., 2014). A standard encoder-decoder uses an input sequence to generate an output sequence that is not necessarily the same length as the input sequence. The encoder (typically a bi-directional RNN) processes the input sequence to produce encodings for each time-step h_n. These encodings are then used to generate the contextual representations that are used as inputs to the decoder. The distinction that can be made between standard encoder-decoder models and RTNet is that the output in RTNet is additionally conditioned on a second sequence (the user's speech features), the continuation of which is unknown at the point in time when the response is encoded. We are therefore conditioning on two separate time series: (1) the response which is considered to be known from R_{START} onwards, and (2) the user's input features which are processed on a causal frame-by-frame basis from the start of the turn pair. This means that the inference network must, on a frame-by-frame basis, generate the probability that the *subsequent* frame is appropriate for the response to to be triggered.

To do this the network must learn to respond to acoustic and linguistic turn-taking cues presented by the user, and then incorporate them with knowledge of the upcoming system response. For example, in cases where the system is about to respond to a user backchannel with a continuation of the system's previous turn, if we intend to simulate the offset distribution in human-human conversations, there should be a relatively high probability that there will be an overlap (Heeman and Lunsford, 2017). To be able to trigger the system response to occur *before* the user finishes speaking, the inference network must be able to do two things. Firstly, it must recognize that the user is uttering a backchannel. Many backchannels are simply one word (e.g. "uh-huh", "yeah"), therefore the inference network cannot rely on the user's linguistic modality, since the ASR result is only received 100ms *after* the the word is finished. It must be

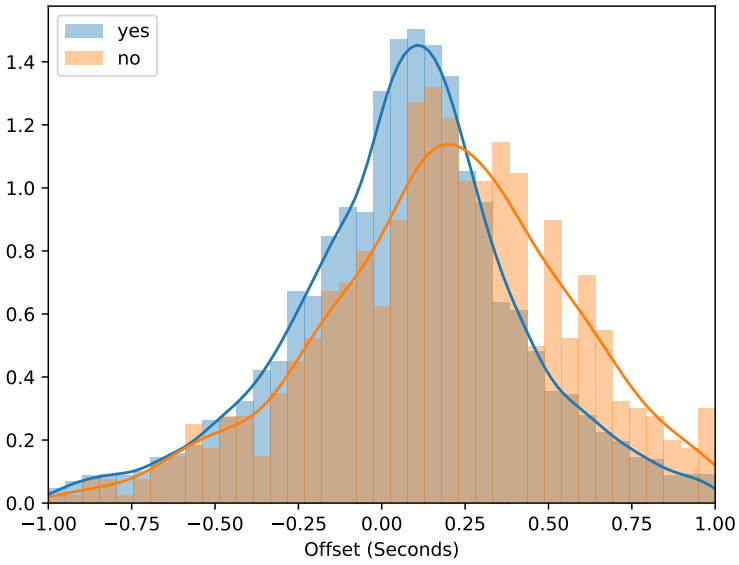


Figure 6.6: Ground truth distribution of "Yes" and "No" dialogue act offsets

able to predict that the user is in the process of uttering a backchannel (rather than taking a longer turn) based solely on the user's acoustic features. Secondly, it must be able to make this prediction *before* the end of the user's word. The sooner it is able to make this prediction, the better it will be able to model the "negative" tail of the offset distribution, which occurs before the user finishes speaking.

If we consider another example where, for instance, the system is responding to a user question with a "No" dialogue act, the distribution of "No" dialogue acts should have a modal offset that is greater than "Yes" offsets (shown in Fig. 6.6). To be able to model this distinction, the encoder must first be able to capture the knowledge that the response is a form of "No" response in the h_z encoding. The inference network must then not only model when the user is likely to stop speaking, but also any semantic aspects of the user's turn that may affect the response timing. The inference network must then incorporate the knowledge about the system response from h_z and model possible interactions between the system response and the user's turn. As a more concrete example (presented in Bögels et al. (2019)) of how the interactions between user turn and system response can affect response timings, if you invite a person to a party, a silence of one second or more might indicate that the response will not be the one you hoped for. Ideally, we would like the network to learn subtleties that model the wide variety of different types of "No" responses and how they function within multiple contexts. The one-word

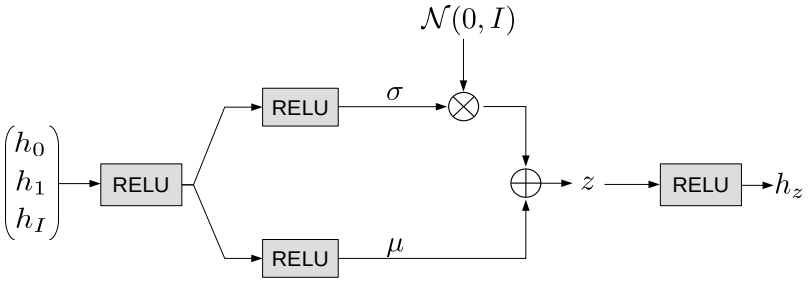


Figure 6.7: VAE

response "No." can potentially carry a wide variety of different pragmatic characteristics based on what context it is uttered in and its acoustic and prosodic characteristics. Accordingly, there should be an impact of these characteristics on the generated response timings.

6.2.3 RTNet-VAE

In this section we describe a variant of the RTNet architecture called RTNet-VAE that enables response timings to be generated more efficiently and allows simpler integration with SDS pipelines. The variant uses a variational autoencoder (VAE) to encode the system responses in a latent space that is interpretable and that we can use to efficiently generate response timings without using the response encoder. In this section we first describe the motivation behind RTNet-VAE and how it can be used within the SDS pipeline. We then describe the implementation details of how the VAE is incorporated into the previously described architecture. An examination of the latent space is presented later in Section 6.4.5.

6.2.3.1 Motivation

A limitation of RTNet as described in the previous section is that there are scenarios where it may be impractical to encode the full system turn before triggering a response. As discussed in 6.2.1.2, we assume that our network is part of the natural language generation (NLG) component in the SDS pipeline (Fig. 2.3). The NLG component will typically receive a semantic representation of the response from the dialogue manager (DM) such as *greet*, *inform*, *agree*, or a similar neural representation. If we wish to apply RTNet (without the VAE) directly on the generated responses, at run-time the RTNet component would have to wait for the full response to be generated by the NLG, which would result in a computational bottleneck. In an incremental system it may also be desirable for the system to start speaking before the entirety of the system response has been generated. These issues present problems for RTNet, which assumes

that the full system response is available during the response encoding stage. Additionally, the computational overhead of running the three stacked Bi-LSTMs is significant.

6.2.3.2 Latent Space

To address these issues, we bypass the encoding stage by directly using the semantic representation output from the DM to control the response timing encodings. We do this by replacing the reduction layer (Fig. 6.5) with a VAE (Fig. 6.7). To train the VAE, we use the same concatenation of encoder hidden states as was used in the RTNet reduction layer ($[h_0; h_1; h_I]$). We use a dimensionality reduction RELU layer to calculate h_{reduce}, which is then split into μ and $\hat{\sigma}$ components (both of dimensionality N_z) via two more RELU layers. $\hat{\sigma}$ is passed through an exponential function to produce σ, a non-negative standard deviation parameter. We sample the latent variable z with the standard VAE method (Kingma and Welling, 2014) using μ, σ, and a random vector from the standard normal distribution $\mathcal{N}(0, \boldsymbol{I})$ with unit variance (where $\boldsymbol{I}$ in this case is the identity matrix). A dimensionality expansion RELU layer is used to transform z into the response encoding h_z, which is the same dimensionality as the output of the encoder:

$$h_{\text{reduce}} = \text{RELU}(W_{\text{reduce}}[h_0; h_1; h_I] + b_{\text{reduce}}) \tag{6.3}$$

$$\mu = \text{RELU}(W_{\mu} h_{\text{reduce}} + b_{\mu}) \tag{6.4}$$

$$\hat{\sigma} = \text{RELU}(W_{\sigma} h_{\text{reduce}} + b_{\sigma}) \tag{6.5}$$

$$\sigma = \exp(\frac{\hat{\sigma}}{2}) \tag{6.6}$$

$$z = \mu + \sigma \odot \mathcal{N}(0, \boldsymbol{I}) \tag{6.7}$$

$$h_z = \text{RELU}(W_{\text{expand}} z + b_{\text{expand}}) \tag{6.8}$$

We impose a Gaussian prior over the latent space using a Kullback-Liebler (KL) divergence loss term:

$$L_{\text{KL}} = -\frac{1}{2N_z}(1 + \hat{\sigma} - \mu^2 - \exp(\hat{\sigma})) \tag{6.9}$$

The L_{KL} loss measures the divergence of the generated distribution from a Gaussian with zero mean and unit variance. L_{KL} is combined with the BCE loss using a weighted sum:

$$L = L_{\text{BCE}} + w_{\text{KL}} L_{\text{KL}} \tag{6.10}$$

As we increase the value of w_{KL} we increasingly enforce the Gaussian prior on the latent space. In doing so our aim is to learn a smooth latent space in which similar types of responses are organized in similar areas of the space.

After we train the latent representation, during inference we can skip the encoding stage of RTNet-VAE and sample z directly from the latent space on the basis of the input semantic representation from the DM. For example, if the DM supplies a *backchannel* semantic representation, we can use the distribution of previously observed (labelled) *backchannel* representations in the latent space to sample a z encoding. There are multiple possible methods that could be used to sample from the latent space. For example, a simple approach would be to sample from preexisting encodings of a given response-type i.e. if we would like to produce the timing for a backchannel, sample an encoding of a response that was already labelled as a backchannel. A second approach, which we examine in section 6.4.7, would be to approximate the distribution of latent variables for a given response-type *within* the latent space. For example, if we have a collection of labelled backchannel responses and their corresponding encodings we can approximate the distribution of $p(z|\mathrm{label} = backchannel)$ using an isotropic Gaussian by simply calculating $\mu_{backchannel}$ and $\sigma_{backchannel}$, the maximum likelihood mean and standard deviations of each of the z dimensions. This approach is explored in section 6.20. A third approach, also examined in section 6.4.7, would be (if the latent space is smooth enough) to calculate directions in the latent space that have different semantic characteristics and then interpolate between them. For example, we show that we can interpolate between the mean and variance parameters of *agree-accept* and *reject* response dialogue acts to generate an estimate for an intermediate "neutral" response. This approach could potentially allow greater flexibility in the response behaviours that are generated. It is worth noting that this interpolation approach makes the assumption that the *true* human response distribution of a combination of two distributions is indeed an interpolation of the two. This may not be the case. The true distribution of a neutral response may be completely different. However, in the absence of labels that can be used to generate the true distribution, the dimensional interpolation approach serves as a useful approximation.

To summarize, RTNet-VAE can be trained without dialogue act labels in a way that exploits linguistic and acoustic features of the response to construct the latent space. Then after training, any ground-truth dialogue act information that exists can be used to identify areas of the latent space to sample from. We propose that one of the advantages to using a VAE is that it avoids prescribing fixed dialogue act labels to responses and rather learns continuous representation of

responses in which response types are not allocated discrete labels. We argue that this can more adequately model how humans communicate since, in reality, the boundaries between response types are not always clear, utterances may have multiple communicative functions, and dialogue act taxonomies can potentially be arbitrarily complex (Bunt et al., 2010). Additionally, there may be paralinguistic inflections that affect the timing of responses that are not captured in purely linguistic representations.

6.2.4 Input Feature Representations

6.2.4.1 Acoustic Features

As acoustic features we use a combination of 40 log-Mel filterbanks, and 17 features from the eGeMAPs feature set (Eyben et al., 2016). The eGeMAPs features are all the features in the set excluding the MFCCs (e.g. pitch, intensity, spectral flux, jitter, etc.). All features are z-normalized on a per-file basis. All the features were extracted using a 50 ms frame-step.

6.2.4.2 Linguistic Features

For the linguistic features we use the word-level annotations from the ms-state transcriptions. These annotations give us the timing for the starts and ends of all words the words in the corpus. In total there are 30080 unique words in these annotations. As our feature representation we use 300 dimensional word embeddings that are initialized with GloVe vectors (Pennington et al., 2014) and then jointly optimized with the rest of the network. The same set of embeddings is used for both the system and user linguistic features, but we use separate linear layers to map the raw embeddings to the input features ($\mathbb{R}^{300} \mapsto \mathbb{R}^{300}$). The motivation behind inserting the linear layer is to aid the system in learning differences in functional properties of the embeddings for the user and system.

To prepare the embedding matrix we first tokenize the words using the Spacy toolkit (Honnibal and Montani, 2017). Since many of the words in the raw corpus annotations contain multiple tokens (e.g. "Let's" gets tokenized to ["Let", "'s"]) we create an initial embedding matrix of size 30080x300 by averaging the pre-trained GloVe vectors in each unique word. Many of the words in the corpus have low count numbers making it difficult to learn useful embeddings. To address this issue we then reduce the embedding dimension down to 10000x300 by merging embeddings that have low word counts with the nearest embedding in vector space. The merge is performed using cosine similarity distance.

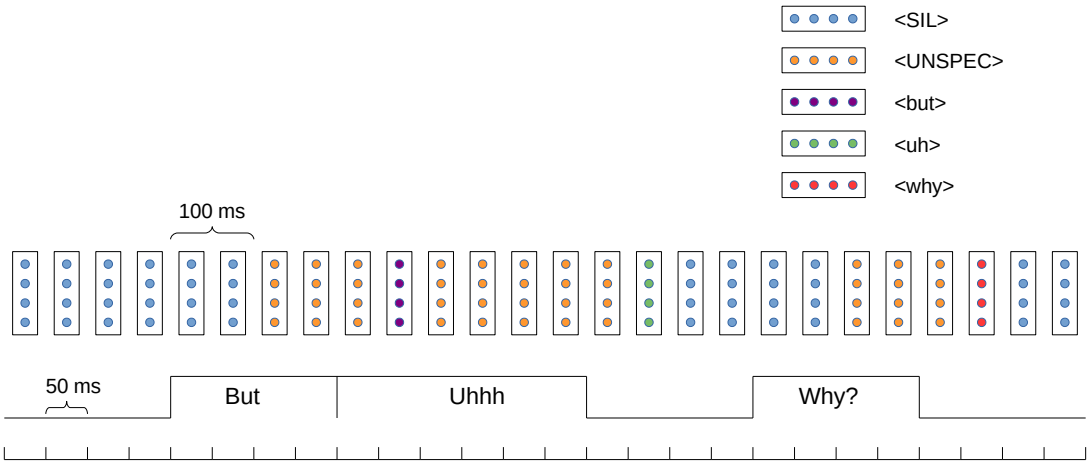


Figure 6.8: The user's linguistic feature representation scheme. The embedding for each word is triggered 100 ms after the ground truth end of the word, in order to simulate ASR delay. The <UNSPEC> embedding is used to represent information about whether a word is being spoken (before it has been recognized) and the length of each word. The <UNSPEC> embedding begins 100ms after a word's start frame.

We also introduce four additional tokens that are specific to our task. The <SIL>, <WAIT>, and <NONE> tokens were discussed previously in section 6.2.2.1. The fourth additional token that we introduce is an unspecified token <UNSPEC>. This token was introduced to represent temporal information in the linguistic embeddings. The way the token is used is shown in Fig. 6.8. As we have done in the previous four chapters, we approximate the processing delay in ASR by delaying the annotation by 100ms after the ground truth frame where the user's word ended. However, we can improve upon representations that simply trigger 100 ms after the speaker has finished their word by including information about when the word started as well. Since voice activity detection (VAD) can typically supply a reliable estimate of when when a word has started, we can use this information to supply the network with the <UNSPEC> embedding 100 ms after the word has started, as shown in Fig. 6.8. Taking as an example the first word "But" that lasts for four frames in the ground truth annotations, for the user's linguistic feature representation this results in three frames of <UNSPEC> embeddings and one frame of a <but> embedding, all of which are shifted by 100ms. This allows the the network to learn temporal information about the user's linguistic features that was previously missing from other continuous systems.

6.3 Experiments

6.3.1 Model Evaluation

6.3.1.1 Generative distance ($\mathrm{KL}_{\mathrm{hist}}$)

For many types of generative models, the question of how to accurately evaluate models is an unsolved issue (Theis et al., 2016). Often the loss that we are minimizing does not offer a reliable way of comparing models and does not guarantee the quality of generated samples. Typically automatic (rather than human-rated) evaluation procedures tend to be domain-specific (e.g. inception score (Salimans et al., 2016) or BLEU (Papineni et al., 2001)) Due to the expense involved in human evaluations, there is often little choice but to resort to automatic ways of evaluation that may not correlate well with human evaluation (Liu et al., 2016).

In our case, a comparatively low BCE loss does not necessarily guarantee that the generated offset distribution of one model will be closer to the target offset distribution than another. To more adequately evaluate the generated distributions, we propose a domain-specific distance measure based KL divergence that more directly compares the generated distribution with the empirical distribution. We calculate two normalized histograms, one using the true offsets and the other using the generated offsets, with 60 bins at 50ms intervals between -1.5 seconds to 1.5 seconds. The histogram KL divergence is calculated using:

$$\mathrm{KL}_{\mathrm{hist}} = D_{\mathrm{KL}}(c^t || c^g) = \sum_b^B c_b^t \log(\frac{c_b^t}{c_b^g}) \tag{6.11}$$

where c^t and c^g are the normalized histograms of the empirical and generated distributions which are indexed by the bin index b. $\mathrm{KL}_{\mathrm{hist}}$ measures how different the generated offset distribution is from the ground truth empirical distribution, within a limited interval. We choose the interval [-1.5, 1.5] to exclude the tails of the distributions which can be unreliable and can cause the output to fluctuate, especially when using smaller sample sizes.

6.3.1.2 Discriminative Metrics

We also report L_{BCE} and L_{KL} for our validation and test sets. These are calculated independently from $\mathrm{KL}_{\mathrm{hist}}$ which requires sampling.

Frame Step-size	50 ms
Acoustic Features	57 Acoustic features: 40 F-Banks, 17 features from GeMAPs
Linguistic Features	Ling Embeddings $\in \mathbb{R}^{10004\times300}$ 2 x LINEAR($\mathbb{R}^{300} \mapsto \mathbb{R}^{300}$), one for user, one for system.
Encoder Network	Acous Bi-LSTM($\mathbb{R}^{57} \mapsto \mathbb{R}^{256}$) Ling Bi-LSTM($\mathbb{R}^{300} \mapsto \mathbb{R}^{256}$) Master Bi-LSTM($\mathbb{R}^{512} \mapsto \mathbb{R}^{512}$) 2 Acous embeddings $\in \mathbb{R}^{2\times256}$
With VAE	Latent Variable Size: $z \in \mathbb{R}^4$ Reduction layer: RELU($\mathbb{R}^{1536} \mapsto \mathbb{R}^{256}$) VAE μ layer: RELU($\mathbb{R}^{256} \mapsto \mathbb{R}^4$) VAE σ layer: RELU($\mathbb{R}^{256} \mapsto \mathbb{R}^4$) Expansion layer: RELU($\mathbb{R}^4 \mapsto \mathbb{R}^{512}$)
Without VAE	Reduction layer: RELU($\mathbb{R}^{1536} \mapsto \mathbb{R}^{512}$)
Inference Network	LSTM Hidden Size: 1024 Output Layer: SIGMOID($\mathbb{R}^{1024} \mapsto \mathbb{R}^1$) <NONE> embedding $\in \mathbb{R}^{1\times512}$
Training Settings	Batch Size= 64; L2 regularization = 1e-7 Optimizer=Adam(betas=0.9 and 0.999) Initial LR=1e-4 LR reduction factor=0.1 on plateau; Patience=10; $R_{\text{MIN (TRAIN)}}$= Start of users's turn-final IPU R_{END}= Ground truth system start time $R_{\text{START}} \sim$ UNIFORM([$R_{\text{MIN (TRAIN)}}$,R_{END}])
Test Settings (Discriminative)	R_{START}= $R_{\text{MIN (TRAIN)}}$
Test Settings (Generative)	Default R_{MAX}= 2 seconds (40 frames) $R_{\text{MIN (TEST)}}$= max([R_{END} - R_{MAX}, $R_{\text{MIN (TRAIN)}}$]) R_{START} ~UNIFORM([$R_{\text{MIN (TEST)}}$, R_{END}]) Appended frames with simulated silence=80 (4 seconds); Temperature=1.0

Table 6.1: Hyperparameter Settings

6.3.2 Training and Testing Procedures

Table 6.1 shows an overview of the default hyperparameter settings used for training and testing our models. The training, validation, and test sets consist of 1732, 64, 642 conversations respectively with 159519, 5986 and 58783 turn pairs. The test set is deliberately chosen to include all of the conversations from the NXT-format annotations (Calhoun et al., 2010). The NXT-format annotations (discussed in Section 2.4) serve as a useful link between two different annotation efforts, the MS-state orthographic annotations of the original switchboard corpus (Godfrey et al., 1992) and the Switchboard Dialog Act Corpus (SWDA) (Stolcke et al., 2000). The NXT annotations address many of the alignment issues that are involved in merging the timing information in the MS-state annotations and dialogue act annotations. We include the entirety of the NXT annotations in our test set so that we have enough labelled dialogue act samples to be able analyse the offset distributions.

We also note that for many turns there may be multiple dialogue act labels. For example, a turn may start with a *yes* DA and then continue with a *statement.* In our analysis of our model's performance on different types of dialogue acts in sections 6.4.1, we label a system response as belonging to a certain dialogue act if it begins with that dialogue act. We label a user turn as belonging to a certain class of dialogue act if it ends with that dialogue act. A consequence of this labelling scheme is that the count for each dialogue act response in our dataset is lower than the total count in the full annotations. We also exclude turn pairs where the system response occurs in full overlap (starts and ends during the same system IPU) in our analysis of dialogue acts. However, these turn pairs with responses in full overlap are included in the training set.

While we found that randomizing R_{START} during training was important for the reasons given in section 6.2.1.2, it presented issues for the stability and reproducibility of our evaluation and test results. We therefore randomize during training and when we are sampling to calculate $\mathrm{KL}_{\mathrm{hist}}$, but during the calculation of discriminative test results (reported in table 6.2) we fix R_{START} to be the first frame of the user's turn-final IPU.

During sampling, it is also necessary to increase the length of the turn pair since the response time may be triggered by the sampling process *after* the ground truth time. We must therefore pad the user's features with extra frames in a way that resembles silence. Since different recordings will have different noise floors and different types of background noise, we artificially generate silence by locating segments of silence in each file. We calculate a mean vector and a covariance matrix of the user's acoustic features during the silence segments. We use the mean

vector and covariance matrix to randomly generate four seconds (80 frames) of silence that is specific to each file and append it to each of the user's acoustic feature sequence. We consider this approach to be a better alternative to zero-padding since for many of the acoustic features, zero does not represent silence. Similar approaches to representing background noise (where the probability distribution of the input features is modelled using Gaussians) can be found in the literature (Ephraim and Malah, 1984; Hirsch and Ehrlicher, 1995; Jongseo Sohn and Wonyong Sung, 1998; Cohen et al., 2009). Noise reduction systems for speech commonly represent background noise statistics as covariance matrices of the short time Fourier transform (STFT) bins (Cohen et al., 2009; Hirsch and Ehrlicher, 1995). It is typical for these covariance matrices to be estimated directly from silence segments in the speech signal using a voice activity detector (VAD) (e.g. Cohen (2003)). It is also worth noting that, during sampling, when a 1 is generated at a given frame, then the offset is calculated from the frame that directly follows this frame. This is done to account for the fact that, in a real-world implementation, actions based on the output of a given frame could only occur in the subsequent frame.

Since the sampling process introduces randomness, we found that the $\mathrm{KL}_{\mathrm{hist}}$ evaluation metric could sometimes fluctuate. To account for this, for each of the experiments we performed the generative evaluation sampling procedure five times. We report the average of $\mathrm{KL}_{\mathrm{hist}}$ for the five trials. The BCE loss evaluation metrics are deterministic so there was no need to do the same for them.

A training detail that was found to improve results was to concatenate two turn adjacent turn pairs together so that the hidden state of the inference LSTM is maintained from the first to the second pair. This effectively doubles the batch size. It also allows the network to incorporate contextual information from the first pair into the predictions for the second pair. It also allows the network to backpropagate from the first pair to the second. The ability to incorporate context from one pair to the next is one of the aspects we wished to investigate. The concatenation of the two allows us to still randomize the batching while incorporating context at the same time (further discussed in section 6.4.3).

6.3.3 Comparison model: Best fixed probability

To the best of our knowledge, there aren't any other published models that we can directly compare ours to. However, we can calculate the best performance that can be achieved using a

		Discriminative				Generative
		L_{BCE} Test	L_{KL} Test	L_{BCE} Valid	L_{KL}Valid	KL_{hist}
Full Model (RTNet without VAE)						
1	Full Model	**0.11177**	–	**0.11312**	–	0.0323
Fixed Probability						
2	Fixed Probability	0.13046	–	0.13115	–	0.2756
RTNet Encoder Ablation						
3	No Encoder	0.12329	–	0.12355	–	0.0372
4	Only Acoustic	0.11438	–	0.11511	–	0.0357
5	Only Linguistic	0.11577	–	0.11652	–	0.0359
RTNet Inference Network Ablation						
6	Only Acoustic	0.11327	–	0.11476	–	0.0368
7	Only Linguistic	0.11776	–	0.11862	–	0.0531
8	Test with context	0.11215	–	0.11374	–	0.0754
9	Train with context	0.11566	–	0.11613	–	0.0485
10	Without pair concatenation	0.11365	–	0.11414	–	0.0384
11	Next Token Prediction	0.11480	–	0.11503	–	0.0412
RTNet-VAE Experiments						
12	$w_{KL} = 0.0$	0.11217	4.10175	0.11339	4.09478	0.0341
13	$w_{KL} = 10^{-4}$	0.11230	2.44277	0.11366	2.42803	0.0326
14	$w_{KL} = 10^{-3}$	0.11354	0.87713	0.11459	0.86906	**0.0322**
15	$w_{KL} = 10^{-2}$	0.11997	0.00003	0.12058	0.00003	0.0388
16	$w_{KL} = 10^{-1}$	0.12021	**0.00000**	0.12085	**0.00000**	0.0388

Table 6.2: This table shows the results of 16 different experiments that investigate ablations and parameter settings. The experiments are evaluated using the metrics described in section 6.3.1. The discriminative metrics (L_{BCE} and L_{KL}) calculated for both the validation and test sets, while the generative metric (KL_{hist}) is calculated on the test set. Lower is better in all cases, and the best result for a given metric column is shown in bold. The L_{KL} metric is only available for the RTNet-VAE experiments since this metric is only calculated when a VAE is being used.

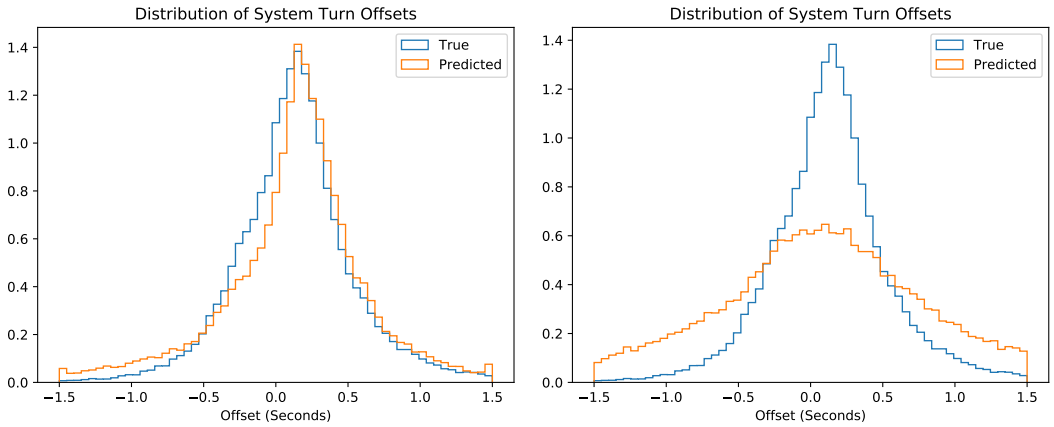
fixed value for y. The best possible fixed y for a given turn pair is:

$$y_{\text{tp}} = \frac{1}{(R_{\text{END}} - R_{\text{START}})/\text{FrameLength}} \tag{6.12}$$

The best fixed y for a set of turn pairs is given by the expected value of y_{tp} in that set:

$$y_{\text{fixed}} = \mathbb{E}[y_{\text{tp}}] \tag{6.13}$$

This represents the best performance that we could achieve if we did not have access to any user or system features. We can use the fixed probability model to put the performance of the rest of our models into context.



(a) RTNet (full model) generated distribution (b) Fixed probability (random) generated distribution

Figure 6.9: Generated offset distributions for the full test set using the full model and the fixed probability (random) model

6.4 Analysis of Performance

In this section we examine the performance of the different components of our two proposed models (RTNet and RTNet-VAE). In section 6.4.1 we look at our baseline performance of the full RTNet model and the generated offset distributions. In section 6.4.2 we examine the encoder of RTNet through an ablation study in which we examine the impact of removing the decoder and the effect of each of the input modalities. In section 6.4.3 we perform a similar ablation for the inference network, as well as investigate the impact of context. In section 6.4.4 we look at the performance of RTNet-VAE. In section 6.4.5 we investigate the trained latent space of RTNet-VAE. In section 6.4.6 we present a way that information about response timings encapsulated in the trained latent space can be exploited for conversational analysis. In section 6.4.7 we investigate methods of sampling from the latent space and show how it would be possible to create a dimensional model of dialogue acts through interpolation. In section 6.5 we present results from the human evaluation study.

6.4.1 Baseline Performance

The offset distribution for the full RTNet model is shown in Fig. 6.9a. If we compare this with Fig. 6.9b, which shows the distribution of predicted offsets using the best possible fixed probability, we can see that the baseline RTNet model is able to replicate many of the features of the true distribution: the mode is correctly generated at the bin between 150-200 ms, the generated offsets have similar kurtosis and skew as the true distribution, and for most parts of

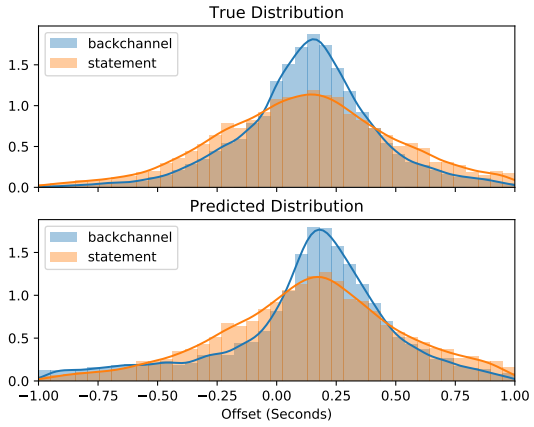


Figure 6.10: RTNet (full model) generated distribution for turn pairs where the *system turn* was either a backchannel or a statement

generated distribution it closely follows the target distribution. Using a fixed probability results in a similar mode, but the distribution has a much higher kurtosis than the true distribution and deviates from the target distribution throughout. Table 6.2 shows our experimental results for the different network configurations that were examined. The differences between the baseline and the fixed probability distributions are reflected in rows one and two of Table 6.2. The full model has the lowest discriminative losses and a low $\mathrm{KL}_{\mathrm{hist}}$value while the fixed probability model has the highest of the BCE losses and the highest $\mathrm{KL}_{\mathrm{hist}}$

Going back to the baseline distribution in Fig. 6.9a, we can see that the part of the distribution that RTNet has the most trouble reproducing is the area between the offsets of -500 ms and 0 ms. This part of the distribution is the most demanding part because it requires that the model anticipate the user's turn-ending. From the plots it is clear that our model is able to do that to a large degree, however not quite as well as humans are able to. We found this area of the distribution to consistently be the hardest part to reproduce. We observe that after the user has stopped speaking (from 0 seconds onward) the generated distribution follows the true distribution very closely.

To look in more detail at how well the system models the offset distribution we can investigate the generated distributions of different labelled dialogue acts. Fig. 6.10 shows a comparison of the generated distributions when the *system* turn starts with a *backchannel* or *statement* dialogue acts. Fig 6.11 shows the same but for when the *user* turn ends with either a *backchannel* or *statement*. We show the equivalent fixed probability generated distributions in Figures 6.12 and 6.13. We can see that in the case distributions generated with the baseline model, they closely

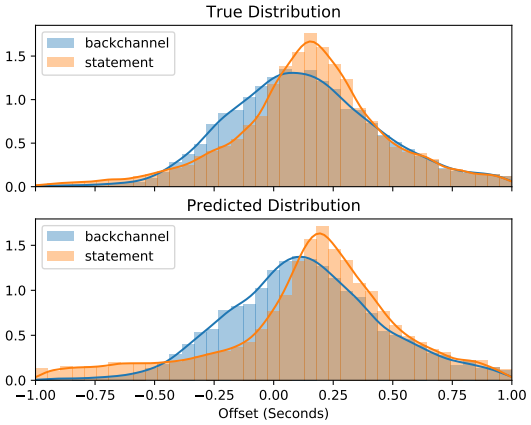


Figure 6.11: RTNet (full model) generated distribution for turn pairs where the *user turn* was either a backchannel or a statement

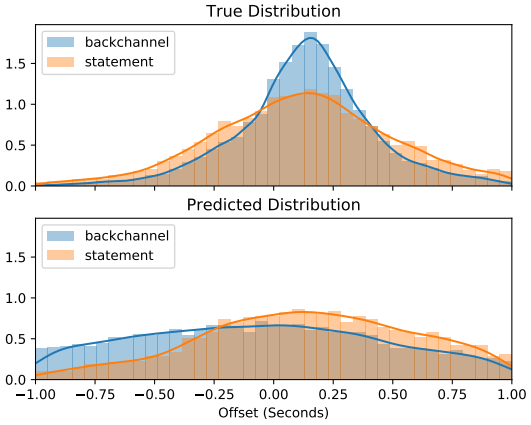


Figure 6.12: Fixed probability (random) generated distribution for turn pairs where the *system turn* was either a backchannel or a statement

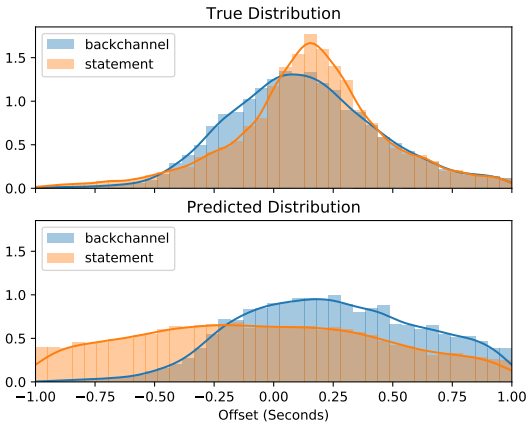


Figure 6.13: Fixed probability (random) generated distribution for turn pairs where the *user turn* was either a backchannel or a statement

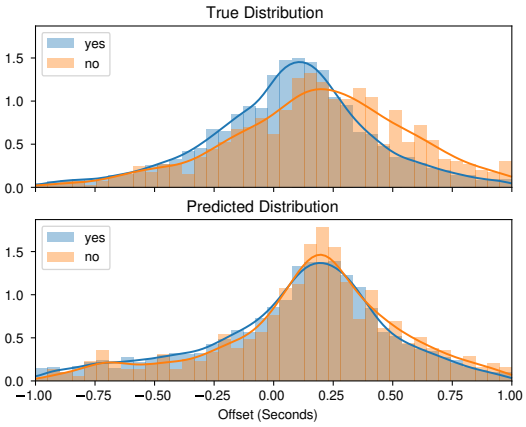


Figure 6.14: RTNet: Generated distribution *without* an encoder for turn pairs where the *system turn* was either a "Yes" or "No"

follow the true distributions. When the system turn starts with a backchannel or statement, the modes of the true distributions are similar (around 200ms) but *statements* have a higher kurtosis. These features are captured by the generated distribution but not by the fixed probability distribution. In the case of the *user* turn ending with a *backchannel* or *statement*, the mode of the offsets for backchannels is earlier than the mode for statements. This behaviour has been observed in previous analyses of backchannels (Levinson and Torreira, 2015). We can see that the generated distribution is able to model the difference in modes as well as the difference in kurtosis of the two distributions.

6.4.2 Encoder Performance

The performance of the response encoder was analysed in an ablation study, the results of which are in rows three through five of Table 6.2. Without the response encoder, the BCE losses are much higher than with any of the models where the encoder is included. From looking at the encoders with only acoustic and linguistic modalities, we can see that the losses benefit more from the acoustic modality than the linguistic modality. However, if we look at the generative distance ($\mathrm{KL}_{\mathrm{hist}}$), the decrease in performance relative to the baseline is -10.5% for only acoustic features, -11.1% for the only linguistic features, and -15.1% for having no encoder. Since the relative improvement gained by the inclusion of both linguistic and acoustic features is greater than its parts, this suggests that there is an interaction between the acoustic and linguistic features in the response encoder that increases performance.

If we consider the impact of the encoder in more detail, we would expect that the network

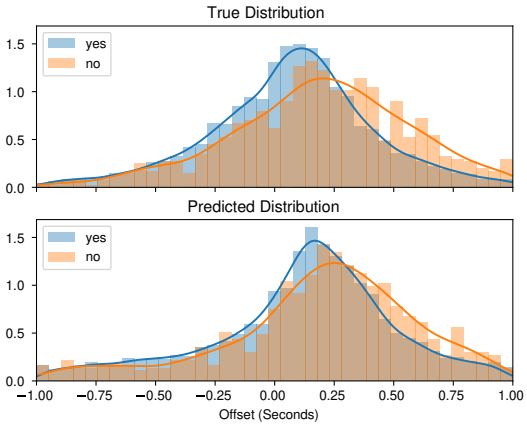


Figure 6.15: RTNet: Generated distribution *with* an encoder for turn pairs where the *system turn* was either a "Yes" or "No"

would not be able to model distributional differences between different types of DA responses without an encoder. In Figures 6.14 and 6.15 we show the generated distributions for *yes* and *no* responses for models without and with the encoder. In the true distributions we can see that the true *no* distribution should have a larger modal offset than the true *yes* distribution and that the *no* distribution should have a larger probability mass from roughly 250ms onward. In Fig. 6.14 we can see that without the encoder, the distributions of the *yes* and *no* offsets are almost exactly the same. In Fig. 6.15 with the introduction of the encoder, we can see that the network is able to model the differences between the modes of the *yes* and *no* offsets, as well as model the larger probability mass from 250ms onward. It is worth observing that the part of the generated distribution where the full model has trouble making a distinction between the two DAs is the segment between roughly -500 ms to 0 ms. This is the same part of the distribution that was identified in the previous section as being difficult to model.

While the encoder is necessary for modelling differences between responses such as *yes* and *no* response DAs, we found that in some cases, there was enough information in the *user's* features to be able to partially predict the distribution of system response offsets. For example, Fig. 6.16 shows the generated distribution of *statement* and *backchannel* system responses without using an encoder. We can see that even without the encoder the generated distributions are able to model some of the differences between the two DAs, although the approximation is not as precise as with the encoder (shown in Fig. 6.10). We propose that a possible reason for the system being able to model the distinctions is due to backchannels being elicited by the user through linguistic and acoustic cues (Morency et al., 2010). Therefore the timing of backchannels can

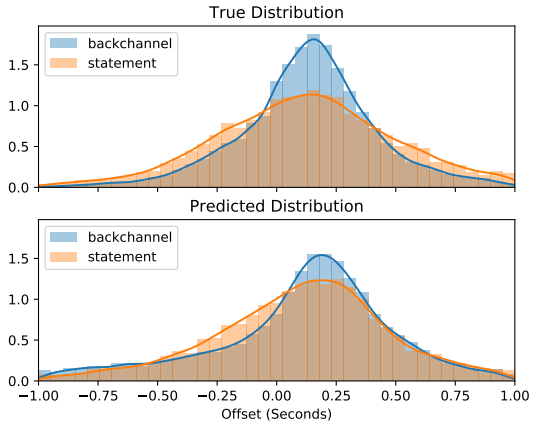


Figure 6.16: RTNet: Generated distribution *without* an encoder for turn pairs where the *system turn* was either a backchannel or a statement

be, at least partially, attributed to the user. In cases such as these, the timing of a response could be predicted if the response *itself* can be predicted on the basis of the context. While this is a possible explanation for our system being able to model the distributional differences without an encoder, a more thorough analysis would be necessary to draw conclusive results.

6.4.3 Inference Network Performance

In rows six to 11 of Table 6.2 we present an ablation of the inference network as well as an examination of some alternative objective functions. Looking at rows six and seven, we can see that removing either the acoustic or linguistic features from the user's features is detrimental to the performance. Removing the acoustic features affects the performance much more than removing the linguistic features. This is what we would expect, since we delay the output of linguistic features by 100ms to account for ASR making, it is harder for the model to predict the endpoint of a user's turn. The model's main reliance on acoustic features in the inference network is in accordance with previous results from other continuous systems (e.g. Skantze (2017b)). It is also worth noting that linguistic information is relatively of much greater importance in the encoder than it is in the inference network. This could potentially have repercussions for practical implementations where, in some cases, it may be possible to only rely on acoustic features and avoid waiting for ASR results without impacting the overall performance.

In rows seven and eight of Table 6.2 we investigate the impact of maintaining context across turn pairs by passing the hidden state of the inference LSTM from one pair to the next. We hypothesized that maintaining context would increase the performance of our models. Our rea-

soning was that if we passed the hidden state of the inference network between pairs, the network could learn how to exploit long term conversational features such as individual's personality characteristics or interpersonal synchrony to predict the response timings. However, we found that using the context actually resulted in worse performance in both cases.

In row eight we show the results from testing our baseline model by passing the hidden state across turn pairs. We thought that, since the baseline model was trained using two concatenated turn pairs, it would be able to exploit the contextual information in a useful way. However, the results show that this was not the case and the performance was worse. In row nine we show the results from an experiment where we tried training our model by passing context across turn pairs. This also resulted in considerably worse performance. We note that under this training scheme, we cannot benefit from fully randomizing the batching since the pairs from a given conversation must be trained on sequentially. We found that removing the randomized batching introduced instability into the training.

In row 10 we show the effects of using only a single turn pair during training rather than concatenating two pairs together as discussed in section 6.3.2. The results confirm that concatenating two pairs increases the performance considerably. While our initial intention in concatenating the two pairs was to incorporate context, the fact that testing with context degrades the performance implies that the performance gains from concatenation may be attributed to something else. The gains may be due to the effective doubling of the batch size or a regularizing effect but the exact cause is uncertain.

In row 11 we show the results from an experiment where, rather than just predicting the probability of the start of the system response in our training loss, we predicted the timing of *each token* in the system response. Our motivation for this experiment was that we hypothesized that the prediction of the timings of the other tokens in the response sequence would help the network learn sequential information about the linguistic features, in a similar manner to traditional LSTM language models (Mikolov et al., 2010). However, the results showed that including these predictions in our loss had a negative impact on our response timing predictions. When we investigated the predictions we found that the decoder was able memorize the timings of the different words within a system response almost precisely. We therefore conclude that the decrease in performance was due to the model focussing on optimizing this memorization rather than learning any useful lexical patterns.

6.4.4 RTNet-VAE Performance

In rows 12 through 16 of table 6.2 we show the results of our experiments with RTNet-VAE, where we introduce a VAE into the response encoder pipeline. The experiments show the results for different settings of the w_{KL} loss weight. We can see that as we increase the value of w_{KL} the L_{BCE} loss increases while the L_{KL} loss decreases. This trade-off is between the distance of the distribution of z from a unit variance Gaussian, and how much the generated y probabilities deviate from the ground truth. However, if we look at the values in our generative evaluation $\mathrm{KL}_{\mathrm{hist}}$ we can see that the w_{KL} for the lowest L_{BCE} ($w_{\mathrm{KL}} = 0.0$) is not the same as the one for the lowest $\mathrm{KL}_{\mathrm{hist}}$ ($w_{\mathrm{KL}} = 10^{-3}$). We also observe that our best performing $\mathrm{KL}_{\mathrm{hist}}$ for the VAE experiments is our best overall value for this distance ($\mathrm{KL}_{\mathrm{hist}} = 0.0322$), but is almost exactly the same as the $\mathrm{KL}_{\mathrm{hist}}$ for the full model ($\mathrm{KL}_{\mathrm{hist}} = 0.0323$). In the experiments where we increased or decreased w_{KL} we found that the $\mathrm{KL}_{\mathrm{hist}}$ performance got worse.

This implies that the response timing predictions benefit from a certain amount of the Gaussian prior imposed on z. We propose that the Gaussian prior helps organize the latent space in a semantically useful way that is more robust. If the Gaussian prior is enforced too much it constrains the expressiveness of z. If it is not constrained enough then the latent space is less robust. The fact that the $\mathrm{KL}_{\mathrm{hist}}$ value for RTNet-VAE is very close to the best achieved with RTNet suggests that the impact of the VAE on the overall RTNet performance is minimal. If we look at some example distributions of dialogue acts generated by RTNet-VAE (shown in figures 6.17 and 6.18) we can see that RTNet-VAE is capable of generating distributions that are of a similar quality to those generated by RTNet (shown in figures 6.10 and 6.15).

6.4.5 Latent Space Analysis

In Fig. 6.19 we show the latent variable z generated using RTNet-VAE and plotted using t-SNE (van der Maaten and Hinton, 2008) for dimensionality reduction. The plot shows the two-dimensional projection of nine different types of dialogue acts. We randomly select 200 different examples of each dialogue act and plot the projection. We can see that the latent space is able to organize the responses by dialogue act types even though it is never trained on dialogue act labels. For example we can see that *backchannels* (shown in orange) are clustered to the far right while *yes* responses (shown in brown) are distributed mostly over the right half of the plot but with a larger spread than than the backchannels. We can infer that the network has learned that many *yes* responses are closer to *backchannels* than, for example, *opinions* (shown in green) or

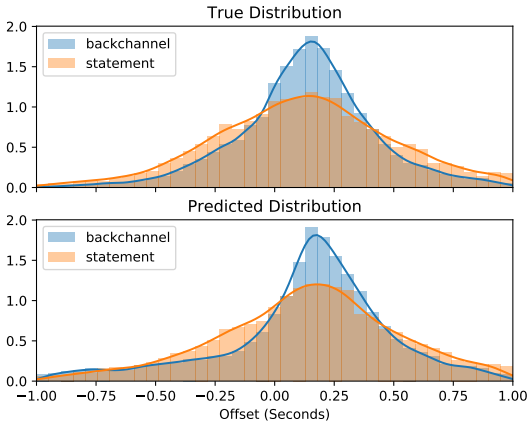


Figure 6.17: RTNet-VAE: Generated distribution for turn pairs where the *system turn* was either a backchannel or a statement

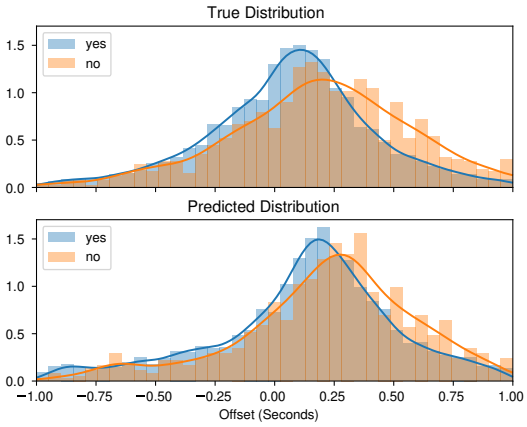


Figure 6.18: RTNet-VAE: Generated distribution for turn pairs where the *system turn* was either a "Yes" or a "No"

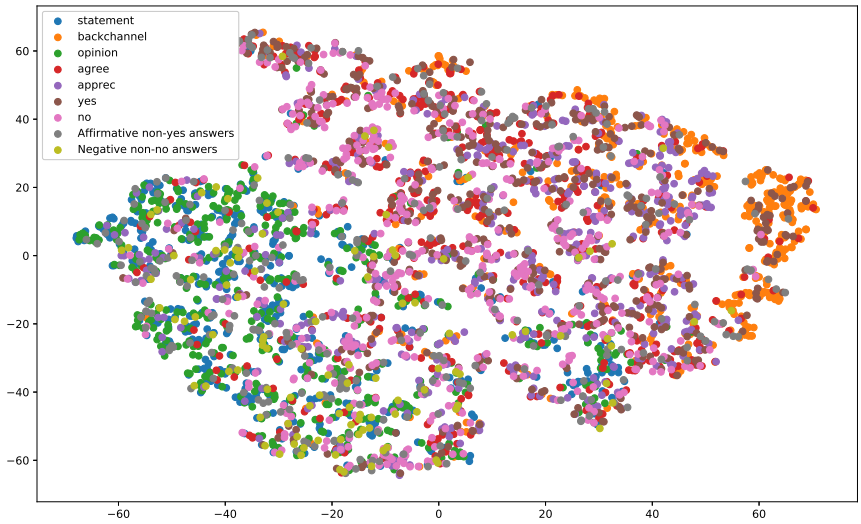


Figure 6.19: TSNE of 9 types of dialogue acts

statements (shown in blue), at least for the purpose of response timing generation. Intuitively this makes sense, since we can imagine that many *yes* responses would have a similar semantic characteristics as backchannels (e.g. A:"You know what I mean?", B:"Yes"). Others may be more similar to statements (e.g. A:"Do you think she will win the election?", B:"Yes, I think she will win the election").

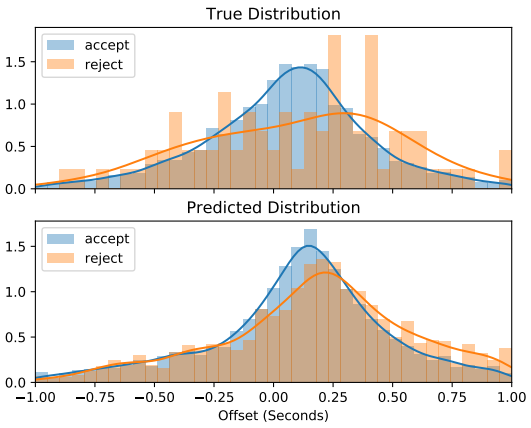


Figure 6.20: Synthetic generation of the offset distributions of accept and reject dialogue acts.

6.4.6 Conversation Analysis (CA) using the Latent Space

In this section we show how human behavioural information captured in the trained network weights can be analysed to gain insights for the purposes of conversational analysis. The method exploits the trained network's ability to generate the distribution of z for a given response type. We use this to analyse the response timing offset distributions of those dialogue act types which for we only have a small number of labelled examples. In other words, when there aren't enough labelled examples to observe the distributional properties we can generate more examples.

To examine the properties of low-count DA response distributions that have been learnt by the latent space, we can run RTNet-VAE multiple times on the turn pairs with the labelled response types. Due to the sampling of z in the encoder, the response encoding for a given turn pair will be different each time we sample it. This enables us to generate an unlimited number of examples of the dialogue act type. We can use this procedure to observe features of the generated distributions. We note that this procedure will not work with RTNet (without the VAE) since the response encoding without the VAE is deterministic.

As an example, we apply this method to an analysis of *agreement/disagreement* dialogue acts. Previous studies in conversation analysis on agreement/disagreement suggest multiple factors that may influence the response timings. As discussed previously, delayed responses have been linked to dispreference (Kendrick and Torreira, 2015). However, interruptions and overlaps have also been associated with strong disagreement (Greatbatch, 1992). Yet Tannen (1989) found overlaps to be an indicator of collaboration and agreement. From these previous studies we can infer that the factors that influence the distribution of response timings for agreement/disagreement may be complex.

The SWBD annotations include annotations for five levels of *agreement*: *agree*, *agree-part*, *maybe*, *reject-part*, *reject*. In our dataset there are only 104 examples of *reject* dialogue acts while there are 3897 examples of *agree* dialogue acts. The low number of *reject* dialogue acts makes it difficult to analyse the differences between the *agree* and *reject* distributions as shown in the top half of Fig. 6.20. To the best of our knowledge, there haven't been any other attempts to compare the timing offsets of *accept* and *reject* dialogue acts on the switchboard data, most likely due to the lack of labelled examples.

To observe the *reject* offset distribution we generate 20 times the original number of examples, increasing the number of *reject* offsets from 104 to 2080. The lower plot of Fig. 6.20 shows the generated *reject* and *accept* distributions. We can now clearly see a difference between the

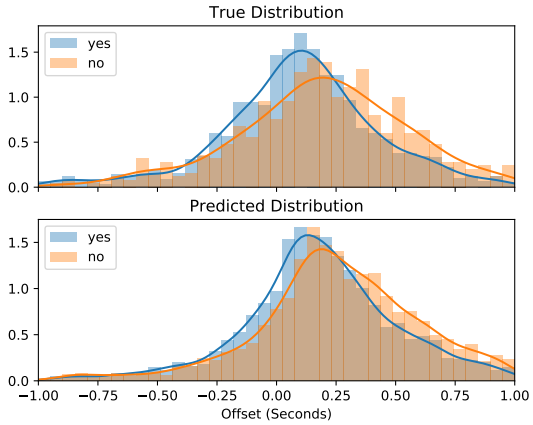


Figure 6.21: Generated distributions of "yes" and "no". Sampled using vector representations .

offsets of the two dialogue acts that was not observable from only using the labelled data. We can see that the *reject* mode is delayed relative to the *accept* mode and that there is relatively more probability mass from +250 ms onward. This suggests that the network has learned that the *reject* dialogue acts have delayed offset properties, supporting the results in Kendrick and Torreira (2015) and Bögels et al. (2019).

As a disclaimer we wish to state that we do not claim that conversational phenomena observed using this generative method can be considered conclusive. Nor do we claim that this can in any way replace real analysis using labelled data. There are also clear limitations to the model's accuracy in that it has difficulty with the details of distributions prior to the user's end point, as mentioned previously. We do believe though that this could potentially be a useful and flexible tool for performing rough conversational behaviour analyses that avoid the need for extensive data labelling. It could potentially be applied to analyse response behaviours for conversational aspects other than dialogue acts such as, for example, sentiment and emotion.

6.4.7 Sampling from the Latent Space

As mentioned in section 6.2.3.1, part of the appeal in using using the VAE in our model is that it enables us to discard the response encoding stage by sampling directly from the trained latent space. In this section we examine two options for sampling from the latent space. In the first we approximate the distribution of latent variables for individual dialogue act response types using isotropic Gaussians. This enables us to efficiently represent the dialogue acts using mean and standard-deviation vectors, a pair for each dialogue act. Figure 6.21 shows examples of distributions generated using Gaussian approximations of the latent space distributions. We can

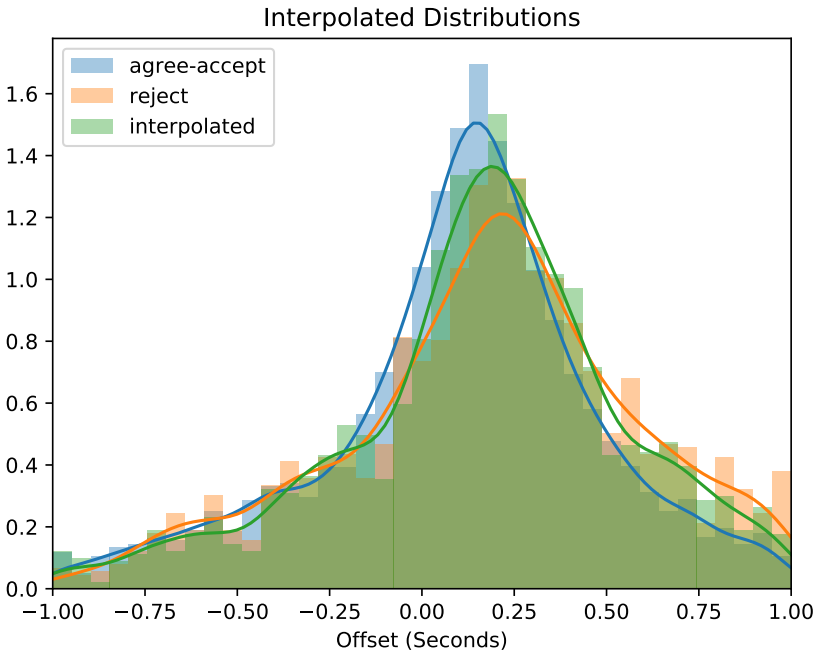


Figure 6.22: Interpolated distributions for reject and agree dialogue acts using the vector representations

see that the generated outputs have similar properties to the true distributions.

In the second option, we use the same parameterized vector representations but we interpolate between different dialogue act parameters to achieve intermediate distributions. This approach is similar to those used in (Roberts et al., 2018; Ha and Eck, 2018). This dimensional approach is flexible in that we give the dialogue manager (DM) more control over the details of the distribution. For example, if the objective of the SDS was to generate an *agree* dialogue act, we could control the *degree* of agreement by interpolating between *disagree* and *agree* vectors. Fig. 6.22 shows an example of a generated interpolated distribution. We can see that the properties of the interpolated distribution (e.g. mode, kurtosis) are perceptually "in between" the *reject* and *agree-accept* distributions. We can use this method to flexibly control aspects of the distribution without having to use an encoder.

6.5 Listening Test

6.5.1 Research Questions

Research has shown that response timings vary based on the semantic content of dialogue responses and the preceding turn (Levinson and Torreira, 2015), that listeners are sensitive to these fluctuations in timing (Bögels and Levinson, 2017), and that listeners make inferences about the upcoming turn based on response timings (Bögels et al., 2019). However, the question of whether certain response timings within different contexts are considered more *realistic* than others has not been fully investigated. We therefore validate our model by posing two research questions:

1. Given a preceding turn and a response, are some response timings considered by listeners to be more realistic than others?
2. In cases where listeners are sensitive to the response timing (if there are any), is our proposed model more likely to generate responses that are considered realistic than a system that generates a modal response time?

6.5.2 Listening Test Design

We designed a listening test to address these questions, in which we asked participants to listen to sets of turn pairs from our dataset where the response timings had been altered. We selected 16 turn exchanges from our dataset. We selected exchanges that deviated from the datasets modal response offset using the following heuristic: First, we remove the top and bottom one percent of the offsets to remove outliers. We then estimate the mode of the distribution using kernel density estimation, which was found to be +157 ms. The mode is then used to split the distribution into two segments. The median values of each of the two segments (-72 ms for *early*, +367 ms for *late*) are then used to partition the distribution into three sections: *early*, *modal*, and *late* (shown in Fig. 6.23). Since our objective is to determine whether some response timings are considered more realistic than others, we chose to examine dialogue acts that are associated with non-modal response timings: namely *backchannels* and *dispreferred* responses. Limiting our turn pairs to only these two response dialogue acts, we then randomly selected 100 backchannel responses and all 59 dispreference examples that are in our test set. We removed all turn pairs with any of the following attributes: unintelligible speech, barge-in, audible cross-talk, background noise, or loud breathing by the responder (to avoid the influence of breathing

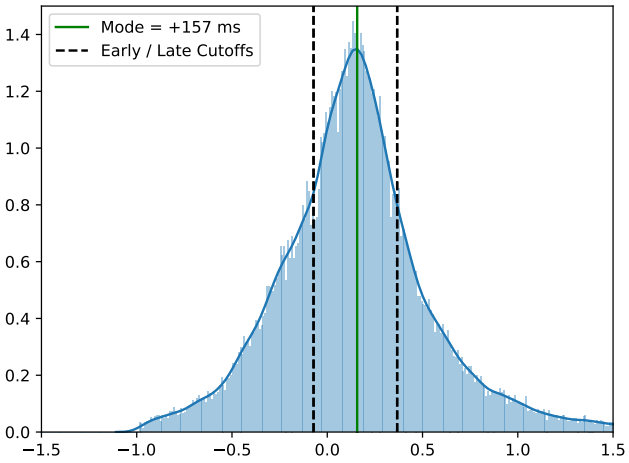


Figure 6.23: Early and late cutoffs that segment the offset distribution into *early*, *modal*, and *late* regions

turn-taking signals (Włodarczak and Heldner, 2016b)). We then selected eight examples of each dialogue act, where four of each dialogue act were classed as *early* and four of each dialogue act were classed as *late*. When selecting the early and late classes, we chose the samples from our subsets that were furthest from a zero offset. We used this approach to to avoid cases where there it is difficult to perceive a difference between modal and either late or early offsets.

For each turn pair we prepared audio files for three different response timings: *true*, *modal*, and *opposite*. The *true* response timing corresponds to the original response time in the turn pair. The *modal* response timing corresponds to the mode of the datasets offsets which was estimated (using kernel density estimation) to be +157 ms. The *opposite* response timing corresponds to the mean response timing of the other non-modal class i.e. when the true offset of an example is *early* the *opposite* response timing is the mean response timing of *late* (and vice-versa). The mean of the early and late timings were -316 ms and +760 ms respectively. We used the ground truth word annotations of the corpus to isolate the first and second turns. The audio files were normalized so that both the first and second turns had the same maximum loudness. We also added white noise at -50 dBFS in order to mask the cuts. The first speakers turn was played through the left audio channel and the second speaker's turn was played through the right audio channel.

The listeners were presented with two versions of the same turn pair at a time, each with different response timings. They were asked to make an A/B selection for which response timing they considered was most likely to be the original response timing. We use an A/B test rather than a three-way choice in order to avoid central tendency bias. They heard each turn turn pair

twice, with 32 questions in total. In the first 16 questions listeners were asked to compare *true* vs. *opposite* timings. In questions 17 to 32 they were asked to compare *true* vs. *modal* timings. The following prompt was used for all questions: "Which response timing sounds like it was produced in the real conversation?". The participants were informed that the response timings were shifted from their original timings. The test was administered through an online survey in which the participants were instructed that headphones were a requirement. We received ethics approval from the Trinity College School of Engineering which was granted to Professor Naomi Harte for the dates between 01/10/2019 and 31/12/2020. We had 25 participants in total (10 male, 15 female). We performed binomial tests for the significance of a given choice in each question. The results of the test are shown in Table 6.3.

6.5.3 Analysis

6.5.3.1 Research Question One

> Given a preceding turn and a response, are some response timings considered by listeners to be more realistic than others?

In the first half of our listening test we compared the *true* offset to the *opposite* offset. We found that listeners were able to identify the true offset in 10 out of the 16 clips with statistical significance. There are also no cases when the opposite offset was selected more frequently than the true offset. We can infer from this is that listeners are able to distinguish whether a response should trend towards early or late offsets on the basis of the context. This has implications for the design of dialogue systems. It indicates that if we wish to be able to realistically generate a wide range of human spoken expressions, care should be taken to not generate an offset that does not suit the dialogue context (user turn and system turn). In other words, based on the dialogue context, listeners consider some response timings to be more realistic than others.

6.5.3.2 Research Question Two

> In cases where listeners are sensitive to the response timing (if there are any), is our proposed model more likely to generate responses that are considered realistic than a system that generates a modal response time?

In the second half of our listening test we compared listener preference between the true offset and the modal offset. There were six cases where there was a statistically significant preference.

Question	Dialogue Act	Class	Offset (Seconds)	# Correct	# Wrong	P value
			Opposite			
1	ans_dispref	L	1.048125	24	1	**0.000002**
2	backchannel	E	-0.173375	17	8	0.107752
3	backchannel	L	0.486375	18	7	**0.043285**
4	ans_dispref	E	-0.109625	18	7	**0.043285**
5	ans_dispref	E	-0.253000	14	11	0.690038
6	ans_dispref	E	-0.177500	15	10	0.424356
7	backchannel	E	-0.222875	17	8	0.107752
8	backchannel	L	0.461375	19	6	**0.014633**
9	ans_dispref	L	0.536125	18	7	**0.043285**
10	backchannel	L	0.473875	18	7	**0.043285**
11	ans_dispref	L	0.534250	19	6	**0.014633**
12	ans_dispref	L	0.797375	19	6	**0.014633**
13	backchannel	E	-0.196125	17	8	0.107752
14	ans_dispref	E	-0.813625	18	7	**0.043285**
15	backchannel	L	0.525625	14	11	0.690038
16	backchannel	E	-0.221125	20	5	**0.004077**
			Modal			
17	ans_dispref	L	1.048125	12	13	1.000000
18	backchannel	E	-0.173375	13	12	1.000000
19	backchannel	L	0.486375	13	12	1.000000
20	ans_dispref	E	-0.109625	13	12	1.000000
21	ans_dispref	E	-0.253000	6	19	**0.014633**
22	ans_dispref	E	-0.177500	10	15	0.424356
23	backchannel	E	-0.222875	13	12	1.000000
24	backchannel	L	0.461375	10	15	0.424356
25	ans_dispref	L	0.536125	6	19	**0.014633**
26	backchannel	L	0.473875	11	14	0.690038
27	ans_dispref	L	0.534250	11	14	0.690038
28	ans_dispref	L	0.797375	20	5	**0.004077**
29	backchannel	E	-0.196125	18	7	**0.043285**
30	ans_dispref	E	-0.813625	20	5	**0.004077**
31	backchannel	L	0.525625	6	19	**0.014633**
32	backchannel	E	-0.221125	14	11	0.690038

Table 6.3: Listening test results table. Questions 1-16 test for preference between *opposite* and *true* offsets. Questions 17-32 test for preference between *modal* and *true* offsets.

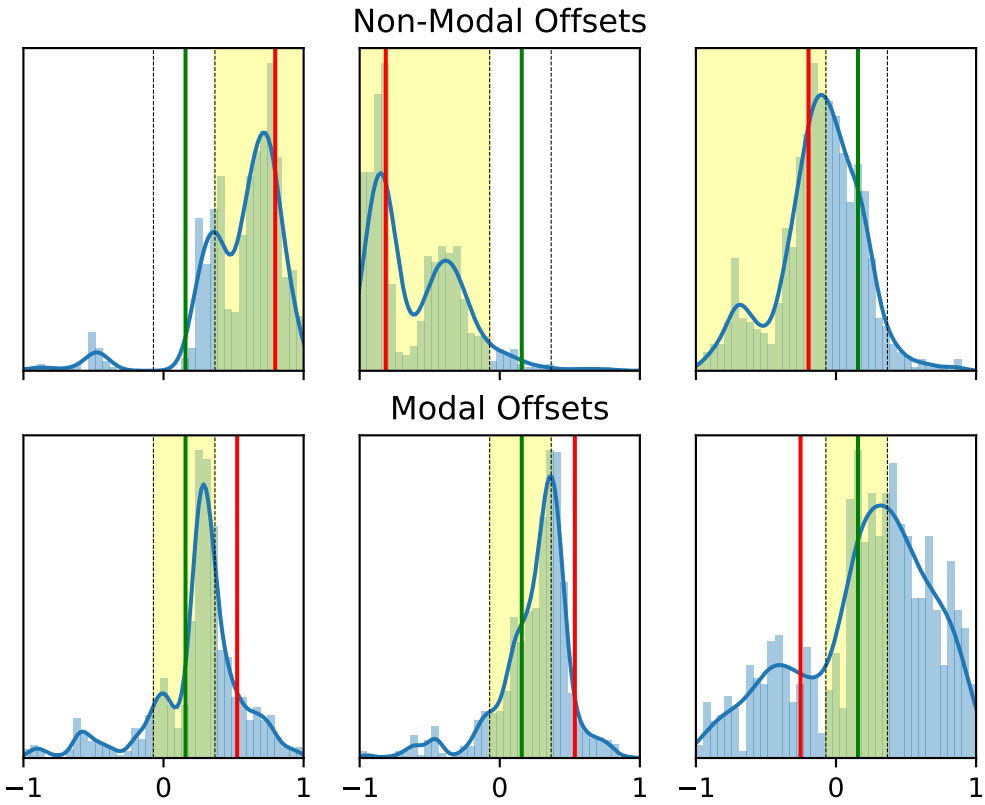


Figure 6.24: RTNet generated distributions for six turn pairs that were found to have statistically significant listener offset preferences in the true vs modal listening tests. The listener preferred region is highlighted in yellow. The true offset is indicated by the red line. The overall mode (+157 ms) is indicated by the green line.

In Fig. 6.24 we show the generated distributions using RTNet for these six cases. In three cases the true (non-modal) offset was significantly preferred (shown in the top row), and in three cases the modal offset was significantly preferred (shown in the bottom row). In our plots the region highlighted in yellow shows the preferred region and the true offset is indicated by the red line.

There are several initial observations that can be made here. Firstly, in cases where the listeners preferred the non-modal offsets (top row), RTNet generated distributions with modes that were close to the original non-modal offsets. It is also notable that the generated modes are never in the opposite region. However, in cases where the listeners preferred modal offsets (bottom row) RTNet generated distributions with modes and that were closer to the overall modal offset region. This presents evidence that listeners are sensitive to similar turn-taking cues that are being used by our RTNet model. We can conclude, in reference to our second research question, that in instances where listeners are sensitive to response timings it is likely that our system will generate response timings that are more realistic than a system that simply generates the mode of the dataset.

6.6 Conclusion

In this chapter we have presented models that can be used to generate the turn switch offset distributions of SDS system responses. It has been shown in prior studies (e.g. Bögels et al. (2019)) that humans are sensitive to these timings and that they can impact how responses are perceived by a listener. We would argue that they are an important element of producing naturalistic interactions that is often overlooked. With the advent of commercial SDS systems that attempt to engage users over extended multi-turn interactions (e.g. Zhou et al. (2020)) generating realistic response behaviours is a potentially desirable addition to the overall experience.

There are a number of aspects of the models that could be improved upon in future work. As discussed, the part of the distribution that the system has a hard time modelling is the overlap area between -500ms to 0ms. We propose that improving the predictive capabilities by modifying the loss function to include more prediction steps (similar to Skantze (2017b)) may improve this. Another research strand that could be investigated is modelling at what point during the user's turn *early planning* of the system response can start. In our current model this is approximated by the randomization of R_{START}, but more principled approaches could be designed. Bögels et al. (2018) found that the order of words in a speakers turn affected a responders latency since different word orderings affecting when early planning could begin. Since this has been shown

to affect offsets in human-human conversations it would be useful investigate applications in the context of an SDS. DeVault et al. (2009) and DeVault et al. (2011) found points where the system has reached a point of maximal understanding. However, this approach does not fully operate the way humans do because it does not begin partial planning of the utterance when partial amounts of the information are available.

Chapter 7

Conclusion

7.1 Overview

The objective of this thesis was to investigate models that can be used to simulate naturalistic turn-taking behaviours in SDSs. The thesis takes the position that traditional endpointing-based turn-taking models, where the system reacts to user pauses rather than predictively anticipates them, cannot fully model naturalistic turn-taking behaviours. Endpointing-based models can be used robustly in current IVAs that perform utilitarian functions (e.g. calenders, weather forecasts, news updates). However, as SDSs become more advanced and engage in more sophisticated conversations, more naturalistic turn-taking behaviours will be desired of them (Ward et al., 2005).

In order to simulate naturalistic turn-taking behaviours (such as fast-turn switches, intentional overlap, backchanneling, and barge-in) SDSs will need to be able to be both *predictive* and *incremental*. They will need to be predictive in the sense that they predict future user behaviour rather than respond to user behaviours that have already occurred. In the *projection theory* of Sacks et al. (1974) they proposed that humans are able to anticipate turn endings before they occur. We argue that SDSs that aspire to displaying naturalistic turn-taking behaviours should be capable of anticipating user behaviour as well. To make these predictive decisions the system must be incremental, in order to make decisions while the user is speaking. The ability to incrementally make these predictive decisions requires models of the turn-taking cues that allow the human turn organization process to take place. As discussed in Section 2.1.2, some of these cues can be subtle, or rely on an understanding of natural spoken language which is inherently

complex.

In this thesis we explored sequential turn-taking models that have both predictive and incremental properties. We began in Chapter 3 by investigating ways of improving CTT modelling as proposed by Skantze (2017b) through an investigation of input features. In the literature on non-sequential models it was generally found that linguistic features were more useful than prosodic features for performing endpointing decisions (e.g. Raux and Eskenazi (2012), Gravano and Hirschberg (2011), Meena et al. (2014)). However, in our investigations we found that acoustic features contributed more to the overall performance of the sequential model. In an SFS experiment we found that loudness, F0 and low order MFCCs contributed the most to the performance. Skantze (2017b) proposed the use of syntactic POS features in CTT models. We compared syntactic features with lexical embeddings and found that, in general, syntactic features were unnecessary when lexical embeddings were available. We also proposed improvements to the loss function and the training procedure.

Motivated by the observation that turn-taking cues can be organized into a hierarchy of temporal granularities, with prosodic and acoustic cues occurring at a fast temporal rate and linguistic cues occurring at a slower temporal rate, in Chapter 4 we proposed a multiscale approach to CTT. We found that there are performance benefits to modelling the acoustic features at a fast temporal rate while modelling the linguistic features more slowly. We also found that the approach is useful for incorporating gaze features into turn-taking models.

In Chapter 5 we proposed a control process that enables turn switch decisions to be made prior to the end of the user's utterance, anticipating the end of the turn. The control process is based on partially observable Markov decision processes (POMDPs) and uses probabilistic output predictions from a modified CTT model. We showed that the POMDP is able to outperform several other baselines. It also enables flexible tuning of the trade-off between latency and false-cut-ins.

In Chapter 6 we proposed our RTNet and RTNet-VAE models which take into account the context of both the user's turn as well as the system's response to generate the distribution of turn-switch offsets. These models use an encoding of the system turn, as well as acoustic and linguistic features extracted from the user's speech signal, to make the binary decision to start speaking or not. The two models represent a class of continuous model that is distinct from CTT models in its objective function and its architecture. RTNet-VAE used a variational autoencoder (VAE) to train an interpretable representation of the response encoding, which allows easier

integration with SDS pipelines. We presented the results of human listening tests which showed that listeners found that some response timings were more natural than others. We showed that in instances where listeners are sensitive to response timings it is likely that our system will generate response timings that are more realistic than a system that generates the modal offset.

7.2 Future Directions

The work in this thesis suggests a number of research directions that have yet to be explored. Among them are:

- **ASR confidence and instability in continuous models**

  In all of our CTT models we make the modelling assumption of perfect ASR results and we model the delay by simply delaying word annotations 100 ms after the ground truth end of the word. However, as discussed in Section 2.2.3, ASR is subject to instability and delays that have implications for the downstream components. The question of how these two aspects should be treated in continuous RNN-based systems has not yet been addressed. In particular, how should ASR confidence scores be incorporated and how can ASR-repair be dealt with efficiently. These are immediate concerns that should be addressed to allow practical incremental implementations to incorporate lexical features.

- **Modelling early planning**

  In our description of RTNet in Chapter 6 we make the modelling assumption that the point at which RTNet receives an encoding from the DM (referred to as R_{START}) is modelled by a uniform distribution over the frames from the start of the user's turn-final IPU to the ground-truth start time. This is a crude approximation of the real performance of the output of an incremental dialogue manager. The literature from CA on early planning (e.g. Bögels et al. (2015)) suggests that there could be better ways to model the points where early planning of a turn can begin. Example ideas of ways this could be done could be to incorporate information theory and/or ASR constraints.

- **Conversation analysis using RTNets**

  In Chapter 6 we proposed that information stored in the latent space of the VAE could potentially be used for CA. Since the model can be trained on large corpora with hundreds of hours of conversations, it should be able to learn information about turn-taking offsets

that occur in a large number of conditions. As proposed in that chapter, the generative nature of the model can potentially be used to investigate turn-taking behaviours caused by different dialogue act contexts, as well as other phenomena such as sentiment and emotion.

- **Generating multimodal turn-taking cues in a robot using RTNet or CTT**

  Continuous models have already been used to detect points that can be used to generate fillers in a fast responsive manner (Lala et al., 2019a). However, their potential for detecting points to generate other forms of turn-grabbing and turn-yielding cues, particularly multimodal ones, have not yet been explored. For example, in an embodied agent non-intrusive head-nods could be generated as backchanneling or feedback behaviours using continuous models. Gaze aversion could be controlled by a continuous model to signal the agent's intention to take the floor when the user's turn is predicted to be coming to an end. Inhaling or mouth-opening could be synthesized as a way of signalling the agent's intention to take a turn. There is a wealth of possibilities for these types of behaviours to be generated in a continuous manner.

- **Multi-party continuous turn-taking modelling**

  In this thesis we have focused on dyadic interactions. However, SDSs are also often used in multi-party scenarios where turn-taking models must predict not just *when* a user will speak but also *who* will speak next (Bohus and Horvitz, 2009, 2010; Oertel, 2017). For example, Bohus and Horvitz (2010) explored implementing turn-taking models in a situated directions-giving robot, where more than one person may be interacting with the robot at a time. The added complexities of implementing continuous models in a multi-party scenario such as this have yet to be explored. For example, the CTT model must account for who is being addressed by a speaker since there are likely points when users will address each-other. The agent's gaze and other multimodal behaviours could also potentially be controlled by a multi-party continuous model.

Bibliography

Mojtaba Khomami Abadi, Jacopo Staiano, Alessandro Cappelletti, Massimo Zancanaro, and Nicu Sebe. Multimodal engagement classification for affective cinema. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference On*. IEEE, 2013.

Hervé Abdi. The Bonferonni and Šidák corrections for multiple comparisons. *Encyclopedia of measurement and statistics*, 3, 2007.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a Human-like Open-Domain Chatbot. *arXiv:2001.09977 [cs, stat]*, January 2020.

Samer Al Moubayed, Jonas Beskow, Gabriel Skantze, and Björn Granström. Furhat: A back-projected human-like robot head for multiparty human-machine interaction. In *Cognitive Behavioural Systems*. Springer, 2012.

Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. The HCRC map task corpus. *Language and speech*, 34(4), 1991.

Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010.

Michaela Atterer, Timo Baumann, and David Schlangen. Towards incremental end-of-utterance detection in dialogue systems. In *Proceedings of the 22nd International Conference on Computational Linguistics*, 2008.

Tadas Baltrušaitis, Peter Robinson, Louis-Philippe Morency, et al. OpenFace: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

Timo Baumann and David Schlangen. INPRO_iSS: A Component for Just-In-Time Incremental Speech Synthesis. In *Proceedings of the ACL 2012 System Demonstrations*, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Timo Baumann and David Schlangen. Open-ended, extensible system utterances are preferred, even if they require filled pauses. In *Proceedings of the SIGDIAL 2013 Conference*, 2013.

Timo Baumann, Casey Kennington, Julian Hough, and David Schlangen. Recognising conversational speech: What an incremental asr should do for a dialogue system and how to get there. In *Dialogues with Social Robots*. Springer, 2017.

Janet Beavin Bavelas and Jennifer Gerwing. The Listener as Addressee in Face-to-Face Dialogue. *International Journal of Listening*, 25(3), September 2011.

Štefan Beňuš. Variability and stability in collaborative dialogues: Turn-taking and filled pauses. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

Štefan Beňuš, Agustín Gravano, and Julia Hirschberg. Pragmatic aspects of temporal accommodation in turn-taking. *Journal of Pragmatics*, 43(12), September 2011.

Sanjay Bilakhia, Stavros Petridis, Anton Nijholt, and Maja Pantic. The MAHNOB Mimicry Database: A database of naturalistic human interactions. *Pattern Recognition Letters*, 66, November 2015.

Sara Bögels and Stephen C. Levinson. The Brain Behind the Response: Insights Into Turn-taking in Conversation From Neuroimaging. *Research on Language and Social Interaction*, 50 (1), January 2017.

Sara Bögels and Francisco Torreira. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52, September 2015.

Sara Bögels, Lilla Magyari, and Stephen C. Levinson. Neural signatures of response planning occur midway through an incoming question in conversation. *Scientific Reports*, 5(1), October 2015.

Sara Bögels, Marisa Casillas, and Stephen C. Levinson. Planning versus comprehension in turn-taking: Fast responders show reduced anticipatory processing of the question. *Neuropsychologia*, 109, January 2018.

Sara Bögels, Kobin H. Kendrick, and Stephen C. Levinson. Conversational expectations get revised as response latencies unfold. *Language, Cognition and Neuroscience*, March 2019.

Dan Bohus and Eric Horvitz. Models for multiparty engagement in open-world dialog. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2009.

Dan Bohus and Eric Horvitz. Facilitating multiparty dialog with gaze, gesture, and speech. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. ACM, 2010.

Dan Bohus and Alexander I. Rudnicky. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3), July 2009.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany, 2016. Association for Computational Linguistics.

Pablo Brusco, Juan Manuel Pérez, and Agustín Gravano. Cross-Linguistic Study of the Production of Turn-Taking Cues in American English and Argentine Spanish. In *Interspeech 2017*. ISCA, August 2017.

Matthew Bull and Matthew Aylett. An Analysis of the Timing of Turn-Taking in a Corpus of Goal-Oriented Dialogue. In *Fifth International Conference on Spoken Language Processing*, 1998.

Harry Bunt. Multifunctionality and multidimensional dialogue act annotation. *Communication-Action-Meaning. Gothenburg*, 2007.

Harry Bunt. The DIT++ taxonomy for functional dialogue markup. *AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts*, 2009.

Harry Bunt, Jan Alexandersson, Jean Carletta, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Kiyong Lee, Volha Petukhova, Andrei Popescu-Belis, Laurent Romary, Claudia Soria, and David Traum. Towards an ISO Standard for Dialogue Act Annotation. *LREC*, 2010.

BusinessWire. Global Intelligent Virtual Assistant Market 2018-2023: Market Value is Projected to Exceed US$ 9 Billion by 2023, Expanding at a CAGR of 32% - ResearchAndMarkets.com. https://www.businesswire.com/news/home/20180723005506/en/Global-Intelligent-Virtual-Assistant-Market-2018-2023-Market, July 2018.

Sasha Calhoun, Jean Carletta, Jason M. Brenier, Neil Mayo, Dan Jurafsky, Mark Steedman, and David Beaver. The NXT-format Switchboard Corpus: A rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue. *Language Resources and Evaluation*, 44(4), December 2010.

Alexandra Canavan, David Graff, and George Zipperlen. CALLHOME American English Speech LDC97S42. Web Download., 1997.

Nicola Cathcart, Jean Carletta, and Ewan Klein. A shallow model of backchannel continuers in spoken dialogue. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003.

Christian Cavé, Isabelle Guaïtella, Roxane Bertrand, Serge Santi, Françoise Harlay, and Robert Espesser. About the relationship between eyebrow movements and Fo variations. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference On*, volume 4. IEEE, 1996.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014. Association for Computational Linguistics.

Noam Chomsky. *Syntactic Structures*. Mouton and Co., The Hague, 1957.

Patricia M. Clancy, Sandra A. Thompson, Ryoko Suzuki, and Hongyin Tao. The conversational use of reactive tokens in English, Japanese, and Mandarin. *Journal of Pragmatics*, 26(3), September 1996.

I. Cohen. Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging. *IEEE Transactions on Speech and Audio Processing*, 11(5), September 2003.

Israel Cohen, Yiteng Huang, Jingdong Chen, and Jacob Benesty. *Noise Reduction in Speech Processing*, volume 2 of *Springer Topics in Signal Processing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

Martin Corley and Oliver W Stewart. Hesitation disfluencies in spontaneous speech: The meaning of um. *Language and Linguistics Compass*, 2(4), 2008.

Iwan De Kok and Dirk Heylen. Multimodal end-of-turn prediction in multi-party meetings. In *Proceedings of the 2009 International Conference on Multimodal Interfaces*. ACM, 2009.

Jan-Peter De Ruiter, Holger Mitterer, and Nick J. Enfield. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82(3), 2006.

Neeraj Deshmukh, Aravind Ganapathiraju, Andi Gleeson, Jonathan Hamaker, and Joseph Picone. Resegmentation of SWITCHBOARD. In *Fifth International Conference on Spoken Language Processing*, 1998.

Nina Dethlefs, Helen Hastie, Verena Rieser, and Oliver Lemon. Optimising incremental dialogue decisions using information density for interactive systems. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012.

Nina Dethlefs, Helen Hastie, Heriberto Cuayáhuitl, Yanchao Yu, Verena Rieser, and Oliver Lemon. Information density and overlap in spoken dialogue. *Computer Speech & Language*, 37, May 2016.

David DeVault, Kenji Sagae, and David Traum. Can I finish?: Learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2009.

David DeVault, Kenji Sagae, and David Traum. Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue & Discourse*, 2(1), 2011.

Dgit. The best smart connected speaker: Apple HomePod vs Google Home vs Amazon Echo. https://dgit.com/apple-homepod-vs-google-home-vs-amazon-echo-53296/, January 2020.

Starkey Duncan. Some signals and rules for taking speaking turns in conversations. *Journal of personality and social psychology*, 23(2), 1972.

Y. Ephraim and D. Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6), December 1984.

Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the International Conference on Multimedia*. ACM, 2010.

Florian Eyben, Klaus R. Scherer, Bjorn W. Schuller, Johan Sundberg, Elisabeth Andre, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing. *IEEE Transactions on Affective Computing*, 7(2), April 2016.

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody. In *Seventh International Conference on Spoken Language Processing*, 2002.

Kerstin Fischer. What computer talk is and isn't: Human-computer conversation as intercultural communication. Vol. 17. *Linguistics-Computational Linguistics. AQ-Verlag*, 2006.

Cecilia E Ford and Sandra A Thompson. Interactional units in conversation: Syntactic, intonational, and pragmatic resources for the management of turns. *Studies in interactional sociolinguistics*, 13, 1996.

Hiroko Furo. *Turn-Taking in English and Japanese: Projectability in Grammar, Intonation and Semantics*. Routledge, 2013.

Tanmay Gangwani, Joel Lehman, Qiang Liu, and Jian Peng. Learning belief representations for imitation learning in pomdps. In *Uncertainty in Artificial Intelligence*. PMLR, 2019.

Simon Garrod and Martin J. Pickering. Why is conversation so easy? *Trends in cognitive sciences*, 8(1), 2004.

Henry Goble and Chad Edwards. A Robot That Communicates With Vocal Fillers Has ... Uhhh ... Greater Social Presence. *Communication Research Reports*, 35(3), May 2018.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. IEEE, 1992.

John Godfrey and Edward Holliman. Switchboard-1 release 2 LDC97S62. *Web Download. Philadelphia: Linguistic Data Consortium*, 1993.

Charles Goodwin. *Conversational Organization: Interaction between Speakers and Hearers*. Academic Press, 1981.

Jonathan Gratch, Ning Wang, Jillian Gerten, Edward Fast, and Robin Duffy. Creating rapport with virtual agents. In *Intelligent Virtual Agents*. Springer, 2007.

Agustín Gravano and Julia Hirschberg. Backchannel-inviting cues in task-oriented dialogue. In *INTERSPEECH*, 2009.

Agustín Gravano and Julia Hirschberg. Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3), July 2011.

David Greatbatch. On the management of disagreement between news interviewees. *Talk at work: Interaction in institutional settings*, 1992.

Isabelle Guaïtella, Serge Santi, and Christian Cavé. Are eyebrow movements linked to voice variations and turn-taking? An experimental investigation. *Gesture*, 2009.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, and Alexander Gruenstein. Streaming End-to-end Speech Recognition for Mobile Devices. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.

Peter A. Heeman and Rebecca Lunsford. Turn-taking offsets and dialogue context. In *Proc. Interspeech 2017*, 2017.

Mattias Heldner and Jens Edlund. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4), October 2010.

Mattias Heldner, Marcin Włodarczak, Štefan Beňuš, and Agustín Gravano. Voice Quality as a Turn-Taking Cue. In *Interspeech 2019*. ISCA, September 2019.

Salah El Hihi and Yoshua Bengio. Hierarchical Recurrent Neural Networks for Long-Term Dependencies. *Advances in neural information processing systems (NIPS)*, 1996.

H.G. Hirsch and C. Ehrlicher. Noise estimation techniques for robust speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, Detroit, MI, USA, 1995. IEEE.

Anna Hjalmarsson. The additive effect of turn-taking cues in human and synthetic voice. *Speech Communication*, 53(1), January 2011.

Anna Hjalmarsson and Catharine Oertel. Gaze direction as a back-channel inviting cue in dialogue. In *IVA 2012 Workshop on Realtime Conversational Virtual Agents*, volume 9. Citeseer, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8), 1997.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: The difficulty of learning long-term dependencies. 2001.

Paul Hömke, Judith Holler, and Stephen C Levinson. Eye blinks are perceived as communicative signals in human face-to-face interaction. *PloS one*, 13(12), 2018.

Matthew Honnibal and Ines Montani. Spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing*, 2017.

Peter Indefrey and Willem JM Levelt. The spatial and temporal signatures of word production components. *Cognition*, 92(1-2), 2004.

Koji Inoue, Pierrick Milhorat, Divesh Lala, Tianyu Zhao, and Tatsuya Kawahara. Talking with ERICA, an autonomous android. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Los Angeles, September 2016. Association for Computational Linguistics.

Ryo Ishii, Shiro Kumano, and Kazuhiro Otsuka. Analyzing mouth-opening transition pattern for predicting next speaker in multi-party meetings. ACM Press, 2016a.

Ryo Ishii, Kazuhiro Otsuka, Shiro Kumano, and Junji Yamato. Using Respiration to Predict Who Will Speak Next and When in Multiparty Meetings. *ACM Transactions on Interactive Intelligent Systems*, 6(2), August 2016b.

Joseph Jaffe and S Feldstein. Rhythms of Dialogue (New York). 1970.

Jongseo Sohn and Wonyong Sung. A voice activity detector employing soft decision based noise spectrum adaptation. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98*, volume 1, May 1998.

Dan Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL labeling project coder's manual. *Draft 13. Technical Report 97-02*, 1997.

Daniel Jurafsky, Elizabeth Shriberg, Barbara Fox, and Traci Curl. Lexical, prosodic, and syntactic cues for dialog acts. In *Discourse Relations and Discourse Markers*, 1998.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2), May 1998.

John Kane and Irena Yanushevskaya. Analysing the Prosodic Characteristics of Speech-Chunks Preceding Silences in Task-Based Interactions. *Interspeech*, 2014.

Adam Kendon. Some functions of gaze-direction in social interaction. *Acta Psychologica*, 26, January 1967.

Kobin H. Kendrick and Francisco Torreira. The timing and construction of preference: A quantitative study. *Discourse Processes*, 52(4), 2015.

Casey Kennington, Spyros Kousidis, and David Schlangen. InproTKs: A Toolkit for Incremental Situated Processing. Association for Computational Linguistics, 2014.

Hatim Khouzaimi, Romain Laroche, and Fabrice Lefevre. Optimising Turn-Taking Strategies With Reinforcement Learning. In *SigDial*. Association for Computational Linguistics, 2015.

D Kingma and J Ba Adam. Adam: A Method for Stochastic Optimisation. In *International Conference on Learning Representations*, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014.

Hanae Koiso, Yasuo Horiuchi, Syun Tutiya, Akira Ichikawa, and Yasuharu Den. An analysis of turn-taking and backchannels based on prosodic and syntactic features in Japanese map task dialogs. *Language and speech*, 41(3-4), 1998.

Mineichi Kudo and Jack Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern recognition*, 33(1), 2000.

Divesh Lala, Pierrick Milhorat, Koji Inoue, Masanari Ishida, Katsuya Takanashi, and Tatsuya Kawahara. Attentive listening system with backchanneling, response generation and flexible turn-taking. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, Saarbrücken, Germany, 2017. Association for Computational Linguistics.

Divesh Lala, Koji Inoue, and Tatsuya Kawahara. Evaluation of Real-time Deep Learning Turn-taking Models for Multiple Dialogue Scenarios. In *Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI '18*, Boulder, CO, USA, 2018. ACM Press.

Divesh Lala, Koji Inoue, and Tatsuya Kawahara. Smooth Turn-taking by a Robot Using an Online Continuous Model to Generate Turn-taking Cues. In *2019 International Conference on Multimodal Interaction on - ICMI '19*, Suzhou, China, 2019a. ACM Press.

Divesh Lala, Shizuka Nakamura, and Tatsuya Kawahara. Analysis of Effect and Timing of Fillers in Natural Turn-Taking. In *Interspeech 2019*. ISCA, September 2019b.

LDC. 2000 HUB5 English Evaluation Speech LDC2002S09. *Web Download. Philadelphia: Linguistic Data Consortium*, 2002.

Yaniv Leviathan and Yossi Matias. Google duplex: An ai system for accomplishing real-world tasks over the phone, May 2018.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1), 2000.

Stephen C Levinson. Pragmatics (Cambridge textbooks in linguistics). 1983.

Stephen C. Levinson. Turn-taking in Human Communication – Origins and Implications for Language Processing. *Trends in Cognitive Sciences*, 20(1), January 2016.

Stephen C. Levinson and Francisco Torreira. Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*, 6, June 2015.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

Zachary C Lipton. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*, 2016.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, 2016. Association for Computational Linguistics.

D. Lu, T. Nishimoto, and N. Minematsu. Decision of response timing for incremental speech recognition with reinforcement learning. In *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, December 2011.

Lilla Magyari, Marcel C. M. Bastiaansen, Jan P. de Ruiter, and Stephen C. Levinson. Early Anticipation Lies behind the Speed of Response in Conversation. *Journal of Cognitive Neuroscience*, 26(11), November 2014.

Angelika Maier, Julian Hough, and David Schlangen. Towards Deep End-of-Turn Prediction for Situated Spoken Dialogue Systems. *Proceedings of INTERSPEECH 2017*, 2017.

Andrei Malchanau. *Cognitive Architecture of Multimodal Multidimensional Dialogue Management*. PhD thesis, Ph. D. Thesis, University of Saarland, Saarbrücken, 2018.

Ryo Masumura, Taichi Asami, Hirokazu Masataki, Ryo Ishii, and Ryuichiro Higashinaka. Online End-of-Turn Detection from Speech Based on Stacked Time-Asynchronous Sequential Networks. ISCA, August 2017.

Ryo Masumura, Tomohiro Tanaka, Atsushi Ando, Ryo Ishii, Ryuichiro Higashinaka, and Yushi Aono. Neural Dialogue Context Online End-of-Turn Detection. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 2018.

David H. McFarland. Respiratory markers of conversational interaction. *Journal of Speech, Language, and Hearing Research*, 2001.

Gary McKeown, Michel F. Valstar, Roderick Cowie, and Maja Pantic. The SEMAINE corpus of emotionally coloured character interactions. In *Multimedia and Expo (ICME), 2010 IEEE International Conference On*. IEEE, 2010.

Raveesh Meena, Gabriel Skantze, and Joakim Gustafson. Data-driven models for timing feedback responses in a Map Task dialogue system. *Computer Speech & Language*, 28(4), 2014.

Håkan Melin, Anna Sandell, and Magnus Ihse. CTT-bank: A speech controlled telephone banking system-an initial evaluation. *TMH-QPSR*, 1, 2001.

Tomer Meshorer and Peter A Heeman. Using Past Speaker Behavior to Better Predict Turn Transitions. In *INTERSPEECH*, 2016.

Thilo Michael and Sebastian Möller. ReTiCo: An open-source framework for modeling real-time conversations in spoken dialogue systems. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, 2019.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent Neural Network Based Language Model. 2010.

Louis-Philippe Morency, Iwan de Kok, and Jonathan Gratch. A probabilistic multimodal approach for predicting listener backchannels. *Autonomous Agents and Multi-Agent Systems*, 20(1), 2010.

Nelson Morgan and Herve Bourlard. Continuous speech recognition. *IEEE signal processing magazine*, 12(3), 1995.

Ryosuke Nakanishi, Koji Inoue, Shizuka Nakamura, Katsuya Takanashi, and Tatsuya Kawahara. Generating Fillers based on Dialog Act Pairs for Smooth Turn-Taking by Humanoid Robot. *IWSDS*, 2018.

Daniel Neiberg and Khiet P. Truong. Online detection of vocal listener responses with maximum latency constraints. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference On*. IEEE, 2011.

Catharine Oertel. *Modeling engagement in multi-party conversations: data-driven approaches to understanding human-human communication patterns for use in human-robot interactions*. PhD thesis, KTH Royal Institute of Technology, Stockholm, 2017.

Richard Ogden. Turn transition, creak and glottal stop in Finnish talk-in-interaction. *Journal of the International Phonetic Association*, 31(1), 2001.

Bengt Oreström. *Turn-Taking in English Conversation*. Krieger Pub Co, 1983.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*. IEEE, April 2015.

Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3), 1987.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Philadelphia, Pennsylvania, 2001. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014. Association for Computational Linguistics.

Volha Petukhova. *Multidimensional Dialogue Modelling*. PhD thesis, Tilburg University, 2011.

Rohit Prabhavalkar, Kanishka Rao, Tara N. Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. A Comparison of Sequence-to-Sequence Models for Speech Recognition. In *Interspeech 2017*. ISCA, August 2017.

Hugo Quené. On the just noticeable difference for tempo in speech. *Journal of Phonetics*, 35 (3), July 2007.

Antoine Raux. *Flexible Turn-Taking for Spoken Dialog Systems*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2008.

Antoine Raux and Maxine Eskenazi. Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, 2008.

Antoine Raux and Maxine Eskenazi. A finite-state turn-taking model for spoken dialog systems. In *HLT-NAACL*. ACL, 2009.

Antoine Raux and Maxine Eskenazi. Optimizing the turn-taking behavior of task-oriented spoken dialog systems. *ACM Transactions on Speech and Language Processing*, 9(1), May 2012.

Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. Let's Go Public! Taking a Spoken Dialog System to the Real World. *Ninth European conference on speech communication and technology*, 2005.

S. Zahra Razavi, Benjamin Kane, and Lenhart K. Schubert. Investigating Linguistic and Semantic Features for Turn-Taking Prediction in Open-Domain Human-Computer Conversation. In *Interspeech 2019*. ISCA, September 2019.

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *ICML*, 2018.

Sean Roberts, Francisco Torreira, and Stephen C. Levinson. The effects of processing and sequence organization on the timing of turn taking: A corpus study. *Frontiers in Psychology*, 6, May 2015.

Amélie Rochet-Capellan and Susanne Fuchs. Take a breath and take the turn: How breathing meets turns in spontaneous dialogue. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1658), December 2014.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A Simplest Systematics for the Organization of Turn-Taking for Conversation. *Language*, 50(4), December 1974.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

Emanuel A. Schegloff. Overlapping talk and the organization of turn-taking for conversation. *Language in Society*, 29(01), January 2000.

Emanuel A Schegloff and Harvey Sacks. Opening up closings. *Semiotica*, 8(4), 1973.

David Schlangen and Gabriel Skantze. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 2011.

David Schlangen, Timo Baumann, Hendrik Buschmeier, Okko Buß, Stefan Kopp, Gabriel Skantze, and Ramin Yaghoubzadeh. Middleware for incremental processing in conversational agents. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010.

Jürgen Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2), 1992.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. In *AAAI*, volume 16, 2016.

Prashanth Gurunath Shivakumar, Naveen Kumar, Panayiotis Georgiou, and Shrikanth Narayanan. Incremental Online Spoken Language Understanding. *arXiv:1910.10287 [cs, eess]*, October 2019.

Elizabeth Shriberg, Andreas Stolcke, Daniel Jurafsky, Noah Coccaro, Marie Meteer, Rebecca Bates, Paul Taylor, Klaus Ries, Rachel Martin, and Carol van Ess-Dykema. Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech? *Language and Speech*, 41(3-4), July 1998.

Elizabeth Shriberg, Andreas Stolcke, and Don Baron. Observations on overlap: Findings and implications for automatic processing of multi-party conversation. In *Seventh European Conference on Speech Communication and Technology*, 2001.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research*, 16, 2002.

Gabriel Skantze. A testbed for examining the timing of feedback using a map task. In *Feedback Behaviors in Dialog*, 2012.

Gabriel Skantze. Predicting and Regulating Participation Equality in Human-robot Conversations: Effects of Age and Gender. In *12th ACM/IEEE International Conference on Human-Robot Interaction*. ACM Press, 2017a.

Gabriel Skantze. Towards a General, Continuous Model of Turn-taking in Spoken Dialogue using LSTM Recurrent Neural Networks. In *Proceedings of SigDial*, Saarbrucken, Germany, 2017b.

Gabriel Skantze and Anna Hjalmarsson. Towards incremental speech generation in dialogue systems. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010.

Gabriel Skantze and Anna Hjalmarsson. Towards incremental speech generation in conversational systems. *Computer Speech & Language*, 27(1), January 2013.

Gabriel Skantze and David Schlangen. Incremental dialogue processing in a micro-domain. Association for Computational Linguistics, 2009.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous. Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron. *Proceedings of the 35 th International Conference on Machine Learning*, 2018.

Kemal Sönmez, Elizabeth Shriberg, Larry Heck, and Mitchel Weintraub. Modeling dynamic prosodic variation for speaker verification. In *Fifth International Conference on Spoken Language Processing*, 1998.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 2014.

Tanya Stivers, Nicholas J. Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter, Kyung-Eun Yoon, et al. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26), 2009.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26 (3), 2000.

Sofia Strömbergsson, Anna Hjalmarsson, Jens Edlund, and David House. Timing responses to questions in dialogue. In *INTERSPEECH*, 2013.

Zachary Sunberg and Mykel Kochenderfer. Online algorithms for POMDPs with continuous state, action, and observation spaces. *Proceedings of the International Conference on Automated Planning and Scheduling*, September 2017.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 2014.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2), 1999.

Deborah Tannen. Interpreting interruption in conversation. *Gender and discourse*, 1989.

P. A. Taylor. Concept-to-speech synthesis by phonological structure matching. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 358(1769), April 2000.

Paul Ten Have. Methodological issues in conversation analysis. *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique*, 27(1), 1990.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv:1511.01844 [cs, stat]*, April 2016.

Francisco Torreira, Sara Bögels, and Stephen C. Levinson. Breathing for answering: The time course of response planning in conversation. *Frontiers in Psychology*, 6, March 2015.

David Traum and Jeff Rickel. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2*. ACM, 2002.

Khiet P. Truong, R. W. Poppe, I. A. de Kok, and D. K. J. Heylen. A multimodal analysis of vocal and visual backchannels in spontaneous dialogs. *Twelfth Annual Conference of the International Speech Communication*, 2011.

Vivian Tsai, Timo Baumann, Florian Pecune, and Justine Cassell. Faster Responses Are Better Responses: Introducing Incrementality into Sociable Virtual Personal Assistants. In Luis Fernando D'Haro, Rafael E. Banchs, and Haizhou Li, editors, *9th International Workshop on Spoken Dialogue System Technology*, volume 579. Springer Singapore, Singapore, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017.

Oriol Vinyals and Quoc Le. A neural conversational model. *ICML Deep Learning Workshop*, 2015.

Marilyn A Walker. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12, 2000.

N. G. Ward, D. Aguirre, G. Cervantes, and O. Fuentes. Turn-Taking Predictions across Languages and Genres Using an LSTM Recurrent Neural Network. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, December 2018.

Nigel Ward and Wataru Tsukahara. Prosodic features which cue back-channel responses in English and Japanese. *Journal of pragmatics*, 32(8), 2000.

Nigel G. Ward, Anais G. Rivera, Karen Ward, and David G. Novick. Root causes of lost time and user stress in a simple dialog system. In *Ninth European Conference on Speech Communication and Technology*, 2005.

Tsung-Hsien Wen and Steve Young. Recurrent neural network language generation for spoken dialogue systems. *Computer Speech & Language*, June 2019.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015. Association for Computational Linguistics.

Ann Wennerstrom and Andrew F. Siegel. Keeping the floor in multiparty conversations: Intonation, syntax, and pause. *Discourse Processes*, 36(2), 2003.

Colin W. Wightman, Stefanie Shattuck-Hufnagel, Mari Ostendorf, and Patti J. Price. Segmental durations in the vicinity of prosodic phrase boundaries. *The Journal of the Acoustical Society of America*, 91(3), March 1992.

Jason Williams, Antoine Raux, and Matthew Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3), 2016.

Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2), April 2007.

Marcin Włodarczak and Mattias Heldner. Respiratory Turn-Taking Cues. In *Interspeech*, September 2016a.

Marcin Włodarczak and Mattias Heldner. Respiratory Belts and Whistles: A Preliminary Study of Breathing Acoustics for Turn-Taking. In *Interspeech*, 2016b.

Marcin Wlodarczak and Petra Wagner. Effects of talk-spurt silence boundary thresholds on distribution of gaps and overlaps. *Interspeech*, 2013.

Raymond Wong. How to order a pizza with Amazon Alexa or Google Home. https://mashable.com/article/how-to-order-pizza-with-amazon-alexa-google-home/, 2017.

Victor H Yngve. On getting a word in edgewise. In *Chicago Linguistics Society, 6th Meeting*, 1970.

S. Young, M. Gašić, B. Thomson, and J. D. Williams. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, 101(5), May 2013.

Tiancheng Zhao, Alan W. Black, and Maxine Eskenazi. An incremental turn-taking model with active system barge-in for spoken dialog systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1), 2020.